



Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Estatística

Dissertação de Mestrado

Um Pacote Computacional para a Análise Estatística  
de Processos de Lei de Potência

por

Israel de Freitas Madureira

Orientador: Prof. Gustavo Leonel Gilardoni Avelle

Julho de 2014

**Israel de Freitas Madureira**

**Um Pacote Computacional para a Análise Estatística  
de Processos de Lei de Potência**

Dissertação apresentada ao Departamento de  
Estatística do Instituto de Ciências Exatas  
da Universidade de Brasília como requisito  
parcial à obtenção do título de Mestre em  
Estatística

**Universidade de Brasília**

**Brasília, Julho de 2014.**

# Agradecimentos

Agradeço a Deus por ter me abençoado e disponibilizado um ambiente que permitiu a minha evolução em todos os aspectos da vida e por ter me dado alegria, vontade e disposição para continuar estudando.

Agradeço a meus pais, Ruy e Marta, que sempre incentivaram meus estudos e investiram tempo e dinheiro para isto. Pelo amor incondicional e por tudo que me ensinaram, inclusive o gosto pela ciência.

Agradeço aos meus irmãos: Carolina, Daniel, Rafael, Adilson e Letícia, que me apoiam e estão sempre ao meu lado, pois além de irmãos, são meus amigos.

Agradeço aos meus familiares e amigos por acreditarem em mim e pelo incentivo, apoio e carinho. Vocês são muito importantes na trajetória da minha vida.

Agradeço ao meu orientador, Gustavo Gilardoni, pela paciência e competência com que me orientou e me ajudou a tornar um profissional melhor. Sua orientação enriqueceu meus conhecimentos e me incentivou a continuar evoluindo no campo da estatística.

Agradeço a todos que me ajudaram, direta ou indiretamente, a realizar este sonho.

# Conteúdo

<b>Agradecimentos</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>3</b>
<b>Resumo</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>1 Introdução</b>	<b>6</b>
1.1 Programa R . . . . .	8
1.2 PPNH e PLP . . . . .	10
1.3 Função de Verossimilhança . . . . .	14
<b>2 Inferência Clássica</b>	<b>19</b>
2.1 Estimadores de Máxima Verossimilhança . . . . .	19
2.2 <i>Bootstrap</i> Paramétrico . . . . .	27
2.3 Funções dos parâmetros . . . . .	28
2.4 Simulação . . . . .	30
2.4.1 Programação para Simulação . . . . .	34
<b>3 Ajuste do Modelo</b>	<b>38</b>
3.1 Gráfico de Duane . . . . .	39
3.2 Gráfico TTT . . . . .	41
3.3 Gráfico Nelson-Aalen . . . . .	43
3.4 Teste de Kolmogorov-Smirnov . . . . .	46
3.5 Teste de Cramér-von Mises . . . . .	49

<b>4</b>	<b>Inferência Bayesiana</b>	<b>52</b>
4.1	Estimação Bayesiana dos Parâmetros . . . . .	55
4.2	Estimação das funções dos parâmetros . . . . .	56
<b>5</b>	<b>Política de Manutenção</b>	<b>59</b>
5.1	Estimação de $\tau$ . . . . .	62
<b>6</b>	<b>Conclusão</b>	<b>64</b>

# Lista de Figuras

1.1	Diferentes Comportamentos de Sistemas . . . . .	11
1.2	Função Intensidade com Falhas Simuladas . . . . .	13
2.1	Número Acumulado de Falhas pelo Tempo - Exemplo 1 . . . . .	22
2.2	Sistemas e Falhas do Exemplo 2 . . . . .	24
2.3	Sistemas e Falhas do Exemplo 3 . . . . .	27
3.1	Duane Plot do Exemplo 6 . . . . .	40
3.2	Duane Plot do Exemplo 1 . . . . .	41
3.3	TTT Plot do Exemplo 1 . . . . .	42
3.4	TTT Plot do Exemplo 6 . . . . .	42
3.5	Esquema Nelson Aalen - Exemplo 4 . . . . .	43
3.6	Nelson Aalen Plot do Exemplo 4 . . . . .	44
3.7	Gráfico Nelson Aalen do Exemplo 2 . . . . .	45
3.8	Gráfico Nelson Aalen do Exemplo 3 . . . . .	46
5.1	Função Intensidade com Manutenções $\tau = 3$ . . . . .	60

# Resumo

Este trabalho teve por objetivo desenvolver e implementar um pacote estatístico escrito em *R* para analisar a confiabilidade em sistemas reparáveis. O pacote se chama *NHPPplp*, do inglês *Non-homogeneous Poisson Process with Power Law Process intensity*. Os dados são modelados de acordo com o Processo de Lei de Potência, um dos mais populares modelos paramétricos na literatura de confiabilidade. Os comandos no pacote estimam os parâmetros do modelo e funções deles, além de verificar a qualidade do ajuste. Ele permite elaborar uma Política de Manutenção Preventiva Ótima para sistemas reparáveis. O estudo envolve tanto a abordagem clássica como a bayesiana.

**Palavras Chave:** *Processo de Poisson Não-Homogêneo, Processo de Lei de Potência, Pacote no R.*

# Abstract

The objective of the this work is to develop and implement a statistical package written in *R* to analyze the reliability of repairable systems. The package is called *NHPPplp* (*Non-homogeneous Poisson Process with Power Law Process intensity*). The Power Law Process is one of the most popular parametric models in the reliability literature. The functions available in the package estimates the model parameters and function of them and also verifies the goodness-of-fit. It allows the user to estimate an Optimal Preventive Policy for repairable systems. The approach includes both classical and bayesian inference.

**key words:** *Non-homogeneous Poisson Process, Power Law Process, package R.*



# Capítulo 1

## Introdução

A análise de confiabilidade ou sobrevivência é o estudo sobre o tempo até a ocorrência de um evento de interesse. Este tipo de análise é utilizado na indústria, medicina, economia, engenharia, etc. Quando o assunto está relacionado à medicina, o evento de interesse pode ser o tempo até a morte do paciente, a cura de uma doença, etc. Para a engenharia, pode ser o tempo até um carro quebrar, uma lâmpada queimar, uma máquina falhar ou um software parar de rodar.

A análise pode ocorrer tanto para sistemas reparáveis quanto não-reparáveis. Sistemas não-reparáveis são aqueles descartados ou retirados do estudo após o evento de interesse ou falha, como por exemplo, a lâmpada que queima ou a morte de um paciente. Os sistemas reparáveis são aqueles que, após falharem, são consertados de forma a continuar funcionando. Exemplos de tais sistemas são carros, celulares, máquinas industriais, etc.

Além do conserto executado após cada falha, os sistemas reparáveis também podem receber uma manutenção preventiva em momentos determinados pelo operador. Uma política de manutenção pode determinar um tempo ótimo a fim de diminuir os custos e aumentar os ganhos em inúmeras empresas que possuem máquinas ou equipamentos.

O estudo de análise de confiabilidade de sistemas reparáveis permite diferentes modelagens para os dados. Entretanto o foco do trabalho está na modelagem chamada Processo de Poisson Não Homogêneo (PPNH) com função de intensidade Weibull ou Processo de Lei de Potência (PLP). Aqui não serão consideradas covariáveis.

Análises deste tipo são complexas e exigem conhecimentos em inferência estatística e estatística computacional. Rotinas algorítmicas pré-estabelecidas seriam de grande valia.

Esta dissertação pretende construir uma "biblioteca" que estará disponível na plataforma *R*, além de servir como manual de seu uso, que possuirá comandos para análise de confiabilidade em sistemas reparáveis. A biblioteca se chamará *NHPPplp*, do inglês *Nonhomogeneous Poisson Process with Power Law Process*. A plataforma *R* é um *software* livre e gratuito para análises estatísticas que permite aos usuários incrementarem novas bibliotecas ou pacotes. Mais informações podem ser obtidas no endereço <http://cran.r-project.org>. Estas bibliotecas possuem um conjunto de comandos, que realizam rotinas algorítmicas com a intenção de fazer análises estatísticas específicas. Atualmente, esta plataforma já contém um pacote chamado "survival", que possui rotinas para análise de confiabilidade em sistemas não-reparáveis. Entretanto, ainda não há disponível um pacote para os sistemas reparáveis. Assim, metodologias estatísticas no programa *R* para análise de confiabilidade em sistemas reparáveis são a maior contribuição do referido trabalho. Além da criação do pacote estatístico no *R*, um estudo bibliográfico e a demonstração de parte da teoria utilizada são produtos desta dissertação.

Estudos sobre confiabilidade em sistemas reparáveis podem ser visto em Crow [1974], que propôs teorias que serão utilizadas aqui, também em Baker [1996], Molitor and Rigdon [1993], Pulcini [2001] e Rigdon and Basu [2000]. O trabalho destes últimos serviram de base para o desenvolvimento de alguns capítulos desta dissertação. Algumas situações práticas podem ser encontradas em Baracho [2001], Freitas [2007], Gilardoni and Colosimo [2007], Hodel [2010] e Motta [2002].

O desenvolvimento da Biblioteca ajuda na estimação de uma modelagem específica, mas não é a única. O assunto sobre sistemas reparáveis é amplo e permite uma série de abordagens, com suposições diferentes das feitas aqui. Este trabalho está focado na forma mais encontrada de lidar com dados de sistemas reparáveis, mas a Biblioteca pode ainda ser ampliada no sentido de incluir novos comandos que abrangem diferentes tipos de funções de intensidade, além de outras técnicas de verificação de Ajuste do Modelo e Política de Manutenção.

Neste capítulo, há uma breve introdução ao Programa  $R$ , ao Processo de Poisson Não Homogêneo (PPNH) e ao Processo de Lei de Potência (PLP) de forma a desenvolver a função de verossimilhança. No Capítulo 2 estão descritas como foram desenvolvidas as estimações dos parâmetros da modelagem escolhida e suas funções dentro da abordagem clássica. O Capítulo 3 descreve como são feitas as análises de ajuste do modelo através de métodos gráficos e por testes de hipóteses estatísticas. No Capítulo 4 está a abordagem bayesiana para as mesmas estimações. Por fim, o Capítulo 5 trata da Política de Manutenção Ótima tanto através da abordagem clássica como da bayesiana.

## 1.1 Programa R

Antes de iniciarmos o assunto propriamente dito, serão apresentadas algumas características necessárias do programa  $R$  para que se entenda um pouco da programação que está no corpo da dissertação e em anexo. O ambiente  $R$  é uma linguagem de programação simples e eficiente, que possui um conjunto de rotinas computacionais para manipulação de dados, cálculos estatísticos e elaboração de gráficos. O ambiente  $R$  é uma versão livre e aberta do  $S$ . A linguagem  $S$  foi criada por Rick Bker, John Chambers e Allan Wilks, nos Laboratórios AT&T. O programa  $R$  encontra-se disponível na internet em <http://cran.r-project.org>. Nele podem ser obtidas mais informações, inclusive apostilas com versões em português, em especial Team et al. [2000], que serviu de base para o desenvolvimento desta seção.

O programa possui oito bibliotecas padrão, assim chamadas porque suas rotinas já estão incluídas no programa, não sendo necessário instalá-las depois. Entretanto, ele permite que novas metodologias sejam acrescentadas através de "Bibliotecas". Bibliotecas são onde ficam armazenadas rotinas específicas e é necessário baixá-las para utilizar seus comandos. É na construção de uma dessas bibliotecas que estamos interessados, chamada de  $NHPP_{plp}$ , do inglês *Nonhomogeneous Poisson Process with Power Law Process*.

A forma mais comum de usar o  $R$  é numa estação de trabalho em ambiente de janelas. Pode-se usar programas adicionais para ajudar na manipulação do  $R$ , como

por exemplo o *RStudio* ([www.rstudio.com](http://www.rstudio.com)). As saídas ou resultados dos comandos são obtidos na mesma janela onde eles foram executados, chamada de `console`, ou uma nova janela será aberta para o caso do resultado ser um gráfico ou figura.

As entidades criadas e manuseadas pelo usuário do programa são denominados de **objetos**. O *R* trabalha com objetos de diferentes tipos, os principais são:

- **Vetores**. São os mais comuns e importantes objetos do *R*, denominados `vector`;
- **Matrizes**. São a generalização multidimensional dos vetores, denominados `matrix`;
- **Listas**. São os mais gerais e pode armazenar dados de diferentes modos, ou seja, numérico, lógico, alfanumérico, etc, denominados `list`;
- **Folhas de Dados**. São estruturas em forma de Tabela matricial, onde as colunas podem ser de diferentes modos como nas Listas e as linhas, em geral, representam um indivíduo da amostra ou uma observação, denominados no *R* como `data.frame`;
- **Funções**. São objetos que podem ser guardados no *R* e futuramente usados, ampliando a capacidade do programa. São denominados `function`.

O símbolo `=` ou `<-` é utilizado para atribuir ao objeto o que ele passará a armazenar. Por exemplo, `x<-2` ou `x=2` quer dizer que `x` é um objeto vetor numérico de dimensão 1 e é igual a 2. A programação pode ser desenvolvida de forma análoga a outros programas, onde `for`, `if`, `while`, ... funcionam de forma semelhante.

Vejamos um exemplo com o uso de colchetes `[]` e cerquilha `#`,

```
> x<-c(1,2,3,4)#vetor numérico
> x[x>2]
[1] 3 4
```

No exemplo, `x` é um vetor com os valores (1, 2, 3, 4) armazenados e `x[x>2]` é um vetor com a restrição de `x>2`, que conforme a saída do programa, é igual a (3, 4). O

uso de `#` permite que sejam feitos comentários, pois o que está a sua direita não é considerado no momento em que o comando é executado.

A biblioteca *NHPPplp* é, na verdade, um conjunto de funções, que quando executadas são chamadas de **comando**. Essas funções possuem parâmetros de entrada, que devem ser preenchidos no momento da utilização da função como comando para obter os resultados desejados. No exemplo abaixo, `plp.ml` é um comando, com seu primeiro parâmetro de entrada igual a `AMSAA`, uma matriz de dados armazenada no biblioteca, da qual falaremos mais tarde. A saída está em forma de folha de dados (`data.frame`), indicando estimativas pontuais e intervalares e os respectivos desvios padrões de dois parâmetros: `shape` e `scale`.

```
> plp.ml(AMSAA)
  Parameter Estimate StdDev CI.low  CI.up
1   shape    0.615    0.103  0.444  0.853
2   scale    3.525    2.558  0.850 14.614
```

## 1.2 PPNH e PLP

Para o desenvolvimento desta seção, seguimos Rigdon and Basu [2000]. Começamos por caracterizar o tempo  $t$  até a ocorrência do evento ou falha. Este "tempo" pode ser medido como o tempo cronológico em dias, meses, horas, minutos ou segundos dependendo do objeto de estudo. Entretanto algumas pesquisas possuem características onde é mais simples ou adequado utilizarmos uma medida diferente do tempo até a ocorrência da falha, por exemplo: a quantidade de quilômetros rodados por um carro até quebrar; quantas vezes a lâmpada é ligada e desligada até queimar; ou, quantas cópias são feitas por uma copiadora até falhar. Todavia, todos são representados pela letra  $t$  de tempo. Como se trata de sistemas reparáveis,  $t_1$  seria o tempo da primeira falha,  $t_2$  da segunda falha e assim por diante, de tal forma que  $0 < t_1 < t_2 < \dots < t_n$ , com  $n$  sendo o número total de falhas observadas em um sistema.

Os tempos entre as falhas são definidos como  $x_i = t_i - t_{i-1}$  com  $i = 1, \dots, n$  onde  $t_0 = 0$ . Se os  $x_i$ 's tendem a crescer com o tempo então o sistema está melhorando, pois

o tempo entre as falhas tende a ficar cada vez maior. Se os  $x'_i$ s tendem a decrescer com o tempo, o sistema está deteriorando e as falhas tendem a acontecer com mais frequência. Há ainda outras situações, onde a esperança dos  $x'_i$ s são constantes no tempo ou decrescem e depois crescem novamente. A figura (1.1) mostra casos simulados com três situações diferentes, onde é possível perceber que: no sistema que tende a melhorar com o tempo, há mais falhas no início e depois elas ficam mais espaçadas; no sistema com intensidade constante, os tempos das falhas não seguem uma regra e estão espalhadas de forma mais ou menos uniformes; e, o sistema que tende a deteriorar, as falhas ficam mais perto umas das outras com o passar do tempo.

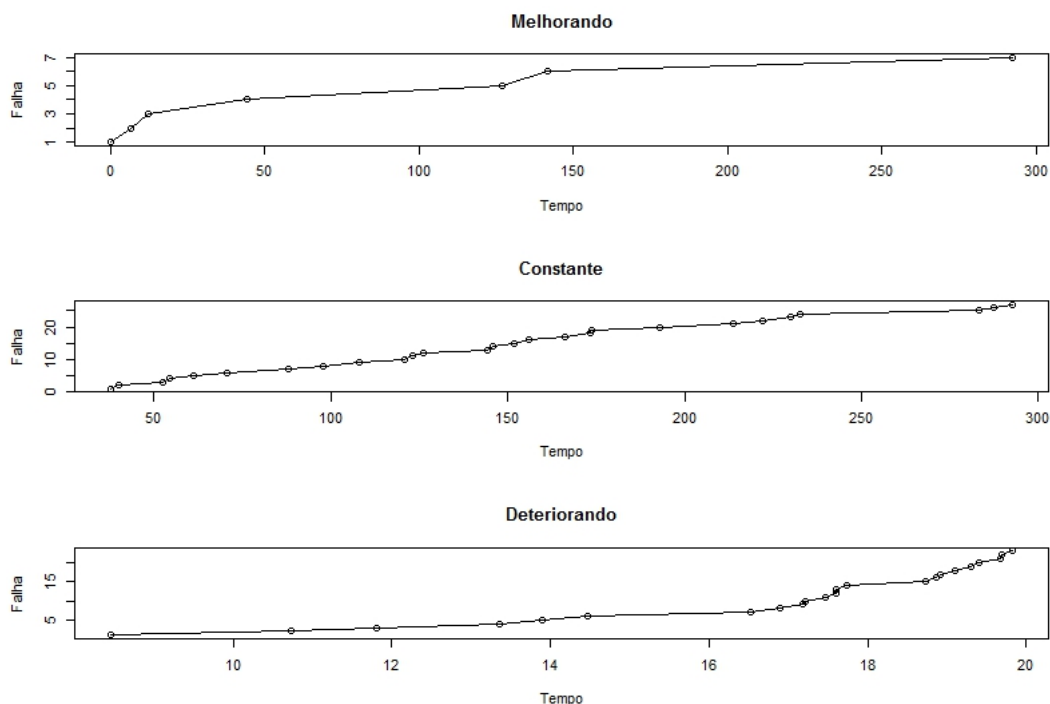


Figura 1.1: Diferentes Comportamentos de Sistemas

Algumas funções são tipicamente estudadas em análise de confiabilidade para sistemas não reparáveis, aqueles descartados após a primeira e única falha. Apresentamos aqui porque possuem funções análogas a elas em análise de confiabilidade para sistemas reparáveis, como mostraremos mais à frente. Inicialmente, caracterizamos a função de distribuição acumulada  $F(t)$ , que é a probabilidade de ocorrer a falha antes do tempo  $t$ ,

$$F(t) = P(T \leq t).$$

Sua função complementar  $S(t) = 1 - F(t)$ , chamada de função de confiabilidade ou sobrevivência, é a probabilidade de ocorrer a falha após o tempo  $t$ . A função de densidade de probabilidade  $f(t)$  pode ser extraída da derivada da  $F(t)$ , ou mesmo da negativa da derivada da função sobrevivência  $S(t)$ ,

$$f(t) = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt}.$$

Há também a função risco  $h(t)$  que é o limite da probabilidade da falha ocorrer em um pequeno intervalo de tempo dado que o sistema não falhou até aquele momento, dividido pelo tamanho deste intervalo. Podemos mostrar que ela é a razão entre a função de densidade de probabilidade  $f(t)$  e a função sobrevivência  $S(t)$ ,

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t < T < t + \Delta t \mid T > t)}{\Delta t} = \frac{f(t)}{S(t)}.$$

No estudo de sistemas reparáveis, introduzimos também a notação  $N(t)$ , que é o número de falhas ocorridas no intervalo  $(0, t]$ , pois diferentemente dos sistemas não reparáveis, os números de falhas podem ser maiores do que um. Desta forma podemos encontrar a esperança do valor médio de falhas no intervalo de tempo  $(0; t]$ ,

$$\Lambda(t) = E(N(t)).$$

Da derivada de  $\Lambda(t)$ , encontramos a razão de ocorrência de falhas *ROCOF*, do inglês *Rate of Occurrence of Failures*. Se a possibilidade de duas ou mais falhas ocorrerem ao mesmo tempo for zero, a *ROCOF* é igual a Função de Intensidade  $\lambda(t)$ , que é análogo à função de risco  $h(t)$  para os sistemas reparáveis, pois é a

probabilidade de ocorrer uma falha em um pequeno intervalo de tempo dividido pelo tamanho deste intervalo, definido por

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{P(N(t, t + \Delta t] = 1)}{\Delta t},$$

onde  $N(t, t + \Delta t] = N(t + \Delta t) - N(t)$  é o número de falhas no intervalo de tempo  $(t, t + \Delta t]$ .

No restante do trabalho, assumiremos que  $N(t)$  é um Processo de Poisson Não Homogêneo (PPNH), isto é,  $N(t)$  satisfaz as seguintes propriedades:

1.  $N(0) = 0$
2.  $N(t)$  tem incrementos independentes, ou seja, para quaisquer valor  $a < b \leq c < d$ , as variáveis aleatórias  $N(a, b]$  e  $N(c, d]$  são independentes.
3. Existe uma função não negativa  $\lambda(t)$ , tal que  $\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{P(N(t, t + \Delta t] = 1)}{\Delta t}$
4.  $\lim_{\Delta t \rightarrow 0} \frac{P(N(t, t + \Delta t] \geq 2)}{\Delta t} = 0$

Sendo que  $\lambda(t)$  é chamada de função de intensidade do processo. Pode-se mostrar que se o processo  $N(t)$  satisfaz essas quatro condições, então  $N(t)$  tem distribuição de Poisson, onde

$$P(N(t) = x) = \frac{[\Lambda(t)]^x e^{-[\Lambda(t)]}}{x!}, \quad x = 0, 1, 2, \dots \quad (1.1)$$

com média e variância,

$$\Lambda(t) = \int_0^t \lambda(u) du. \quad (1.2)$$

A escolha do processo de Poisson se justifica porque estamos sob a suposição de reparo mínimo (*as bad as old- ABAO*), isto é, o sistema volta à mesma condição que estava antes da falha ocorrer, após realizado o reparo. A figura (1.2) mostra que não há descontinuidade na curva da função de intensidade após os tempos de falha. Em outras palavras, quando estamos sob reparo mínimo, o sistema volta a ter a mesma



probabilidade de falha que havia imediatamente antes da falha acontecer. Observe também que o tempo de duração para o reparo não é considerado, pois em geral, trabalhamos com o tempo em operação do processo.

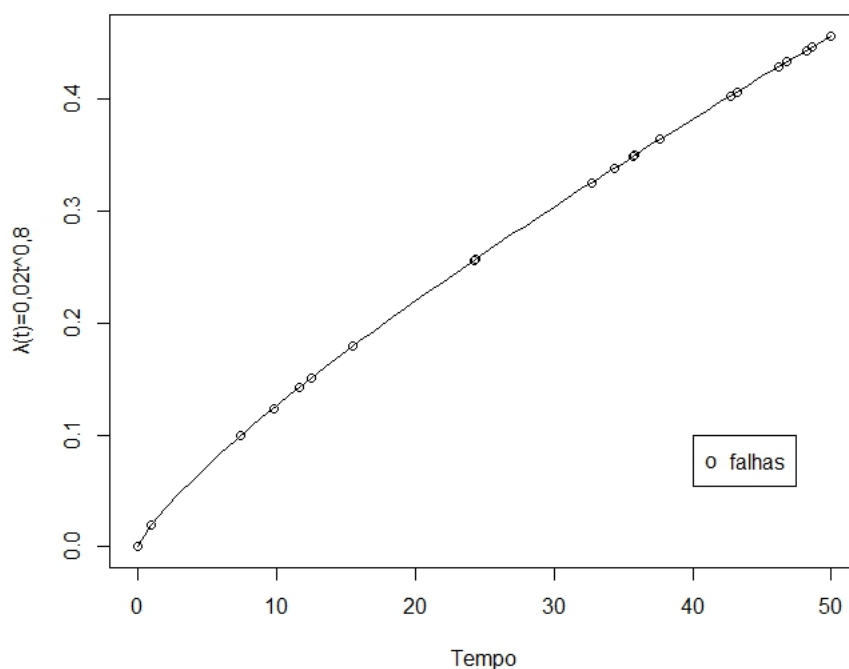


Figura 1.2: Função Intensidade com Falhas Simuladas

Se a função de intensidade  $\lambda(t) = \lambda$  é constante no tempo, o processo é dito estacionário, ou seja,  $E(N(t; t + s))$  é independente de  $t$ , isto caracteriza um processo homogêneo, atendido pela sigla PPH, de Processo de Poisson Homogêneo. Entretanto, se a função de intensidade  $\lambda(t)$  depende do tempo  $t$ , ele é chamado de Processo de Poisson Não- Homogêneo, PPNH. O caso de PPH será tratado apenas como caso particular do PPNH.

A função de intensidade  $\lambda(t)$  pode depender de  $t$  de diferentes maneiras, sendo assim, temos várias opções para a modelagem paramétrica. Abaixo, seguem algumas:

$$\begin{aligned}
\lambda(t) &= \alpha \left[ 1 - \frac{1}{\sqrt{1+t/\beta}} \right] && \text{proposta em Gilardoni and Colosimo [2007]} \\
&\lambda(t) = e^{\alpha+\beta t} && \text{conhecida como log-linear} \\
\lambda(t) &= \left( \frac{\beta}{\theta} \right) \left( \frac{t}{\theta} \right)^{\beta-1} && \text{proposta em Crow [1974]}
\end{aligned} \tag{1.3}$$

Esta última proposta é chamada de função intensidade Weibull ou Processo da Lei de Potência (PLP), é a mais utilizada na área de confiabilidade de sistemas reparáveis, pois além da capacidade de modelar casos de processos que melhoram, deterioram ou com intensidade constante, ela possui uma manipulação matemática mais simples. A tabela (1.1) mostra a flexibilidade e os diferentes comportamentos que a função de intensidade PLP  $\lambda(t)$  pode adquirir para diferentes valores de  $\beta$ :

Valor	Propriedade
$0 < \beta < 1$	$\lambda(t)$ decrescente
$\beta = 1$	$\lambda(t)$ constante
$1 < \beta < 2$	$\lambda(t)$ crescente e côncava
$\beta = 2$	$\lambda(t)$ crescente e linear
$\beta > 2$	$\lambda(t)$ crescente e convexa

Tabela 1.1: Diferentes comportamentos de  $\lambda(t)$  no PLP

### 1.3 Função de Verossimilhança

Na continuação, apresentamos a derivação da função de verossimilhança para um único sistema, seguindo o desenvolvimento de Rigdon and Basu [2000]. A distribuição de Poisson é a base para encontrarmos as distribuições dos tempos das falhas  $t_1, t_2, \dots, t_n$ , sendo  $n$  o número de total de falhas do sistema, suas funções de sobrevivência  $S(t_i)$ , suas funções acumuladas  $F(t_i)$  para quaisquer  $i$  e a função conjunta dos  $t_i$ s, que será a base para a construção da função de verossimilhança.

A seguir se desenvolverão duas verossimilhanças. Isto porque, há duas formas de escolher a estratégia de parada de um estudo, seja pelo número de falhas pré-determinado ou fixo, chamado truncamento por falha, seja pelo tempo total de observação do sistema, chamado truncamento temporal. Esta escolha faz diferença

na análise de confiabilidade, pois a função conjunta  $f(t_1, t_2, \dots, t_n)$  vai ser a função de verossimilhança para o caso de truncamento por falha, onde  $n$  é fixo e  $t_n$  é uma variável aleatória. Entretanto, no caso de truncamento temporal, o tempo total de observação do sistema, representado por  $T$ , é fixo e  $n$  é uma variável aleatória, sendo assim sua função de verossimilhança é  $f(t_1, t_2, \dots, t_n, n)$ , como mostraremos mais a frente. Nesse caso,  $T$  é um tempo que chamamos de censura, pois em  $T$  não ocorreu uma falha, mas o sistema é observado até este momento.

Começamos com o caso de truncamento por falha. Há uma relação entre a distribuição de  $N(t)$  e  $S(t)$  para a primeira falha  $t_1$ ,

$$S_1(t_1) = P(T_1 > t_1) = P(N(0, t_1] = 0) = e^{-\int_0^{t_1} \lambda(u) du},$$

ou seja, a probabilidade da primeira falha ocorrer após  $t_1$  equivale à probabilidade de não ocorrer nenhuma falha até  $t_1$ . Assim, temos a função de distribuição acumulada  $F_1(t_1)$ ,

$$F_1(t_1) = 1 - e^{-\int_0^{t_1} \lambda(u) du}.$$

Logo,

$$f_1(t_1) = -\frac{d}{dt_1} S_1(t_1) = -\frac{d}{dt_1} e^{-\int_0^{t_1} \lambda(u) du} = \lambda(t_1) e^{-\int_0^{t_1} \lambda(u) du}.$$

Da mesma forma pode-se encontrar as densidades condicionais ( $f_2(t_2 | t_1), f_3(t_3 | t_2), \dots, f_n(t_n | t_{n-1})$ ). Por exemplo

$$f_2(t_2 | t_1) = -\frac{d}{dt_2} S_2(t_2 | t_1) = -\frac{d}{dt_2} P(t_2 > t_2 | t_1) = -\frac{d}{dt_2} e^{-\int_{t_1}^{t_2} \lambda(u) du} = \lambda(t_2) e^{-\int_{t_1}^{t_2} \lambda(u) du}.$$

Analogamente,

$$\begin{aligned}
f_j(t_j | t_{j-1}, t_{j-2}, \dots, t_1) &= f_j(t_j | t_{j-1}) \\
&= -\frac{d}{dt_j} S_j(t_j | t_{j-1}) \\
&= -\frac{d}{dt_j} P(T_j > t_j | t_{j-1}) \\
&= -\frac{d}{dt_j} e^{-\int_{t_{j-1}}^{t_j} \lambda(u) du} \\
&= \lambda(t_j) e^{-\int_{t_{j-1}}^{t_j} \lambda(u) du} \quad j = 0, 1, 2, \dots \quad (1.4)
\end{aligned}$$

A função conjunta  $f(t_1, t_2, t_3, \dots, t_n)$  é dada pela multiplicação das funções de densidade de probabilidade,

$$\begin{aligned}
L(\lambda) = f(t_1, t_2, \dots, t_n) &= f_1(t_1) f_2(t_2 | t_1) f_3(t_3 | t_2, t_1) f_n(t_n | t_{n-1}, t_{n-2}, \dots, t_1) \\
&= \left( \prod_{i=1}^n \lambda(t_i) \right) e^{-\Lambda(t_n)}. \quad (1.5)
\end{aligned}$$

Aqui chegamos ao primeiro objetivo que era desenvolver a função de verossimilhança  $L(\lambda)$  em caso de truncamento por falha. Para chegarmos a função de verossimilhança em caso de truncamento temporal  $L(\lambda) = f(t_1, t_2, t_3, \dots, t_n, n)$ , em que  $T$  é fixo, logo  $n$  é aleatório, começamos pela definição da função,

$$\begin{aligned}
f(t_1, t_2, t_3, \dots, t_n, n) &= \lim_{\Delta(t_i) \rightarrow 0} \frac{P(t_1 < T_1 \leq t_1 + \Delta t_1, \dots, t_n < T_n \leq t_n + \Delta t_n, n)}{\prod_{i=1}^n \Delta(t_i)} \\
&= \lim_{\Delta(t_i) \rightarrow 0} P(N(0, t_1) = 0) P(N[t_1, t_1 + \Delta t_1] = 1) \dots \\
&\quad \dots P(N[t_n, t_n + \Delta t_n] = 1) \dots \\
&\quad \dots P(N(t_n + \Delta t_n, t] = 0) / \left( \prod_{i=1}^n \Delta(t_i) \right). \quad (1.6)
\end{aligned}$$

Como o processo de Poisson possui incrementos independentes, foi possível particionar o tempo total e encontrar as suas probabilidades em função do número de falhas  $N(t)$  para cada "comprimento do tempo", que possuem distribuições de Poisson

conhecidas. Substituindo as funções de distribuição na equação (1.6), temos

$$\begin{aligned}
L(\lambda) = f(t_1, t_2, t_3, \dots, t_n, n) &= e^{-\Lambda(T)} \left( \lim_{\Delta t_1 \rightarrow 0} \frac{\Lambda(t_1 + \Delta t_1) - \Lambda(t_1)}{\Delta t_1} \right) \times \dots \\
&\times \dots \left( \lim_{\Delta t_n \rightarrow 0} \frac{\Lambda(t_n + \Delta t_n) - \Lambda(t_n)}{\Delta t_n} \right) \\
&= \left( \prod_{i=1}^n \lambda(t_i) \right) e^{-\Lambda(T)}. \tag{1.7}
\end{aligned}$$

Assim temos a função de verossimilhança  $L(\lambda)$  para o caso de truncamento temporal, assim como já tínhamos para o caso de truncamento por falha. Se comparamos (1.5) e (1.7), temos que  $L(\lambda)$  é semelhante para ambos os tipos de truncamento, fazendo  $t_n = T$ . Se a análise for para  $k$  sistemas independentes, com  $k \geq 2$ , a função de verossimilhança é  $L(\lambda) = \prod_{i=1}^k L_k(\lambda)$ .

Os valores para os parâmetros que maximizam as funções de verossimilhança  $L(\lambda)$  são chamadas de Estimadores de Máxima Verossimilhança EMV, que serão as estimações para os parâmetros de forma ( $\beta$ ) e de escala ( $\theta$ ) da função de intensidade do PLP. Substituindo  $\lambda$  nas funções de verossimilhança (1.5) e (1.7) pela função de intensidade (1.3), temos a função  $L(\lambda)$  para um único sistema a ser otimizada,

$$L(\beta, \theta) = \exp \left( - \left( \frac{T}{\theta} \right)^\beta \right) \prod_{i=1}^n \left( \frac{\beta}{\theta} \left( \frac{t_i}{\theta} \right)^{\beta-1} \right) \tag{1.8}$$

onde  $T = t_n$  em caso de truncamento por falha.

# Capítulo 2

## Inferência Clássica

Com base na função de verossimilhança, apresentamos um algoritmo para calcular os Estimadores de Máxima Verossimilhança (EMV) dos parâmetros e Intervalos de Confiança (IC) deles. Um método de maximização de funções aparece como primeira opção para o cálculo dos EMV's. As Inferências sobre funções dos parâmetros serão obtidas pelo Método Delta. O método *bootstrap* é uma segunda opção de estimação disponível tanto para os parâmetros quanto para funções dos parâmetros. Mostraremos como simular dados PPNH que envolve um método direto chamado Transformação Integral de Probabilidade no apêndice do capítulo. Todos esses casos são acompanhados de exemplos utilizando os comandos que estão na biblioteca *NHPPplp*.

### 2.1 Estimadores de Máxima Verossimilhança

Na utilização da biblioteca *NHPPplp*, basta executar o comando `plp.ml`, que exige a entrada do banco de dados em formato de *matriz* ou *folha de dados*, para obter como saída a estimação pontual e intervalar para os parâmetros. Apresentaremos nesta seção quais técnicas são usadas para isso e alguns exemplos.

A *matriz* ou a *folha de dados* deve possuir duas ou três colunas: a 1<sup>a</sup>, com os tempos de falha e de censura observados; a 2<sup>a</sup>, com indicação de 1 (ou **TRUE**) caso o tempo se refira a uma falha ocorrida e 0 (ou **FALSE**) caso o tempo se refira a uma censura (os tempos  $T$ 's em caso de truncamento temporal); a 3<sup>a</sup> coluna indica o

sistema correspondente através de valores numéricos ou alfanuméricos, ela é optativa quando se trata de um único sistema. Vejamos o exemplo abaixo:

```
> transformers[1:8,]
      times failure system
[1,]  8.839         1     1
[2,] 17.057         1     1
[3,] 21.887         0     1
[4,]  9.280         1     2
[5,] 16.442         1     2
[6,] 21.887         0     2
[7,] 10.445         1     3
[8,] 13.533         0     3
```

No exemplo, é possível verificar a estrutura dos dados de entrada. Ele mostra as oito primeiras observações de uma matriz denominada `transformers`, que estará disponível na biblioteca aqui desenvolvida, onde a primeira coluna indica os tempos de falhas do sistema, a segunda, se é falha (1) ou censura (0) e a terceira, os sistemas.

Começaremos estimando os parâmetros quando há um único sistema ( $k = 1$ ) em estudo. Transformamos  $L(\lambda)$  em  $\ell(\lambda)$ , que é o logaritmo de  $L(\lambda)$ . Isto se deve ao fato de que os valores de  $x$  que maximizam uma função qualquer  $f(x)$  são os mesmos que maximizam  $\log f(x)$ , sendo que esta última expressão é, em geral, mais tratável matematicamente. Desenvolvendo (1.8) encontramos,

$$\ell(\beta, \theta) = - \left( \frac{T}{\theta} \right)^\beta + n \log \beta - n \beta \log \theta + (\beta - 1) \sum_{i=1}^n \log t_i \quad (2.1)$$

onde  $t_i$  é cada tempo de falha e  $T$  é o tempo total de observação do sistema ou o tempo da última falha  $T = t_n$  em caso de truncamento por falha.

Os valores de  $\beta$  e  $\theta$  que maximizam a função (2.1) são

$$\hat{\beta} = \frac{n}{\sum_{i=1}^n \log(T/t_i)} \quad (2.2)$$

$$\hat{\theta} = \frac{T}{n^{\frac{1}{\hat{\beta}}}}. \quad (2.3)$$

Para encontrarmos estes valores, derivamos a função (2.1) em relação a cada um dos parâmetros e igualamos a 0. No caso de  $\hat{\theta}$ , ele depende do valor de  $\beta$ , então, para estimá-lo substituímos  $\beta$  por seu EMV  $\hat{\beta}$ .

A expressão  $\frac{2n\beta}{\hat{\beta}}$  tem distribuição Qui-Quadrado  $\chi^2$  com  $2(n-1)$  graus de liberdade e  $w = \left(\frac{\hat{\theta}}{\theta}\right)^{\hat{\beta}}$  também é quantidade pivotal para o caso de truncamento por falha, conforme Rigdon and Basu [2000]. Logo, podemos obter Intervalos de Confiança (IC) para  $\beta$  e  $\theta$ ,

$$\frac{\chi_{\alpha/2; 2(n-1)}^2 \hat{\beta}}{2n} < \beta < \frac{\chi_{1-\alpha/2; 2(n-1)}^2 \hat{\beta}}{2n} \quad (2.4)$$

$$\frac{\hat{\theta}}{a^{1/\hat{\beta}}} < \theta < \frac{\hat{\theta}}{b^{1/\hat{\beta}}} \quad (2.5)$$

sendo  $\alpha$ , o grau de significância,  $\hat{\beta}$  e  $\hat{\theta}$ , os EMV's e  $a$  e  $b$  são os percentis de  $w$  obtidos em nosso programa por simulação.

Em caso de truncamento temporal,  $\frac{2n\beta}{\hat{\beta}}$ , condicionado a  $n$ , tem distribuição Qui-Quadrado  $\chi^2$  com  $2n$  graus de liberdade, ou seja, basta substituímos  $\chi_{2(n-1)}^2$  por  $\chi_{(2n)}^2$  em (2.4) para adquirir uma estimação do IC para  $\beta$ . Para o  $\theta$ , não há métodos exatos para estimação do IC. Porém, podemos usar (2.5) como forma aproximada de estimar um IC para  $\theta$  Rigdon and Basu [2000].

**Exemplo 1.** *Trata-se de dados disponíveis na referência Rigdon and Basu [2000]. São os tempos de falha de um gerador de aeronave, truncado por falha em  $n = 14$ , conforme tabela (2.1).*

```
> aalen.plot(gerador)#cria gráfico
> Ex.gerador<-plp.ml(gerador)
```



Número	Tempo de Falha	Número	Tempo de Falha
1	10	8	731
2	55	9	1308
3	166	10	2050
4	205	11	2453
5	341	12	3115
6	488	13	4017
7	567	14	4596

Tabela 2.1: Dados do Exemplo 1

> Ex.gerador

	Parameter	Estimate	StdDev	CI.low	CI.up
1	shape	0.483	0.124	0.239	0.723
2	scale	19.504	42.714	0.121	152.969

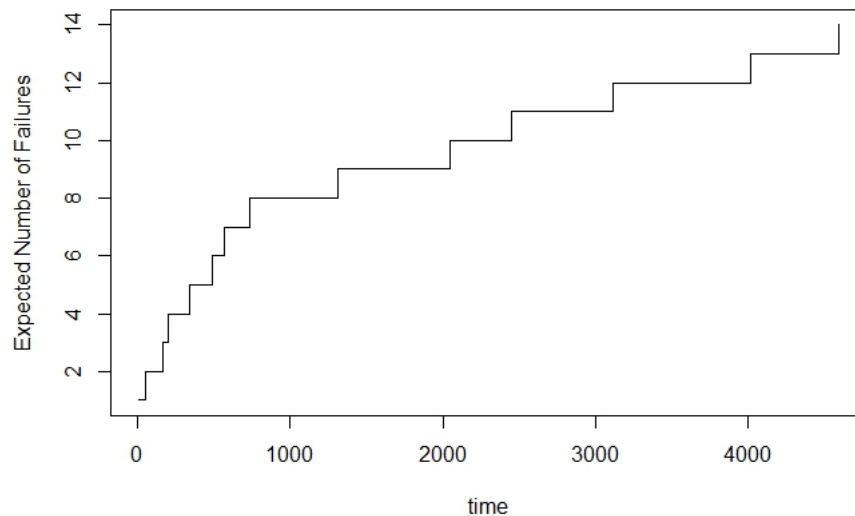


Figura 2.1: Número Acumulado de Falhas pelo Tempo - Exemplo 1

No exemplo,  $\hat{\beta} < 1$ , ou seja, a intensidade estimada é decrescente. Isto significa que o sistema melhora com o passar do tempo. A forma côncava da figura (2.1), que é o número acumulado de falhas pelo tempo, também é um indício dessa melhora.

Caso tenhamos  $k \geq 2$  sistemas, os valores para  $\beta$  e  $\theta$ , em geral, só podem ser estimados numericamente. Internamente, o comando `plp.ml` utiliza-se da função de otimização `optim`, disponível no *R*. O método é iterativo e nos fornecerá os valores

de  $\hat{\beta}$  e  $\hat{\theta}$  que maximizam a função log-verossimilhança quando  $k \geq 2$ ,

$$\begin{aligned} \ell(\beta, \theta) = & -\sum_{j=1}^k \left(\frac{T_j}{\theta}\right)^\beta + \sum_{j=1}^k n_j \log \beta \dots \\ & \dots - \beta \sum_{j=1}^k n_j \log \theta + (\beta - 1) \sum_{j=1}^k \sum_{i=1}^{n_j} \log t_{ij}, \end{aligned} \quad (2.6)$$

sendo que  $t_{ij}$  é o tempo de falha  $i$  do sistema  $j$ ,  $n_j$  é o número de falhas observadas no  $j$ 'ésimo sistema,  $T_j$  é o tempo máximo observado no sistema  $j$  e  $T_j = t_{n_j}$  para o caso de truncamento por falha.

O comando `optim` fornece a Matriz Hessiana no ponto estimado. A Matriz Hessiana da função da log-verossimilhança é constituída pelos valores das derivadas parciais de 2ª ordem. A negativa desta matriz é a informação observada, que mede a quantidade de informação que uma variável aleatória possui sobre um parâmetro. A informação nos pontos estimados para um vetor de parâmetros  $\phi$  é definida por

$$i(\hat{\phi}) = - \left( \frac{\partial}{\partial \phi_i \partial \phi_j} \log(f(X, \hat{\phi})) \Big|_{\hat{\phi}} \right).$$

Sendo assim, se invertermos esta matriz obtemos as variâncias e covariâncias assintóticas dos parâmetros nos pontos estimados. Isso permite uma estimação intervalar para  $\beta$  e  $\theta$ . A matriz de variâncias e covariâncias de  $\hat{\beta}$  e  $\hat{\theta}$  é do tipo  $2 \times 2$ ,

$$\begin{pmatrix} \text{var}(\hat{\beta}) & \text{cov}(\hat{\beta}, \hat{\theta}) \\ \text{cov}(\hat{\beta}, \hat{\theta}) & \text{var}(\hat{\theta}) \end{pmatrix}.$$

Os desvios padrões de  $\hat{\beta}$  e  $\hat{\theta}$  são  $\hat{\sigma}_{\hat{\beta}} = \sqrt{\text{var}(\hat{\beta})}$  e  $\hat{\sigma}_{\hat{\theta}} = \sqrt{\text{var}(\hat{\theta})}$ .

Em caso de  $k \geq 2$  sistemas, temos três possibilidades de estimação dos parâmetros, utilizando as opções do comando `plp.ml`: `log=FALSE`, `log=TRUE` (padrão do comando) e `bootstrap=TRUE`. Esta última será apresentada na próxima seção. Na opção `log=FALSE`, pela teoria assintótica do EMV's, podemos obter Intervalos de Confiança para  $\beta$  e  $\theta$ ,

$$\beta : \hat{\beta} \pm z_{\alpha/2} \hat{\sigma}_{\hat{\beta}}$$

$$\theta : \hat{\theta} \pm z_{\alpha/2} \hat{\sigma}_{\hat{\theta}}$$

sendo que  $z_{\alpha/2}$  é o percentil  $\alpha/2$  de um Distribuição Normal Padrão,  $\alpha$  é o grau de significância,  $\hat{\sigma}$ 's são os desvios padrões estimados de  $\hat{\beta}$  e  $\hat{\theta}$  e  $n$  é o número de falhas ou eventos observados.

A opção `log=TRUE` é uma versão mais robusta do algoritmo pois considera as restrições de  $\beta > 0$  e  $\theta > 0$ , que o são por definição. Parametrizamos  $\eta = \log(\beta)$  e  $\zeta = \log(\theta)$  e maximizamos a log-verossimilhança (2.6) para  $k \geq 2$  sistemas, onde substituímos  $\beta = e^\eta$  e  $\theta = e^\zeta$ . Uma vez obtidas as estimativas de  $\hat{\eta}$  e  $\hat{\zeta}$ , segue pela propriedade de invariância dos EMV's que  $\hat{\beta} = e^{\hat{\eta}}$  e  $\hat{\theta} = e^{\hat{\zeta}}$ . Similarmente, os Intervalos de Confiança podem ser obtidos como

$$\beta : e^{\hat{\eta} \pm z_{\alpha/2} \frac{\hat{\sigma}_{\hat{\eta}}}{\sqrt{n}}}$$

$$\theta : e^{\hat{\zeta} \pm z_{\alpha/2} \frac{\hat{\sigma}_{\hat{\zeta}}}{\sqrt{n}}}.$$

**Exemplo 2.** *Trata-se de dados disponíveis no artigo (Crow [1975]), possui 3 sistemas truncados em  $T = 200$ , onde há registrados  $n = 36$  falhas ( $n_1 = 10, n_2 = 15$  e  $n_3 = 11$ ), conforme tabela (2.2) e figura (2.2). Os dados foram simulados considerando  $\beta = 0,5$  e  $\theta \simeq 2,78$ . Eles estão armazenados na Biblioteca NHPPplp com o nome **AMSAA**.*

Utilizando-se do comando `plp.ml`, podemos obter os valores de  $\beta$  (*shape*) e  $\theta$  (*scale*), suas estimações intervalares e desvios padrões conforme saída do programa. Como padrão, os Intervalos de Confiança possuem 95% de confiabilidade (`conf=0.95`) e `log=TRUE`, por consequência os IC's não são simétricos em relação às estimações pontuais. O IC para  $\beta$ , (0,444;0,853), fica a esquerda de 1, ou seja, temos evidência de que o sistema está melhorando com o tempo.

```
> horizontal.plot(AMSAA)#cria gráfico com linhas horizontais
```

Sistema 1	Sistema 2	Sistema 3
4,3	0,1	8,4
4,4	5,6	32,5
10,2	18,6	44,7
23,5	19,5	48,4
23,8	24,2	50,6
26,4	26,7	73,6
74,0	45,1	98,7
77,1	45,8	112,2
92,1	75,7	129,8
197,2	79,7	136,0
	98,6	195,8
	120,1	
	161,8	
	180,6	
	190,8	

Tabela 2.2: Dados do Exemplo 2

```
> Ex.AMSAA<-plp.ml(AMSAA)
```

```
> Ex.AMSAA
```

```
Parameter Estimate StdDev CI.low CI.up
1 shape 0.615 0.103 0.444 0.853
2 scale 3.525 2.558 0.850 14.614
```

A figura (2.2) tem cada sistema representado por uma linha reta horizontal e os pontos  $\ominus$  são os tempos de falha. Percebe-se que há mais falhas no início e que se tornam mais espaçadas com o tempo, confirmando a tendência de melhora.

O comando `plp.ml` produz um objeto do tipo lista. Além da folha de dados com as estimativas e desvios padrões de  $\beta$  e  $\theta$ , há outras informações que ficaram armazenadas no objeto `Ex.AMSAA`. Por exemplo, a matriz de variâncias e covariâncias de  $\hat{\beta}$  e  $\hat{\theta}$ .

```
> Ex.AMSAA$varcov
```

```
          shape    scale
shape 0.01051771 0.243347
scale 0.24334697 6.542099
```

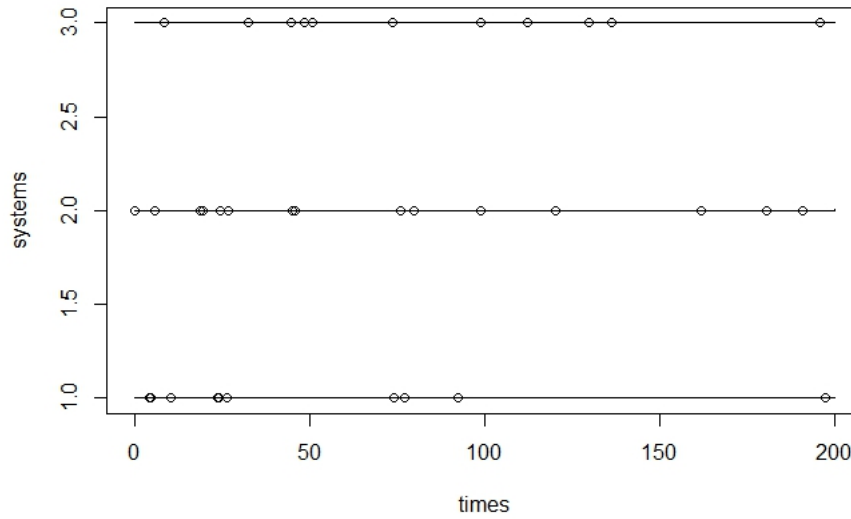


Figura 2.2: Sistemas e Falhas do Exemplo 2

**Exemplo 3.** Nos artigos (Gilardoni and Colosimo [2007]), (Gilardoni and Colosimo [2011]) e (Oliveira et al. [2012]), há uma exemplo que envolve 30 transformadores de potência de 300 e 345 quilovolts de uma Companhia Elétrica brasileira. Eles foram observados entre janeiro de 1.999 e julho de 2.001, totalizando 21.888 horas. Em nossa análise, cada transformador é visto como um sistema, se ele passa por uma manutenção que o deixa como novo, o sistema é censurado neste ponto e na continuação do estudo é acrescentado um novo sistema para este mesmo transformador "reformado". Ao todo, temos 40 sistemas, os tempos de falhas ou censurados estão armazenados na matriz `transformers`, suas oito primeiras observações estão mostradas na página (20). A figura (2.3) tem cada um de seus sistemas representado por uma linha reta, com comprimento igual ao tempo total de observação do sistema e as falhas indicadas por  $\ominus$ .

Com o comando `plp.ml`, estimamos os valores para  $\beta$  e  $\theta$  e seus Intervalos de Confiança de 90%. Pela saída do programa `R`, vemos que  $1,435 < \beta < 2,774$  e  $20,172 < \theta < 29,431$ .

```
> horizontal.plot(transformers)
> Ex.transformers<-plp.ml(transformers,conf=.90)
```

```
> Ex.transformers
```

	Parameter	Estimate	StdDev	CI.low	CI.up
1	shape	1.995	0.400	1.435	2.774
2	scale	24.366	2.798	20.172	29.431

A matriz de variâncias e covariâncias de  $\hat{\eta}$  e  $\hat{\zeta}$ , que equivalem a  $\log(\hat{\beta})$  e  $\log(\hat{\zeta})$ , também fica disponível no objeto `Ex.transformers`, conforme saída do programa.

```
> Ex.transformers$log.varcov
```

	log(shape)	log(scale)
log(shape)	0.040180633	-0.007007641
log(scale)	-0.007007641	0.013185735

É possível obter a estimativa sem o uso da parametrização em  $\eta$  e  $\zeta$ , incluindo a opção `log=FALSE` no comando `plp.ml`. Neste caso, os Intervalos de Confiança de 90%, são  $1,337 < \beta < 2,653$  e  $19,764 < \theta < 28,968$ . Eles são simétricos em relação a estimativa pontual.

```
> plp.ml(transformers,conf=.90,log=FALSE)
```

	Parameter	Estimate	StdDev	CI.low	CI.up
1	shape	1.995	0.400	1.337	2.653
2	scale	24.366	2.798	19.764	28.968

Neste exemplo,  $\lambda(t)$  é crescente e esse crescimento tem uma relação aproximadamente linear com o tempo  $t$  ( $\beta \approx 2$ ). A figura (2.3) representa os 40 sistemas em semirretas horizontais, truncadas em diferentes tempos  $T_j$ . Em alguns sistemas não foram observadas falhas.

## 2.2 *Bootstrap* Paramétrico

A técnica *bootstrap* tornou-se bastante difundida, em especial, com a era dos computadores, pois ela se utiliza do potencial de cálculo dessas máquinas que podem

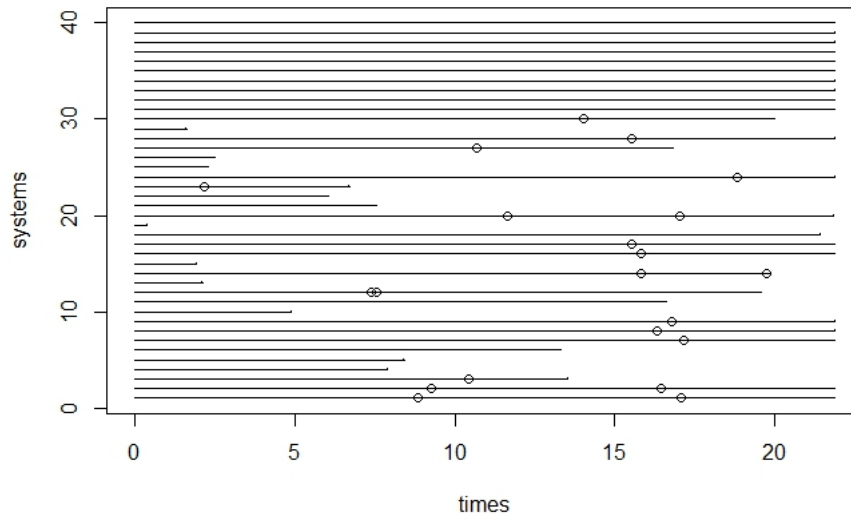


Figura 2.3: Sistemas e Falhas do Exemplo 3

executar milhares ou milhões de cálculos e repetições em segundos, mais informações podem ser obtidas em Davison [1997].

No caso da estimação *bootstrap*, são realizados os seguintes passos,

1. Calculam-se os Estimadores de Máxima Verossimilhança EMV de  $\beta$  e  $\theta$ , conforme seção (2.1);
2. Geram-se  $B$  amostras com o mesmo número de sistemas, tipos de truncamentos e  $\hat{\beta}$  e  $\hat{\theta}$  estimados no item anterior. A forma de simular amostras está descrita na seção (2.4).
3. Para cada uma das amostras  $B$ , são calculados Estimadores de Máxima Verossimilhança de  $\beta$  e  $\theta$ , ou seja,  $\hat{\beta}_1^*, \hat{\beta}_2^*, \dots, \hat{\beta}_B^*$  e  $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*$ ;
4. Os quantis das sequências do item anterior são usados para estimar um Intervalo de Confiança IC para os parâmetros  $\beta$  e  $\theta$ .

Veja a aplicação no caso do Exemplo 3.

```
> plp.ml(transformers,bootstrap=TRUE,n.boot=1000)
Parameter Estimate StdDev CI.low CI.up
```

1	shape	1.995	0.400	1.347	2.955
2	scale	24.366	2.798	19.455	30.516
3	shape.boot	1.995	0.431	1.450	2.846
4	scale.boot	24.366	4.263	20.857	34.072

Os valores são próximos aos encontrados sem a opção `bootstrap=TRUE`. A estimação pontual é a mesma, pois os EMV's de  $\beta$  e  $\theta$  são usados em ambos os casos (com ou sem a opção `bootstrap=TRUE`). A desvantagem desse método é o tempo de execução do comando, que pode ser demorado quando `n.boot` (número  $B$  de amostras) for muito grande. Por padrão, `n.boot=1000` na programação.

## 2.3 Funções dos parâmetros

Muitas vezes não estamos interessados diretamente na estimação dos parâmetros, mas em alguma função deles. Tipicamente, há três funções  $g(\beta, \theta)$  de maior interesse: a função de intensidade  $\lambda(T_0) = \frac{\beta}{\theta} \left(\frac{T_0}{\theta}\right)^{\beta-1}$ , a média  $\Lambda(T_0) = \left(\frac{T_0}{\theta}\right)^\beta$  e o tempo ótimo de manutenção  $\tau = \theta \left[\frac{C_M}{(\beta-1)C_R}\right]^{\frac{1}{\beta}}$ . Nesta seção, apresentamos as duas primeiras, pois a terceira será apresentada no Capítulo 5.

Para um certo valor de  $T_0$ , podemos estimar os valores de  $\lambda(T_0)$  conforme função (1.3), substituindo os valores de  $\beta$  e  $\theta$  estimados conforme seção 2.1. O mesmo é possível para  $\Lambda(T_0)$ , que no caso de PLP, é

$$\Lambda(T_0) = \int_0^{T_0} \lambda(x) dx = \left(\frac{T_0}{\theta}\right)^\beta. \quad (2.7)$$

Assim obtemos os estimadores pontuais para  $\lambda(T_0)$  e  $\Lambda(T_0)$ .

Para construir um estimador intervalar para estas funções, note que há uma relação entre a matriz de variâncias e covariâncias estimadas de parâmetros  $\phi$  e a variância estimada das funções dos parâmetros  $g(\phi)$ , explicado pelo Método Delta, com mais informações em (Casella and Berger [2010]). Escrito de forma matricial,

$$\hat{Var}_{\hat{\phi}} g(\hat{\phi}) \approx [\nabla g(\hat{\phi})'] \Sigma_{\hat{\phi}} [\nabla g(\hat{\phi})],$$

onde,  $\nabla g(\hat{\phi})$  é o gradiente de  $g$  avaliado em  $(\hat{\phi})$ ,  $\nabla g(\hat{\phi})'$  é matriz transposta de



$\nabla g(\hat{\phi})$  e  $\Sigma_{\hat{\phi}}$  é a matriz de variância e covariância de  $\hat{\phi}$ . Isto aplicado ao nosso caso, onde  $\phi = (\beta, \theta)$  e  $\hat{\phi} = (\hat{\beta}, \hat{\theta})$ , temos

$$\widehat{Var}_{\hat{\beta}, \hat{\theta}} g(\hat{\beta}, \hat{\theta}) \approx \begin{pmatrix} g'_{\beta}(\hat{\beta}, \hat{\theta}) & g'_{\theta}(\hat{\beta}, \hat{\theta}) \end{pmatrix} \begin{pmatrix} \widehat{var}(\hat{\beta}) & \widehat{cov}(\hat{\beta}, \hat{\theta}) \\ \widehat{cov}(\hat{\beta}, \hat{\theta}) & \widehat{var}(\hat{\theta}) \end{pmatrix} \begin{pmatrix} g'_{\beta}(\hat{\beta}, \hat{\theta}) \\ g'_{\theta}(\hat{\beta}, \hat{\theta}) \end{pmatrix}.$$

As estimativas para as funções de intensidade  $\lambda(T_0)$  e a média  $\Lambda(T_0)$  são obtidas pelo uso do comando `plp.ml`. Suas variâncias podem ser obtidas tanto pela matriz de variância e covariância de  $\hat{\beta}$  e  $\hat{\theta}$  (opção `log=FALSE`) quanto de  $\hat{\eta}$  e  $\hat{\zeta}$  (opção `log=TRUE` - padrão do comando). É preciso acrescentar os valores de  $T_0$  para os quais se deseja obter estimativas de  $\lambda(T_0)$  e  $\Lambda(T_0)$ . Temos a continuação dos Exemplos 2 e 3 para  $\lambda(T_0)$  e  $\Lambda(T_0)$  no tempo  $[0; 200]$  e  $[0; 30]$  respectivamente.

```
> plp.ml(AMSAA, mean.plp=200, intensity=200)
      Parameter Estimate StdDev CI.low  CI.up
1      shape      0.615   0.103  0.444  0.853
2      scale      3.525   2.558  0.850 14.614
3  Mean(200)     12.000   2.000  8.656 16.636
4 Intensity(200)  0.037   0.025  0.010  0.141
>
> plp.ml(transformers, mean.plp=30, intensity=30)
      Parameter Estimate StdDev CI.low  CI.up
1      shape      1.995   0.400  1.347  2.955
2      scale     24.366   2.798 19.455 30.516
3  Mean(30)      1.514   0.404  0.898  2.553
4 Intensity(30)  0.101   0.029  0.057  0.177
```

Para o Exemplo 2, a esperança pontual é de 12 falhas até o tempo  $T = 200$ . Os IC's são  $0,01 < \lambda(200) < 0,141$  e  $8,656 < \Lambda(200) < 16,636$ , com 95% de confiança. No Exemplo 3, a estimativa é de 1,5 falhas em 30 horas. O comando usa como padrão `log=TRUE`, logo, a matriz de variância e covariância de  $\hat{\eta}$  e  $\hat{\zeta}$  (`Ex.AMSAA$log.varcov` e `Ex.transformers$log.varcov`) foram as usadas para estimar  $\lambda(T_0)$  e  $\Lambda(T_0)$  e seus

intervalos de confiança.

As estimativas e os desvios padrão da função de intensidade  $\lambda(T_0)$  e função de média  $\Lambda(T_0)$  também podem ser obtidas com a opção `bootstrap=TRUE` no comando `plp.ml`. Neste caso, os valores  $\hat{\beta}_1^*, \hat{\beta}_2^*, \dots, \hat{\beta}_B^*$  e  $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*$  são substituídos nas equações (1.3) e (2.7), produzindo  $\hat{\lambda}(T_0)_1^*, \hat{\lambda}(T_0)_2^*, \dots, \hat{\lambda}(T_0)_B^*$  e  $\hat{\Lambda}(T_0)_1^*, \hat{\Lambda}(T_0)_2^*, \dots, \hat{\Lambda}(T_0)_B^*$ , de onde são calculados os desvios padrão e os IC's para  $\lambda(T_0)$  e  $\Lambda(T_0)$ . As estimativas pontuais são obtidas pela substituição de  $\hat{\lambda}(T_0) = \sum_{i=1}^B \frac{\hat{\lambda}(T_0)_i^*}{B}$  e  $\hat{\Lambda}(T_0) = \sum_{i=1}^B \frac{\hat{\Lambda}(T_0)_i^*}{B}$  dos dados observados nas equações (1.3) e (2.7). Observamos a aplicação da técnica no Exemplo 3 e  $T_0 = 30$ .

```
> plp.ml(transformers,bootstrap=TRUE,mean.plp=30,intensity=30)
```

	Parameter	Estimate	StdDev	CI.low	CI.up
1	shape	1.995	0.400	1.347	2.955
2	scale	24.366	2.798	19.455	30.516
3	shape.boot	1.995	0.436	1.416	2.761
4	scale.boot	24.366	4.078	21.002	33.825
5	Mean(30)	1.485	0.412	0.710	3.712
6	Intensity(30)	0.103	0.046	0.067	0.325

As estimativas pontuais quando `bootstrap=TRUE` não diferem muito dos EMV's. Os IC's se mostraram maiores do que os obtidos pelo Método Delta com a mesma confiança.

## 2.4 Simulação

Para a simulação de dados PPNH, uma opção é simular o tempo  $t_1$ , em seguida o tempo  $t_2 \mid t_1$  e assim por diante até o tempo  $t_n \mid t_{n-1}, \dots, t_2, t_1$ , conforme a função de distribuição em (1.4). Caso haja muitas falhas, esta simulação pode ser demorada e trabalhosa computacionalmente. Entretanto, utilizamos um método onde vários tempos de falhas podem ser amostrados em poucos passos, tornando a rotina computacional menos onerosa.

Em casos de truncamento por falha, utilizamos a função condicional a  $t_n$  que está definida abaixo,

$$f(t_1, t_2, \dots, t_{n-1} | t_n) = \frac{f(t_1, t_2, \dots, t_{n-1}, t_n)}{f(t_n)}. \quad (2.8)$$

O numerador desta fração está definido em (1.5). Para encontrar a expressão de  $f(t_n)$  também utilizaremos o fato de estarmos tratando de um Processo de Poisson,

$$f(t_n) = -\frac{d}{dt}S_{T_n}(t) = -\frac{d}{dt}P(T_n > t) = -\frac{d}{dt}P(N(t) < n) = -\frac{d}{dt} \sum_{i=0}^{n-1} \frac{[\Lambda(t)]^i e^{-\Lambda(t)}}{i!}.$$

Desenvolvendo a derivada chegamos a

$$f(t_n) = \lambda(t)e^{-\Lambda(t)} \sum_{i=0}^{n-1} \frac{1}{i!} ((\Lambda(t))^i - i \Lambda(t))$$

$i^{-1}$ .(2.8)Abrindo os termos da série na equação anterior, temos

$$f(t_n) = \frac{\lambda(t) [\Lambda(t)]^{n-1}}{\Gamma(n)} e^{-\Lambda(t)}. \quad (2.9)$$

Como resultado, podemos montar a densidade condicional a  $t_n$ ,

$$\begin{aligned} f(t_1, t_2, \dots, t_{n-1} | t_n) &= \frac{f(t_1, t_2, \dots, t_{n-1}, t_n)}{f(t_n)} \\ &= \frac{(\prod_{i=1}^n \lambda(t_i)) e^{-\int_0^{t_n} \lambda(x) dx}}{\frac{\lambda(t) [\Lambda(t)]^{n-1}}{\Gamma(n)} e^{-\Lambda(t)}} \\ &= \frac{(n-1)! (\prod_{i=1}^{n-1} \lambda(t_i))}{[\Lambda(t_n)]^{n-1}} \\ &= (n-1)! \prod_{i=1}^{n-1} \frac{\lambda(t_i)}{\Lambda(t_n)}. \end{aligned} \quad (2.10)$$

De acordo com o teorema 25 de Rigdon and Basu [2000] a função  $f(t_1, t_2, \dots, t_{n-1} | t_n)$  é uma função de densidade conjunta de  $(n-1)$  estatísticas de ordem de uma amostra com função de distribuição acumulada,

$$G(y) = \begin{cases} 0, & y \leq 0 \\ \Lambda(y)/\Lambda(t_n), & 0 < y \leq t_n \\ 1, & y > t_n \end{cases} \quad (2.11)$$

Em caso de truncamento temporal, utilizamos a função condicional  $f(t_1, t_2, \dots, t_n \mid N(T) = n)$ ,

$$f(t_1, t_2, \dots, t_n \mid N(T) = n) = \frac{f(t_1, t_2, \dots, t_{n-1}, t_n)}{f(n)}. \quad (2.12)$$

Temos o numerador da fração em (1.7). Precisamos encontrar  $f(n)$ ,

$$f(n) = P(N(0, t] = n) = \frac{e^{-\Lambda(t)} [\Lambda(t)]^n}{n!}. \quad (2.13)$$

Substituindo as expressões (1.7) e (2.13) na equação (2.12), chegamos a

$$f(t_1, t_2, t_3, \dots, t_n \mid N(T) = n) = (n!) \prod_{i=1}^n \frac{\lambda(t_i)}{\Lambda(T)}. \quad (2.14)$$

De acordo com o Teorema 26 de Rigdon and Basu [2000],  $f(t_1, t_2, \dots, t_n \mid N(T) = n)$  é função de densidade conjunta de  $(n)$  estatísticas de ordem de uma amostra com função de distribuição acumulada,

$$G(y) = \begin{cases} 0, & y \leq 0 \\ \Lambda(y)/\Lambda(T), & 0 < y \leq T \\ 1, & y > T \end{cases} \quad (2.15)$$

As funções acumuladas para o truncamento por falha 2.11 e para o truncamento temporal 2.15 são semelhantes, fazendo  $t_n = T$ . Este resultado permite que amostras PPNH sejam simuladas com base na Transformação Integral de Probabilidade, com mais detalhes em Casella and Berger [2010]. Para o caso específico de um PLP, podemos substituir  $\Lambda(y)$  por  $(y/\theta)^\beta$  e  $\Lambda(T)$  por  $(T/\theta)^\beta$ , conforme equação (2.7),

chegando ao seguinte resultado

$$G(y) = \begin{cases} 0, & y \leq 0 \\ (y/T)^\beta, & 0 < y \leq T \\ 1, & y > T \end{cases}$$

onde,

$$\left(\frac{y}{T}\right)^\beta = P(Y \leq y) = P\left(\frac{Y}{T} \leq \frac{y}{T}\right) = P\left(\left(\frac{Y}{T}\right)^\beta \leq \left(\frac{y}{T}\right)^\beta\right).$$

Sendo assim, podemos simular dados PLP, utilizando a distribuição uniforme  $U = (y/T)^\beta$  e isolando  $y$  ( $y = u^{\frac{1}{\beta}}T$ ).

Antes portanto, devemos simular um valor para  $t_n$  (tempo da última falha) em caso de truncamento por falha ou  $N(T) = n$  em caso de truncamento temporal. Simular  $t_n$  depende da função de intensidade  $\lambda(t)$  escolhida e para simular  $N(T) = n$ , usamos a Distribuição de Poisson para qualquer  $\lambda(t)$ . Em caso de truncamento por falha,  $t_n$  é distribuído conforme função (2.9). A função de distribuição de probabilidade de  $t_n$  dado que se trata de um PLP é

$$f(t_n) = \frac{1}{\Gamma(n)} \frac{\beta}{\theta} \left(\frac{t_n}{\theta}\right)^{n\beta-1} \exp\left(-\left(\frac{t_n}{\theta}\right)^\beta\right).$$

Esta não é uma distribuição de densidade de probabilidade convencional e pode ser difícil de simular  $t_n$  com ela diretamente. Entretanto, se recorremos a expressão  $H(y) = \left(\frac{y}{\theta}\right)^\beta$ , percebemos que esta expressão tem distribuição Gamma com parâmetros  $n$  e 1, conforme demonstração,

$$f_{H(y)}(y) = f(H^{-1}(y)) \left| \frac{\partial H^{-1}(y)}{\partial y} \right|.$$

Sabendo que  $H^{-1}(y) = \theta y^{1/\beta}$  e sua derivada em relação a  $y$  é  $\frac{\theta}{\beta} y^{1/\beta-1}$ , temos,

$$f_{H(y)}(y) = \frac{1}{\Gamma(n)} \frac{\beta}{\theta} \left(\frac{\theta y^{1/\beta}}{\theta}\right)^{n\beta-1} \exp\left(-\left(\frac{\theta y^{1/\beta}}{\theta}\right)^\beta\right) \frac{\beta}{\theta} y^{1/\beta-1}.$$

Simplificando a expressão chegamos a,

$$f_{H(y)}(y) = \frac{1}{\Gamma(n)} y^{n-1} \exp(-y),$$

ou seja,  $H(y)$  tem distribuição Gama  $(n, 1)$ . Como  $y = \theta(H(y))^{1/\beta}$ , podemos gerar um valor para  $H(y)$  e substituir  $\beta$  e  $\theta$  por valores escolhidos e assim temos valores simulados para  $y$ , que representa o valor de  $t_n$  (tempo da última falha do sistema).

Em se tratando de truncamento temporal, o caso é mais simples, pois  $n \sim \text{Poisson}((T/\theta)^\beta)$ , para valores de  $T$ ,  $\beta$  e  $\theta$  dados, conforme equação (2.13).

### 2.4.1 Programação para Simulação

Utilizando-se da teoria apresentada acima, funções foram criadas para a biblioteca *NHPPplp* para gerar amostras de um PPNH. Há três opções de simulação: amostras PLP para o caso de truncamento por falha; amostras PLP para truncamento temporal; e, amostras PPNH com outros tipos de função de intensidade para casos de truncamento temporal.

Para gerar uma amostra PLP com truncamento por falha, seguimos os seguintes passos,

1. Escolha os valores para o número de falhas  $n$ , para  $\beta$  e para  $\theta$  desejados.
2. Gera-se um valor para  $H(y)$  com distribuição Gama( $n, 1$ ). O programa *R* possui um comando (`rgamma`) que realiza esta função e que é utilizado internamente pela biblioteca.
3. Substitui-se  $H(y)$  simulado,  $\beta$  e  $\theta$  na equação  $t_n = \theta(H(y))^{1/\beta}$  para obtermos  $t_n$ , o valor da última falha;
4. Geram-se  $(n - 1)$  valores uniformes  $(0, 1)$ , que será armazenado no vetor  $u$ . O programa *R* possui um comando (`runif`) que realiza esta função.
5. Calcula-se  $t_i = u_{(i)}^{1/\beta} t_n$ , onde  $u_{(i)}$  é a  $i$ 'ésima estatística de ordem do vetor  $u$ .

Para gerar uma amostra PLP com truncamento temporal, os passos são similares ao caso com truncamento por falha,

1. Escolha os valores para o tempo  $T$  de truncamento do sistema, para  $\beta$  e para  $\theta$  desejados.
2. Gera-se um valor para  $n$  que possui distribuição Poisson( $(T/\theta)^\beta$ ). O programa  $R$  possui um comando (`rpois`) que realiza esta função.
3. Geram-se  $n$  valores uniformes  $(0, 1)$ , que será armazenado no vetor  $u$ .
4. Calcula-se  $t_i = u_{(i)}^{1/\beta} T$ , onde  $u_{(i)}$  é a  $i$ 'ésima estatística de ordem do vetor  $u$ .

O comando `plp.sim` gera as amostras desejadas. A saída no programa  $R$  para este comando é uma estrutura de dados como descrita na página (20), isto é, uma matriz, cuja 1ª coluna com tempos de falhas simulados, a 2ª coluna com indicação de censura (0 para censura e 1 para falha) e a 3ª coluna com o número do sistema correspondente. No *input* da função,  $TT$  é o vetor de tempos para sistemas com truncamento temporal e  $n$  é o vetor de valores para o número de falhas em casos com truncamento por falha.

**Exemplo 4.** *Caso simulado com 4 sistemas: um truncado em  $T = 40$ ; outro em  $T = 60$ ; um truncado por falha em  $n = 2$ ; e, outro em  $n = 3$ . Os valores para o parâmetro de forma  $\beta$  e o parâmetro de escala  $\theta$  escolhidos foram 2 e 35, respectivamente.*

```
> Ex.Sim<-plp.sim(beta=2,theta=35,TT=c(40,60),n=c(2,3))
```

```
> Ex.Sim
```

```
      times failures systems
1 39.770125         1       1
2 40.000000         0       1
3  6.887409         1       2
4 44.003938         1       2
5 44.176595         1       2
6 60.000000         0       2
7 16.399703         1       3
8 36.507612         1       3
9 29.936502         1       4
```

10	64.823013	1	4
11	71.975362	1	4

O exemplo tem 11 tempos de falhas ou censuras. São 4 sistemas: os dois primeiros são sistemas com truncamento temporal, por isso há indicação de censura na 2ª coluna em  $T = 40$  e  $T = 60$ . Os sistemas 3 e 4 são de truncamento por falha, por isso não há censuras.

Há também a possibilidade de simulação com uma função de intensidade  $\lambda(t)$  diferente do PLP, desde que sejam indicadas as expressões para a função média e a função inversa da função (2.15). Fornecida estas informações ao programa, a amostra é gerada. Nossa biblioteca oferece somente a opção de gerar amostras em caso de truncamento temporal.

**Exemplo 5.** *Um sistema truncado em  $T = 30$  com parâmetros  $\alpha = -4,5$  e  $\beta = 0,1$  para caso PPNH com função intensidade log-linear, ou seja,  $\lambda(t) = \exp(\alpha + \beta t)$ .*

De acordo com a equação (1.2), a média é

$$\Lambda(t) = \int_0^t e^{\alpha + \beta u} du = \frac{e^{\alpha + \beta t} - e^{\alpha}}{\beta}.$$

Substituindo as expressões  $\Lambda(t)$  e  $\Lambda(y)$  da log-linear em (2.15), temos

$$G(y) = e^{\beta(y-t)}.$$

Substituindo  $y$  pela função inversa  $G^{-1}(y)$  e  $G(y)$  por  $u$ , pois ela tem distribuição uniforme  $(0, 1)$ , temos

$$u = e^{\beta(G^{-1}(y)-t)}.$$

Isolando  $G^{-1}(y)$ ,



$$G^{-1}(y) = \frac{\log u}{\beta} + t.$$

A programação abaixo `f1` está representando a média  $\Lambda(t)$  e `inverse1`, a função inversa  $G^{-1}(y)$ . Isto gerou um sistema com 6 falhas e truncado em  $T = 50$ .

```
> f1<-function(TT,args){exp(args[1]+args[2]*TT)/args[2]} #media log-linear
> inverse1<-function(u,TT,args){log(u)/args[2]+TT} #funcao inversa
> Ex.Loglinear<-NHPP.sim(f1,inverse1,TT=50,c(-4.5,.1)) #simula casos log-linear
> Ex.Loglinear
```

```
      times censored system
[1,] 40.10427      1      1
[2,] 40.77490      1      1
[3,] 47.29283      1      1
[4,] 47.78450      1      1
[5,] 48.45553      1      1
[6,] 49.91833      1      1
[7,] 50.00000      0      1
```

# Capítulo 3

## Ajuste do Modelo

Um simples gráfico do tempo  $t$  contra o número acumulado de falhas  $N(t)$  pode nos dar uma ideia de modelagem possível, conforme figura (1.1). De acordo com o comportamento do gráfico, é possível perceber uma tendência do sistema, se os tempos entre as falhas estão cada vez menores, o gráfico ficará côncavo, ou seja, o sistema está deteriorando. Caso os tempos entre as falhas estão cada vez maiores, o gráfico se apresentará convexo e o sistema deve estar melhorando. Por fim, se o gráfico parece uma reta, uma modelagem que indique que o sistema nem deteriora nem melhora pode ser o mais adequado.

Além dessa sugestão simples de gráfico, vamos apresentar outros três tipos de gráficos. O Gráfico de Duane e o Gráfico do Tempo Total sob Teste (TTT) são usados para verificar o ajuste do modelo paramétrico escolhido em casos que envolvem um único sistema, com referência em Rigdon and Basu [2000]. O Gráfico Nelson-Aalen é a generalização do gráfico do tempo  $t$  contra o número acumulado de falhas  $N(t)$ , mencionado no parágrafo anterior, pois é útil para verificar o ajuste do modelo para estudos que possuem um ou mais sistemas, com referência em Borgan [2005] e aplicação em Gilardoni and Colosimo [2011].

Apresentaremos também dois Testes de Hipóteses que permitem a estimação de um p-valor que ajudará na decisão pela modelagem PPNH com função intensidade PLP dos dados ou na rejeição desta hipótese. O Teste de Kolmogorov-Smirnov, com mais detalhes em Siegel [1975], é aplicável para dados de um único sistema e o Teste de Crámer-von Mises, com mais detalhes em Crow [1975], para  $k \geq 1$  sistemas.

Outros testes de hipóteses que não serão trabalhados aqui podem ser vistos em Baker [1996].

### 3.1 Gráfico de Duane

O Gráfico de Duane surge pelo fato de que em um sistema modelado com PPNH e função de intensidade PLP

$$E(N(t)) = \Lambda(t) = \left(\frac{t}{\theta}\right)^\beta.$$

Logo,

$$\log E\left(\frac{N(t)}{t}\right) = (\beta - 1) \log t - \beta \log \theta. \quad (3.1)$$

Percebe-se pela equação (3.1) que  $\log(E(N(t)/t)$  e  $\log t$  possuem uma relação linear. Sendo assim, o gráfico de Duane consiste em plotar  $N(t_i)/t_i$  versus  $t_i$ , com  $i = 1, 2, \dots, n$  em escala logarítmica dupla. Caso a relação seja aproximadamente linear, o sistema poderia ser modelado como um PLP.

Se ao invés de  $t$  fixo, ele fosse aleatório, ou seja, se o sistema fosse de truncamento por falha, teríamos

$$\frac{N(t_i)}{t_i} = \frac{i}{t_i}.$$

Neste caso os pontos do gráfico de Duane teriam a seguinte esperança (Rigdon and Basu [2000], páginas 90 e 91),

$$\left(E(\log t_i), E\left(\log \frac{i}{t_i}\right)\right) = \left(\log \theta + \frac{\psi(i)}{\beta}, \log i - \log \theta - \frac{\psi(i)}{\beta}\right),$$

ou seja, não se espera linearidade para este caso e o gráfico pode ser até mesmo não monótono. Assim mesmo, Duane propôs em 1964 plotar os pontos

$$(\log t_i, \log (N(t_i)/t_i)),$$

inclusive a inclinação do gráfico pode ser usada para estimar  $(\beta - 1)$  e seu intercepto estima  $\beta \log \theta$ . A estimação de  $\beta$  neste caso é quase tão eficiente quanto o Estimador de Máxima Verossimilhança de  $\beta$ , conforme Molitor and Rigdon [1993].

**Exemplo 6.** *Simulação de um sistema com truncamento temporal no tempo  $T = 100$ , e parâmetros  $\beta = 2$  e  $\theta = 30$ .*

```
> Ex.plot<-plp.sim(beta=2,theta=30,TT=100)
```

```
> Ex.plot
```

	times	failures	systems
1	35.24656	1	1
2	44.91598	1	1
3	49.98605	1	1
4	57.79833	1	1
5	75.82595	1	1
6	77.16032	1	1
7	81.99765	1	1
8	90.76120	1	1
9	93.27601	1	1
10	99.24290	1	1
11	100.00000	0	1

```
> duane.plot(Ex.plot)
```

Pela figura (3.1), é possível notar uma relação aproximadamente linear entre  $\log(t_i)$  e  $\log(N(t_i)/t_i)$ . Isto significa que a modelagem PLP é adequada como escolha para representar os dados como esperávamos.

Vejamos agora a figura (3.2), que é a aplicação do gráfico de Duane aos dados do Exemplo 1, obtido pelo comando `duane.plot(gerador)`. A curva imaginária obtida

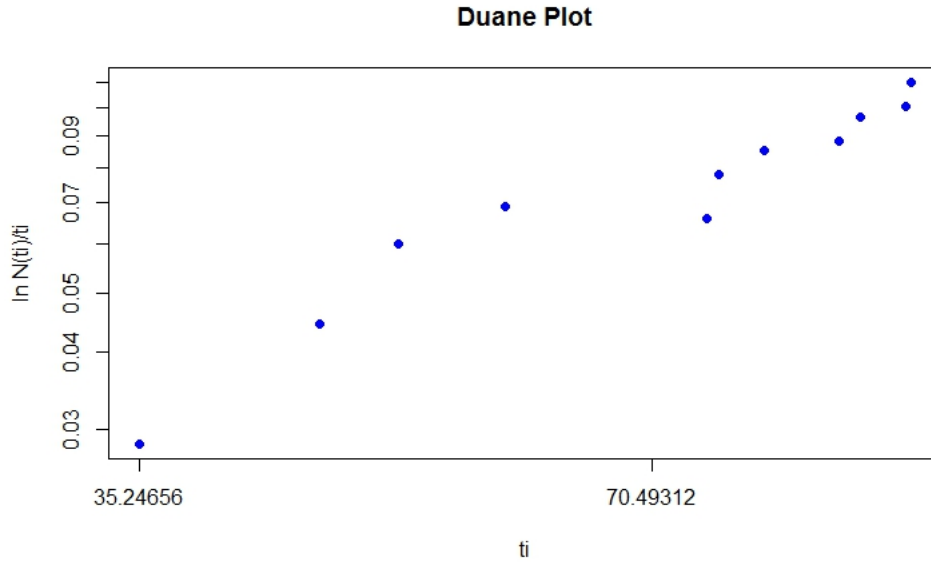


Figura 3.1: Duane Plot do Exemplo 6

é aproximadamente linear o que indica que a modelagem PLP pode ser adequada. Pode-se observar que a inclinação da reta imaginária formada no gráfico da figura (3.2) é decrescente. Isto significa que  $\hat{\beta} - 1 < 0$  ou  $\hat{\beta} < 1$ , ou seja, o sistema está melhorando com o tempo.

## 3.2 Gráfico TTT

O gráfico TTT é utilizado, assim como o gráfico de Duane, para um único sistema reparável sob um processo PPNH e com função de intensidade PLP. Ele foi desenvolvido por Barlow e Campo Barlow [1975] para sistemas não reparáveis e depois adaptado para este caso específico de sistema reparável.

Para entender o gráfico TTT, precisamos entender primeiramente como ele funciona para os sistemas não reparáveis. TTT significa Tempo Total sob Teste. Imagine que  $t_{(1)}, t_{(2)}, \dots, t_{(n)}$  representam os tempos de falhas em sistemas não-reparáveis, ou seja, em cada  $t_{(i)}$  pelo menos um sistema foi retirado de estudo. Tome  $(x_i = t_{(i)} - t_{(i-1)})$  e faça,

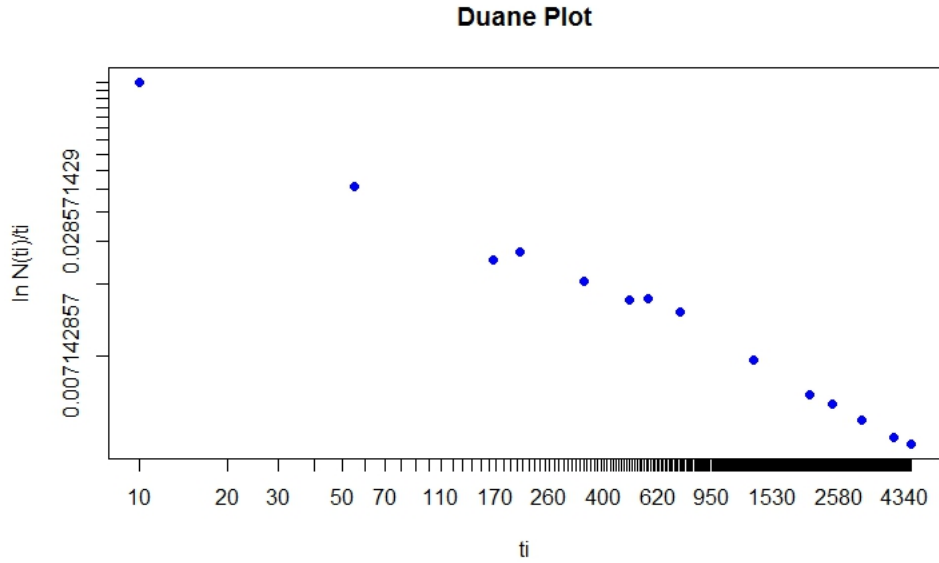


Figura 3.2: Duane Plot do Exemplo 1

$$H_0 = 0$$

$$H_j = nx_1 + (n-1)x_2 + \dots + (n-j+1)x_j, \quad j \geq 1,$$

sendo  $n$  o número total de sistemas e  $S_j$  o tempo total de todos os sistemas em teste até a  $j$ 'éssima falha. O gráfico TTT consiste em plotar  $(u_j = H_j/H_n)$  versus  $(j/n)$ , que converge para o ponto  $(\mu, \phi(\mu))$ , com

$$\phi(\mu) = \frac{1}{\mu} \int_0^{F^{-1}(\mu)} [1 - F(t)] dt,$$

onde  $F$  é a função de densidade acumulada dos  $t_{(i)}$ 's e  $\mu$  é a média da distribuição. Caso essa distribuição seja uma exponencial com parâmetro  $\mu$ , pode-se provar que  $\phi(\mu) = \mu$ , com  $0 < \mu < 1$ , ou seja, neste caso o gráfico convergiria para uma reta diagonal com extremos em  $(0, 0)$  e  $(1, 1)$ .

Para aplicar este mesmo gráfico em um sistema reparável modelado como PPNH e função de intensidade PLP e  $t_1, t_2, \dots, t_n$ , sendo os tempos de falha em um único sistema, é preciso efetuarmos a seguinte transformação,

$$w_j = -\log \frac{t_{n-j}}{t_n}.$$

Através do Lema 1 abaixo e sabendo que  $(t_{n-j}/t_n)^\beta$  é uniformemente distribuído no intervalo  $(0, 1)$ , de forma análoga ao mostrado em (2.4), podemos concluir que  $w_j$  é estatística de ordem com distribuição exponencial com média  $\beta$ . Logo, se utilizarmos de  $w_j$  no lugar de  $t_i$  do sistema não reparável, temos um gráfico TTT que tenderia a diagonal de um quadrado unitário caso o sistema pudesse ser modelado como PLP.

**Lema 1.** *Se  $U$  tem distribuição uniforme no intervalo  $(0, 1)$ , então  $X = -\theta \log U$  tem distribuição exponencial com média  $\theta$ .*

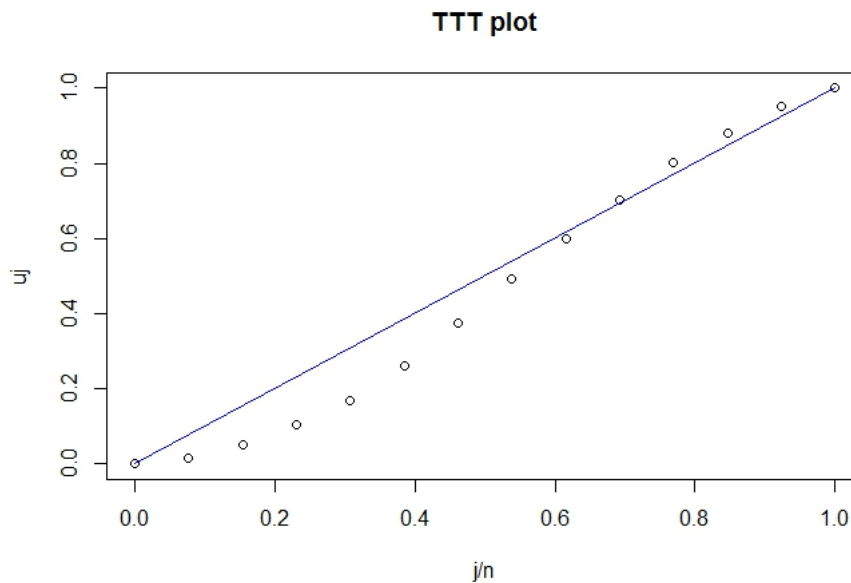


Figura 3.3: TTT Plot do Exemplo 1

As figuras (3.3) e (3.4) são os gráficos TTT aplicados aos dados dos Exemplos 1 e 6. Os comandos foram `ttt.plot(gerador)` e `ttt.plot(Ex.plot)`. Apesar de ser um julgamento subjetivo, quando se verifica o gráfico, notamos que não há grandes desvios dos pontos em relação a reta diagonal  $[(0, 0), (1, 1)]$ , logo a escolha da modelagem PPNH com função intensidade PLP parece ser adequada para ambos os exemplos.

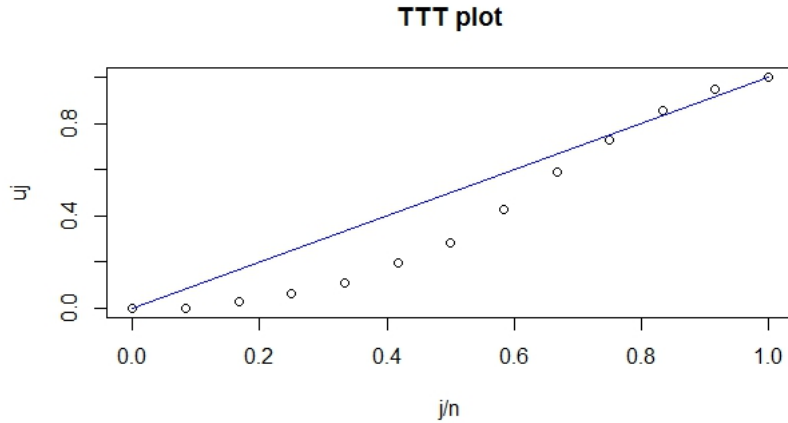


Figura 3.4: TTT Plot do Exemplo 6

### 3.3 Gráfico Nelson-Aalen

O Estimador de Nelson-Aalen é um estimador não paramétrico para a função acumulada da média  $\Lambda(t) = \int_0^t \lambda(x)dx$ . Ele não assume certas suposições, logo ele será um excelente instrumento que nos permitirá observar se o modelo paramétrico proposto é aceitável ou não, através de um método gráfico. Ele difere do gráfico de Duane e gráfico TTT por ser aplicável em casos onde há mais de um sistema em estudo, o que é muito comum na prática, desde que esses PPNH's possuam a mesma função de intensidade  $\lambda(t)$ .

Para encontrarmos o Estimador de Nelson-Aalen  $\bar{\Lambda}_{NA}(t)$ , primeiramente ordenamos os tempos de falha  $t_{(1)}, t_{(2)}, \dots, t_{(n)}$ , do menor para o maior de todos os sistemas. Em seguida atribuímos um valor  $k_i$ , indicando o número de sistemas em operação ou em observação para cada tempo  $t_{(i)}$ . Fazemos  $m_i$ , o número de falhas ocorridas em  $t_{(i)}$ . Em geral  $m_i = 1$ , mas pode acontecer de 2 ou mais falhas ocorrerem em  $t_{(i)}$ . O estimador é

$$\bar{\Lambda}_{NA}(t_{(i)}) = \bar{\Lambda}_{NA}(t_{(i-1)}) + \frac{m_i}{k_i},$$

onde  $\bar{\Lambda}_{NA}(t_0) = 0$ . O esquema da figura (3.5) ajuda a entender como  $k_i$ 's,  $m_i$ 's são calculados para se chegar aos Estimadores  $\bar{\Lambda}_{NA}(t_{(i)})$ 's, conforme mostra a saída do programa aplicado aos dados do Exemplo 4.



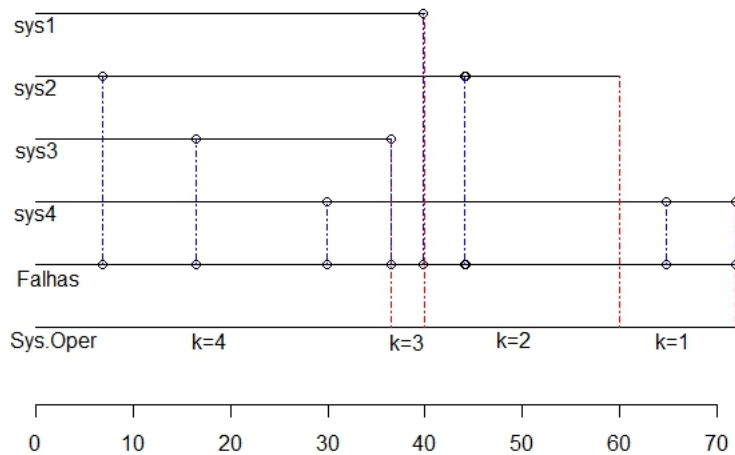


Figura 3.5: Esquema Nelson Aalen - Exemplo 4

```
> aalen.plot(Ex.Sim)
```

	Times	Estimate.N.Aalen
[1,]	6.887409	0.250000
[2,]	16.399703	0.500000
[3,]	29.936502	0.750000
[4,]	36.507612	1.000000
[5,]	39.770125	1.333333
[6,]	44.003938	1.833333
[7,]	44.176595	2.333333
[8,]	64.823013	3.333333
[9,]	71.975362	4.333333

Caso  $k = 1$ , ou seja, há um único sistema sob estudo, então o Estimador Nelson-Aalen ( $\bar{\Lambda}_{NA}(t_i)$ ) é igual ao número acumulado de falhas até  $t_i$  ( $N(t_i)$ ). Observe que para um único sistema, o gráfico seria como na figura (1.1).

Na biblioteca *NHPPplp*, o comando disponível é `aalen.plot` que plota os valores dos estimadores ( $\bar{\Lambda}_{NA}(t_i)$ ) versus os tempos ( $t'_i$ s). Através da análise da curva formada, é possível se ter uma ideia do valor de  $\beta$ . Um gráfico de ( $\bar{\Lambda}_{NA}(t)$ ) por ( $t$ ) seria convexo, se o sistema estivesse deteriorando, côncavo caso houvesse indício de

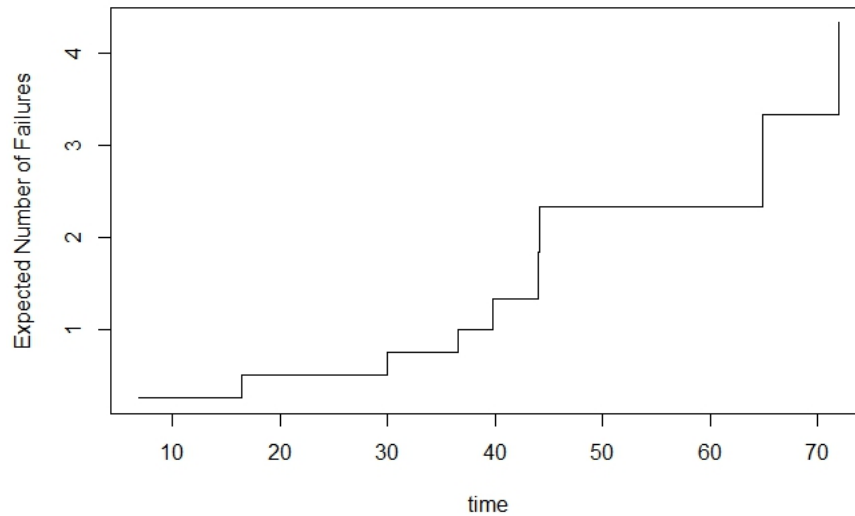


Figura 3.6: Nelson Aalen Plot do Exemplo 4

melhora com o tempo, e linear se não apresentasse tendência nem de melhora, nem de deterioramento. O comando permite comparar a média PLP com parâmetros  $\beta_0$  e  $\theta_0$ , que devem ser informados ao programa, com os estimadores Nelson-Aalen dos dados observados. Assim uma curva  $\hat{\Lambda}_{PLP}(t_i) \times t_i$  é formada sobre a curva do Estimador Nelson-Aalen, podendo ser feita a comparação gráfica entre elas.

A aplicação do gráfico Nelson-Aalen nos dados do Exemplo 2, conforme figura (3.7), mostra que um gráfico côncavo, indicando que  $\beta < 1$ , ou seja, o sistema melhora com o tempo, como já havíamos estimado. A segunda imagem da figura (3.7) possui duas curvas sobrepostas,  $\bar{\Lambda}_{NA}(t_i) \times t_i$  e  $\hat{\Lambda}_{PLP}(t_i) \times t_i$ , com  $\beta = 0.615$  e  $\theta = 3.525$  no caso do PLP. Além do gráfico, o comando também fornece os estimadores Nelson-Aalen para cada  $t_i$ . Caso  $\beta$  e  $\theta$  sejam especificados, as médias PLP também são fornecidas. No exemplo abaixo, restringimos aos 20 primeiros tempos para melhor visualização.

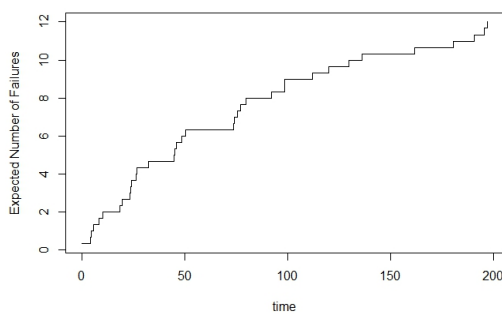
```
> aalen.plot(AMSAA)[1:10,]
      Times Estimate.N.Aalen
[1,]  0.1          0.3333333
[2,]  4.3          0.6666667
[3,]  4.4          1.0000000
```

```

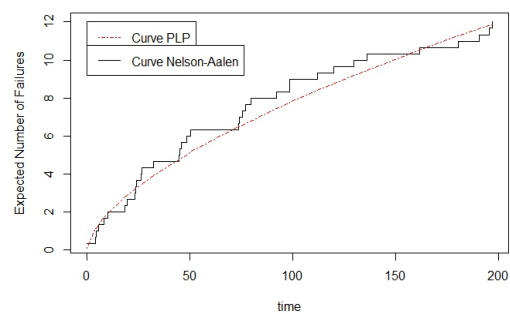
[4,] 5.6      1.3333333
[5,] 8.4      1.6666667
[6,] 10.2     2.0000000
[7,] 18.6     2.3333333
[8,] 19.5     2.6666667
[9,] 23.5     3.0000000
[10,] 23.8    3.3333333

> aalen.plot(AMSAA,beta=0.615,theta=3.525)[11:20,]
      Times Estimate.N.Aalen Estimate.PLP
[1,] 24.2      3.666667  3.269969
[2,] 26.4      4.000000  3.449718
[3,] 26.7      4.333333  3.473774
[4,] 32.5      4.666667  3.920175
[5,] 44.7      5.000000  4.769098
[6,] 45.1      5.333333  4.795299
[7,] 45.8      5.666667  4.840937
[8,] 48.4      6.000000  5.008147
[9,] 50.6      6.333333  5.146947
[10,] 73.6     6.666667  6.480776

```



(a)



(b)

Figura 3.7: Gráfico Nelson Aalen do Exemplo 2

Na continuação do Exemplo 3, pela primeira imagem da figura (3.8), é possível perceber que as falhas tendem a crescer com o tempo, devido ao formato convexo da

curva do gráfico, ou seja,  $\beta > 1$  e também, na segunda imagem, percebemos como os dados são bem explicados pela distribuição paramétrica PLP com  $\beta = 1,995$  e  $\theta = 24,366$ . Além da elaboração dos gráficos, o comando disponibilizou as estimativas Nelson-Aalen e médias PLP para cada tempo  $t_i$ .

```
> aalen.plot(transformers)[1:10,]
```

	Times	Estimate.N.Aalen
[1,]	2.168	0.02777778
[2,]	7.396	0.06003584
[3,]	7.541	0.09229391
[4,]	8.839	0.12800819
[5,]	9.280	0.16372248
[6,]	10.445	0.19943676
[7,]	10.668	0.23515105
[8,]	11.664	0.27086534
[9,]	14.041	0.30932687
[10,]	15.524	0.34778841

```
>
```

```
> aalen.plot(transformers,beta=1.995,theta=24.366)[1:10,]
```

	Times	Estimate.N.Aalen	Estimate.PLP
[1,]	2.168	0.02777778	0.008013158
[2,]	7.396	0.06003584	0.092686026
[3,]	7.541	0.09229391	0.096346552
[4,]	8.839	0.12800819	0.132263403
[5,]	9.280	0.16372248	0.145755065
[6,]	10.445	0.19943676	0.184538841
[7,]	10.668	0.23515105	0.192482407
[8,]	11.664	0.27086534	0.229999147
[9,]	14.041	0.30932687	0.332984892
[10,]	15.524	0.34778841	0.406834422

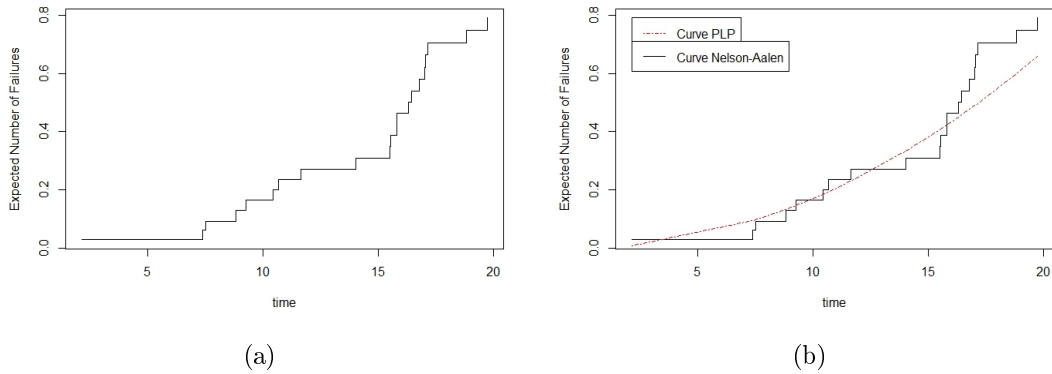


Figura 3.8: Gráfico Nelson Aalen do Exemplo 3

### 3.4 Teste de Kolmogorov-Smirnov

O Teste de Kolmogorov-Smirnov é um teste de aderência, isto é, consiste em analisar o grau de concordância entre a distribuição de um conjunto de valores de uma amostra e alguma distribuição teórica, com maiores explicações em Siegel [1975]. Para isto se utiliza da distribuição acumulada dada a distribuição teórica e compara com a distribuição acumulada observada. Este teste é um tipo específico do Teste de Lilliefors. Assim como os métodos gráficos, Duane e TTT, ele é aplicável somente para casos onde há um único sistema. Ele nos servirá para testar se o sistema pode ser modelado como PLP. Para entendermos como o teste foi desenvolvido segue uma explicação para o caso de truncamento por falha. Em se tratando de truncamento temporal o teste é realizado de forma análoga, fazendo  $T_n = T$ , com mais detalhes em Rigdon and Basu [2000].

Para analisar os dados, utilizamos a razão

$$R_i = \frac{\Lambda(t_i)}{\Lambda(t_n)}.$$

No caso de um PLP, temos

$$R_i = \frac{\Lambda(t_i)}{\Lambda(t_n)} = \frac{(t_i/\theta)^\beta}{(t_n/\theta)^\beta} = \left(\frac{t_i}{t_n}\right)^\beta.$$

Então, se aplicarmos o logaritmo na sequência dos dados  $R_i$ 's, temos

$$-\beta \log \frac{t_{n-1}}{t_n}, -\beta \log \frac{t_{n-2}}{t_n}, \dots, -\beta \log \frac{t_1}{t_n},$$

que é distribuído conforme uma estatística de ordem, com  $n - 1$  elementos, de uma Distribuição Exponencial com média 1, conforme Lema 1. Se dividirmos todos os elementos por  $\beta$ , temos

$$-\log \frac{t_{n-1}}{t_n}, -\log \frac{t_{n-2}}{t_n}, \dots, -\log \frac{t_1}{t_n}, \quad (3.2)$$

que chamaremos de  $Z_1, Z_2, \dots, Z_{n-1}$ . Eles são estatísticas de ordem com distribuição Exp  $(1/\beta)$ .

Agora com estes resultados é possível aplicar o teste, comparando a distribuição empírica transformada em (3.2) e a Distribuição Exponencial com média  $1/\beta$ . A hipótese nula é que os dados são distribuídos de acordo com um PLP, logo, a hipótese alternativa é que a distribuição não é um PLP.

Como dito anteriormente, o Teste de Kolmogorov-Smirnov se utiliza das Funções Acumuladas para efetuar as comparações. A Função Acumulada dos  $Z_i$ 's observados é

$$S_Z(z) = \begin{cases} 0, & z < z_1 \\ j/(n-1), & z_j \leq z < z_{j+1}, j = 1, 2, \dots, n-2 \\ 1, & z \geq z_{n-1} \end{cases}$$

No caso específico de truncamento por falha, perde-se um ponto na transformação dos tempos de falhas, agora temos  $n - 1$  observações. Vamos compará-la a uma distribuição teórica Exp $(1/\beta)$ , cuja a distribuição acumulada é

$$F_Z(z) = 1 - \exp(-z/\beta_0), \quad z > 0,$$

sendo que  $\beta_0$  é o valor do parâmetro  $\beta$  a ser testado. Conforme Rigdon and Basu [2000], um bom valor para  $\beta$  seria  $\beta_0^*$ , onde

$$\beta_0^* = \frac{\sum_{j=1}^{n-1} Z_j}{n-1}. \quad (3.3)$$

A estatística do Teste ( $D$ ) de Kolmogorov-Smirnov é a máxima diferença encontrada entre as duas distribuições. Assim, calculamos os valores  $Z_i$  em cada uma delas e obtemos:

$$D = \max(|F_Z(z_j) - S_Z(z_j)|, |F_Z(z_j) - S_Z(z_{j-1})|).$$

No nosso caso, quanto maior o valor de  $D$ , maior a evidência de que a modelagem PLP não é a adequada, por outro lado, quanto menor o valor de  $D$ , há maior evidência de que a modelagem PLP pode ser adequada para a situação. O programa  $R$  possui um comando que é usado internamente pela biblioteca. Ele fornece o  $p$ -valor do teste e depende do tamanho da amostra  $n$  e da estatística  $D$ . Lembrando que os Testes de Hipóteses nos fornecem somente evidências para uma tomada de decisão, mas não permite uma afirmação do tipo "verdade absoluta" para a hipótese aceita. Este cuidado deve ser ainda maior quando a amostra é pequena.

Se aplicarmos o teste no Exemplo 6, chegamos ao seguinte resultado:

```
> ks.fitplp(data=Ex.plot,beta=2)
One-sample Kolmogorov-Smirnov test
data:  Transf
D = 0.2201, p-value = 0.535
alternative hypothesis: two-sided
```

A estatística  $D = 0.2201$  gera um  $p$ -valor  $> 0.05$ , logo aceitamos a hipótese nula de que os dados podem ser modelados por um PLP com  $\beta = 2$ . O teste verifica somente os valores para  $\beta$ , sendo necessário a utilização de outros métodos se o interesse for verificar também valores para  $\theta$ . Se um valor para  $\beta$  não for indicado, o comando realiza os cálculos com  $\beta$  conforme (3.3).

Observe a continuação do Exemplo 1, onde aceitamos a hipótese nula ( $p$ -valor  $> 0,05$ ), ou seja, os dados podem ser modelados como PLP.

```
> ks.fitplp(gerador)
One-sample Kolmogorov-Smirnov test
data:  Transf
```

D = 0.1771, p-value = 0.7476

alternative hypothesis: two-sided

### 3.5 Teste de Cramér-von Mises

O teste de ajuste de Cramér-von Mises é útil para testar a modelagem tanto em caso de um único sistema como no caso de  $k \geq 2$  sistemas, com mais detalhes em Crow [1975]. Assumimos que cada sistema  $j$  é observado em  $(0, T_j)$ ,  $j = 1, \dots, k$  e  $k$  é o número total de sistemas. Se  $t_n = T_j$  (truncamento por falha), fazemos  $m_j = n_j - 1$  e se  $t_n < T_j$  (truncamento temporal), fazemos  $m_j = n_j$ . O tamanho da amostra será  $m = \sum_{j=1}^k m_j$ . Antes da aplicação do teste é necessário que a transformação dos  $m$  tempos de falha em  $z_i$ 's,

$$z_i = \frac{t_{ij}}{T_j},$$

sendo que  $t_{ij}$  é o tempo de falha  $i$  do  $j$ 'ésimo sistema e  $T_j$  é tempo total de observação do sistema  $j$ . Os valores  $z_i$ 's precisam ser colocados em ordem crescente.

Usando a estatística  $W_m^2$  com valor de  $m$  moderadamente grande, podemos testar se os dados seguem uma distribuição PPNH com função intensidade PLP para um  $\beta = \beta_0$  fixo, de acordo com Crow [1975].

$$W_m^2 = \frac{1}{12m} + \sum_{j=1}^m \left( z_j^{\beta_0} - \frac{2j-1}{2m} \right)^2.$$

Aqui estamos considerando que a hipótese nula  $H_0$  é que os dados seguem uma distribuição PPNH com função de intensidade PLP e  $\beta = \beta_0$  fixo, ou seja, para valores grandes de  $W_m^2$  rejeita-se  $H_0$  e valores pequenos de  $W_m^2$ , aceita-se  $H_0$ . Entretanto, normalmente estamos interessados em testar se os dados seguem uma distribuição PPNH com intensidade PLP mas não em um valor específico de  $\beta$ . Então, se substituirmos  $\beta_0$  por  $\bar{\beta}$  (estimador não viesado para  $\beta$  e obtidos dos dados observados),  $W_m^2$  torna-se uma estatística não paramétrica, ou seja, independe do verdadeiro valor de  $\beta$  para qualquer valor de  $m$  (conforme indicado em Crow [1975]), sendo que



$$\bar{\beta} = \frac{n-2}{n} \hat{\beta}$$

onde  $\hat{\beta}$  é o EMV dos dados.

Assim, podemos gerar valores para  $W_m^2$  através do Método de Monte Carlo. Isto permite estimar um p-valor do teste, rejeitando ou aceitando  $H_0$ , dependendo da significância escolhida.

Vejamos a aplicação nos Exemplos 2 e 4:

```
> cramer.fitplp(AMSAA)
Cramer-Von-Miser Test with df= 36
T= 0.06952911    p-value= 0.5290667
>
> cramer.fitplp(Ex.Sim)
Cramer-Von-Miser Test with df= 7
T= 0.04259778    p-value= 0.8324667
```

Conforme os p-valores (0,53 e 0,83, respectivamente), aceitamos  $H_0$ , ou seja, os dados seguem distribuição NHPP com funções intensidade PLP para ambos os casos como já era esperado.

# Capítulo 4

## Inferência Bayesiana

Diferente da Inferência Clássica, onde um parâmetro  $\phi$  é considerado um valor fixo mas desconhecido, a Inferência Bayesiana considera que o parâmetro  $\phi$  é uma variável aleatória não observada, que portanto possui uma distribuição de probabilidade própria.

As informações que possuímos sobre o parâmetro são atualizadas através da Teorema de Bayes, por isso o nome de Inferência Bayesiana,

$$p(\phi | x) = \frac{p(x, \phi)}{p(x)} = \frac{p(x | \phi)p(\phi)}{p(x)} = \frac{p(x | \phi)p(\phi)}{\int p(\phi, x)d\phi}.$$

Note que  $1/p(x)$ , que não depende de  $\phi$ , funciona como uma constante normalizadora de  $p(\phi | x)$ . Para cada valor de  $x$ , a função  $L(\phi; x) = p(x | \phi)$  fornece a verossimilhança de cada um dos possíveis valores de  $\phi$  (vetor dos parâmetros) enquanto  $p(\phi)$  é chamada de distribuição a priori de  $\phi$ . Com estas duas distribuições, chegamos a distribuição a posteriori de  $\phi$ ,  $p(\phi | x)$ , ou seja,

$$p(\phi | x) \propto p(\phi)L(\phi; x). \tag{4.1}$$

De posse da posteriori, podemos fazer as inferências necessárias sobre os parâmetros. Em nosso trabalho, os parâmetros de interesse são  $\phi = (\beta, \theta)$  e funções deles, tais como  $\lambda(T_0)$ ,  $\Lambda(T_0)$  e  $\tau = \theta \left( \frac{Rc}{\beta-1} \right)^{1/\beta}$ . Isto envolverá duas situações, a primeira é o estudo de um único sistema ( $k = 1$ ) e a segunda é quando ( $k \geq 2$ ) sistemas.

Supondo um único sistema ( $k = 1$ ), o artigo Oliveira et al. [2012] propõe fazer

a parametrização em termos de  $\beta$  e  $\eta$ , onde  $\beta$  é o parâmetro de forma (*shape*) e  $\eta = \Lambda(T) = (T/\theta)^\beta$ , ou seja,  $\eta$  é o número esperado de falhas no período  $(0; T]$ .

Usando a parametrização anterior, a verossimilhança (1.8) é

$$L(\beta, \eta) = c[\beta^n e^{-n\beta/\hat{\beta}}][\eta^n e^{-\eta}] \propto \gamma(\beta | n + 1, n/\hat{\beta})\gamma(\eta | n + 1, 1), \quad (4.2)$$

onde  $c = \prod_{j=1}^n t_j^{-1}$ ,  $\hat{\beta} = n/\sum_{j=1}^n \log(T/t_j)$  é o Estimador de Máxima Verossimilhança de  $\beta$  e  $\gamma(x|a, b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)}$  é função de densidade Gama com média  $\frac{a}{b}$  e variância  $\frac{a}{b^2}$ , sendo que  $a$  e  $b$  são chamados de hiperparâmetros.

Notamos na equação (4.2) que  $\beta$  e  $\eta$  são ortogonais. A atualização do conhecimento sobre  $\beta$  e  $\eta$  vai envolver somente a mudança nos hiperparâmetros se escolhermos uma distribuição conjugada a priori. A distribuição conjugada é tal que as distribuições a priori e a posteriori pertencem a mesma classe. Neste caso, a priori é a multiplicação de duas Gamas, representada por  $\pi(\beta, \eta)$ ,

$$\pi(\beta, \eta) \propto \gamma(\beta | a_\beta, b_\beta)\gamma(\eta | a_\eta, b_\eta). \quad (4.3)$$

De acordo com a equação (4.1), a posteriori será proporcional a multiplicação de ambas as funções de densidade (4.2) e (4.3),

$$\pi(\beta, \eta | t_1, \dots, t_n, T) \propto L(\beta, \eta)\pi(\beta, \eta) \propto \gamma(\beta | a_\beta + n, b_\beta + n/\hat{\beta})\gamma(\eta | a_\eta + n, b_\eta + 1).$$

As distribuições marginais a posteriori de  $\beta$  e  $\eta$  são  $\gamma(\beta | a_\beta + n, b_\beta + n/\hat{\beta})$  e  $\gamma(\eta | a_\eta + n, b_\eta + 1)$  respectivamente. Assim, podemos obter estimações pontuais, variâncias e os Intervalos de Credibilidade ICr para  $\beta$  e  $\theta$ . No resto deste trabalho assumiremos sempre funções de perda quadrática, onde o estimador pontual de um parâmetro  $\phi$  é dado por  $E(\phi | D)$ , sendo  $D$  os dados observados. Por exemplo  $\hat{\eta} = E(\eta | D) = \frac{a_\eta + n}{b_\eta + 1}$ . Um ICr com  $100(1 - \alpha)\%$  de confiança para  $\phi$  é tal que  $P(\phi | D \in ICr) \geq 1 - \alpha$ . O ICr tem papel semelhante ao Intervalo de Confiança que utilizamos na Inferência Clássica.

Quando estamos tratando de ( $k \geq 2$ ) sistemas, a verossimilhança é mais com-

plexa. Neste caso, a parametrização em  $\beta$  e  $\eta$  é tal que  $\beta$  é o parâmetro de forma (*shape*) e  $\eta = \sum_{i=1}^k (T_i/\theta)^\beta$ , com  $k$  sendo o número de sistemas. Mantemos a mesma priori e obtemos a posteriori abaixo, conforme artigo Oliveira et al. [2012],

$$\pi(\beta, \eta \mid D) \propto \gamma(\beta \mid a_\beta + n, b_\beta + n/\hat{\beta}) \times \gamma(\eta \mid a_\eta + n, b_\eta + 1) \times e^{nF(\beta)},$$

onde  $D$  representa os dados observados e  $F(\beta)$  é tal que,

$$F(\hat{\beta}) - F(\beta) = KL \left( \left( \frac{T_1^{\hat{\beta}}}{\sum_{i=1}^K T_i^{\hat{\beta}}}, \dots, \frac{T_K^{\hat{\beta}}}{\sum_{i=1}^K T_i^{\hat{\beta}}} \right) \parallel \left( \frac{T_1^\beta}{\sum_{i=1}^K T_i^\beta}, \dots, \frac{T_K^\beta}{\sum_{i=1}^K T_i^\beta} \right) \right) \geq 0,$$

onde  $KL((p_1, \dots, p_k) \parallel (q_1, \dots, q_k)) = \sum_{i=1}^k p_i \log p_i / q_i$  é a divergência de Kullback-Leibler.

Desta forma, a distribuição marginal a posteriori de  $\eta$  é  $\gamma(\eta \mid a_\eta + n, b_\eta + 1)$  e podemos obter facilmente a estimação pontual, a variância e o ICr(1 -  $\alpha$ )100%. Entretanto, a distribuição marginal a posteriori de  $\beta$  é proporcional a  $\gamma(\beta \mid a_\beta + n, b_\beta + n/\hat{\beta})e^{nF(\beta)}$  e a inferência sobre  $\beta$  precisará lidar com esta última expressão,  $e^{nF(\beta)}$ , como veremos na seção 4.1.

A escolha da distribuição a priori depende da informação que se tem antes do estudo. Por isso, também é estudado pela Inferência Bayesiana, prioris que são consideradas neutras ou não informativas, pois, em algumas situações práticas, há pouca ou nenhuma informação sobre os parâmetros. Desta forma, houve diferentes propostas na literatura. Algumas são indicadas em Oliveira et al. [2012] como  $\pi(\beta, \eta) \propto (\beta\eta)^{-1}$  ou  $\pi(\beta, \eta) \propto \beta^{-2}\eta^{-1}$  e a generalização das anteriores  $\pi(\beta, \theta) \propto \beta^{-\delta-1}\eta^{-1}$  com ( $\delta < n$ ). Este último caso, equivale a fazermos  $a_\beta = -\delta$  e  $b_\beta = a_\eta = b_\eta = 0$  na equação (4.3).

## 4.1 Estimação Bayesiana dos Parâmetros

A inferência sobre  $\beta$  e sobre  $\eta$  quando  $k = 1$  é relativamente simples, pois suas distribuições posteriores são Gamas conhecidas. Fazendo as escolhas apropriadas das densidades a priori e usando os dados observados, podemos atualizar nosso conhecimento sobre  $\beta$  e  $\eta$ . A biblioteca disponibiliza o comando `bayes.plp`. As opções `prior.beta` e `prior.eta` são os vetores  $(a_\beta, b_\beta)$  e  $(a_\eta, b_\eta)$  respectivamente e `data` deve armazenar os dados observados.

Vejam a aplicação da técnica no exemplo 1,

```
> bayes.plp(prior.beta=c(-2,0),prior.eta=c(0,0),data=gerador)
```

	Parameter	Estimate	HPD.CI.low	HPD.CI.up
1	beta'shape'	0.414	0.196	0.653
2	eta	14.000	7.145	21.487
3	theta'scale'	20.655	0.000	85.485

Utilizamos prioris não informativas, sendo  $a_\beta = -2$  e  $b_\beta = a_\eta = b_\eta = 0$  e aplicado na função (4.3). O resultado é próximo ao encontrado na Inferência Clássica para as estimações pontuais, como vemos na página 22. Os ICr's muito menores do que os IC's.

Como a distribuição Gama não é simétrica, intervalos nos quantis  $\alpha/2$  e  $1-\alpha/2$  da distribuição não são os de menor comprimento. O intervalo de menor comprimento é chamado de Intervalo de Credibilidade com Maior Densidade a Posteriori (MDP) ou *Highest Posterior Density* (HPD) em inglês. Usando a amostra simulada da distribuição a posteriori, a nossa biblioteca no *R* estima Intervalos de Credibilidade HPD aproximados para os parâmetros e suas funções.

Quando ( $k \geq 2$ ) sistemas, a inferência sobre  $\beta$  envolve um Algoritmo de Aceitação/Rejeição (páginas 226 e 227, Casella and Berger [2010]) com os seguintes passos,

1. Gere  $\bar{\beta} \sim Gama(a_\beta + n, b_\beta + n/\hat{\beta})$  e  $u \sim Uniforme(0, 1)$ ;
2. Se  $u \leq \exp(n[F(\beta) - F(\hat{\beta})])$ , aceita-se  $\bar{\beta}$ , caso contrário ele é rejeitado.
3. Repita os 1º e 2º procedimentos até obter  $\bar{\beta}_1, \dots, \bar{\beta}_m$ , sendo  $m$ , o tamanho da amostra da distribuição a posteriori.

A média  $\frac{1}{m} \sum_{i=1}^m \bar{\beta}_i$  se aproxima de  $E(\beta|D)$  para  $m$  grande. Os quantis da sequência  $\bar{\beta}_1, \dots, \bar{\beta}_m$  são usados para o cálculo do Intervalo de Credibilidade  $(1 - \alpha)100\%$ . A amostragem da distribuição de  $\eta$  é mais simples pois  $\eta|D \sim \gamma(a_\eta + n, b_\eta + 1)$

Vejamos a aplicação nos exemplos 2 e 3.

```
> bayes.plp(prior.beta=c(-1,0),prior.eta=c(0,0),AMSAA)
```

	Parameter	Estimate	HPD.CI.low	HPD.CI.up
1	beta'shape'	0.598	0.407	0.792
2	eta	36.000	24.676	47.994
3	theta'scale'	3.712	0.103	8.812

```
>
```

```
> bayes.plp(prior.beta=c(20,10),prior.eta=c(10,.5),data=transformers)
```

	Parameter	Estimate	HPD.CI.low	HPD.CI.up
1	beta'shape'	1.997	1.425	2.646
2	eta	20.667	13.654	28.059
3	theta'scale'	25.051	20.554	30.157

Os resultados também são próximos aos encontrados pela Inferência Clássica. Para o exemplo 3, utilizamos uma priori mais informativa, conforme informação obtida em Oliveira et al. [2012].

## 4.2 Estimação das funções dos parâmetros

Baseado na inferência feita pelas posteriores encontradas, podemos fazer também a inferência de  $G(\beta, \eta)$ , funções de  $\beta$  e  $\eta$ . Estamos interessados em quatro funções  $G(\beta, \eta)$ , sendo que a função relacionado a Política de Manutenção, trataremos no capítulo 5. Quando estamos tratando de um único sistema ( $k = 1$ ), as outras três funções são

$$\begin{aligned}
\theta &= \frac{T}{\eta^{\frac{1}{\beta}}} && \text{Parâmetro de Escala} \\
\lambda(T_0) &= \frac{\beta}{\theta} \left(\frac{T_0}{\theta}\right)^{\beta-1} = \frac{\beta (T_0)^{\beta-1}}{\left(\frac{T}{\eta^{1/\beta}}\right)^\beta} = \frac{\beta\eta}{T} \left(\frac{T_0}{T}\right)^{\beta-1} && \text{Função de Intensidade do } PLP \\
\Lambda(T_0) &= \left(\frac{T_0}{\theta}\right)^\beta = \left(\frac{T_0}{T}\right)^\beta \left(\frac{T}{\theta}\right)^\beta = \left(\frac{T_0}{T}\right)^\beta \eta && \text{Média do } PLP.
\end{aligned}$$

Se  $k \geq 2$  sistemas, as funções  $G(\beta, \eta)$  de interesse são expressas da seguinte forma,

$$\begin{aligned}
\theta &= \left(\frac{\sum_{i=1}^k T_i^\beta}{\sum_{i=1}^k n_i}\right)^{\frac{1}{\beta}} && \text{Parâmetro de Escala} \\
\lambda(T_0) &= \frac{\beta\eta T_0^{\beta-1}}{\sum_{i=1}^k T_i^\beta} && \text{Função de Intensidade } PLP \\
\Lambda(T_0) &= \frac{\eta T_0^\beta}{\sum_{i=1}^k T_i^\beta} && \text{Média do } PLP
\end{aligned}$$

Como a função de densidade de  $\theta$ ,  $\lambda(T_0)$  e  $\Lambda(T_0)$  são complicados de se obter analiticamente, utilizamos o método Monte Carlo (página 240, Casella and Berger [2010]) para estimá-los, conforme os seguintes passos,

1. Gere  $\bar{\beta}_1, \dots, \bar{\beta}_m$  e  $\bar{\eta}_1, \dots, \bar{\eta}_m$ , que possuem funções de densidade conforme o número de sistemas em estudo;
2. Substitua nas fórmulas  $G(\beta, \eta)$ , obtendo  $\bar{\theta}_1, \dots, \bar{\theta}_m, \bar{\lambda}(T_0)_1, \dots, \bar{\lambda}(T_0)_m$  e  $\bar{\Lambda}(T_0)_1, \dots, \bar{\Lambda}(T_0)_m$ .

A técnica gera boas aproximações para os valores esperados das funções  $G(\beta, \eta)$  com  $m$  grande, fazendo  $E(\theta|D) \simeq \frac{1}{m} \sum_{i=1}^m \bar{\theta}_i$ ,  $E(\lambda(T_0)|D) \simeq \frac{1}{m} \sum_{i=1}^m \bar{\lambda}_i(T_0)$  e  $E(\Lambda(T_0)|D) \simeq \frac{1}{m} \sum_{i=1}^m \bar{\Lambda}_i(T_0)$ , sendo  $D$  os dados observados. O Intervalo de Credibilidade HPD pode ser obtido pelos quantis das sequências  $\bar{\theta}_1, \dots, \bar{\theta}_m, \bar{\lambda}_1(T_0), \dots, \bar{\lambda}_m(T_0)$  e  $\bar{\Lambda}_1(T_0), \dots, \bar{\Lambda}_m(T_0)$ .

Aplicando a teoria aos exemplos 2 e 3, chegamos ao seguinte resultado:

```

> bayes.plp(prior.beta=c(-1,0),prior.eta=c(0,0),AMSAA,
>           intensity=200,mean.b=200)
      Parameter Estimate HPD.CI.low HPD.CI.up
1  beta'shape'    0.597    0.409    0.804
2           eta   36.000   24.676   47.994
3  theta'scale'    3.709    0.107    8.905
4  Mean.plp(200)  12.008    8.111   15.953
5  Intensity(200)  0.108    0.059    0.158
>
> bayes.plp(prior.beta=c(20,10),prior.eta=c(10,.5),data=transformers,
>           intensity=30,mean.b=30)
      Parameter Estimate HPD.CI.low HPD.CI.up
1  beta'shape'    1.992    1.412    2.605
2           eta   20.667   13.654   28.059
3  theta'scale'   25.053   20.780   30.577
4  Mean.plp(30)    1.494    0.907    2.125
5  Intensity(30)    1.370    0.812    2.046

```

Percebe-se que é utilizado o mesmo comando `bayes.plp`, acrescentando as opções `intensity` e `mean.b`. Os valores encontrados para exemplo 3 estão bem próximas aos encontrados pela Inferência Clássica na seção 2.3.



# Capítulo 5

## Política de Manutenção

Um dos principais interesses da análise da confiabilidade de sistemas reparáveis é desenvolver uma Política de Manutenção. Veja por exemplo Baracho [2001], Gilardoni and Colosimo [2007] e Oliveira et al. [2012]. Em muitas situações práticas, o custo com reparos é alto e torna-se mais eficiente realizar uma manutenção preventiva, denominada apenas por "manutenção". Ela é realizada em períodos determinados que chamaremos de  $\tau$ . O objetivo deste capítulo é desenvolver uma rotina para encontrar um  $\tau$  ótimo, a fim de evitar custo com reparos e prolongar a vida útil do sistema.

Neste trabalho, consideramos a manutenção **perfeita**, isto é, o sistema volta a estar "como novo" após a manutenção (*as good as new*- AGAN), chamado de Processo de Renovação. Sendo assim, estamos tratando de dois processos dentro de um único sistema, o Processo de Poisson e o Processo de Renovação. Na verdade, após a manutenção perfeita, consideramos como se um novo sistema tivesse começado, ou seja, um único sistema será tratado como vários sistemas independentes e identicamente distribuídos, como mostra a figura (5.1). Dentro desta lógica, a manutenção faz sentido para os sistemas que deterioram com o tempo, isto é, quando  $\beta > 1$ . Neste período entre as manutenções perfeitas, o Processo é de Poisson, pois os reparos são mínimos.

Supomos que as manutenções acontecerão em intervalos de mesmo tamanho,  $\tau$ . No intervalo total do sistema  $(0; T)$  ocorrerão  $m$  manutenções em  $(\tau; 2\tau; 3\tau; \dots; (m-1)\tau; (m)\tau)$ , onde  $m = \lfloor T/\tau \rfloor$ , ou seja, a parte inteira da divisão  $T/\tau$ .

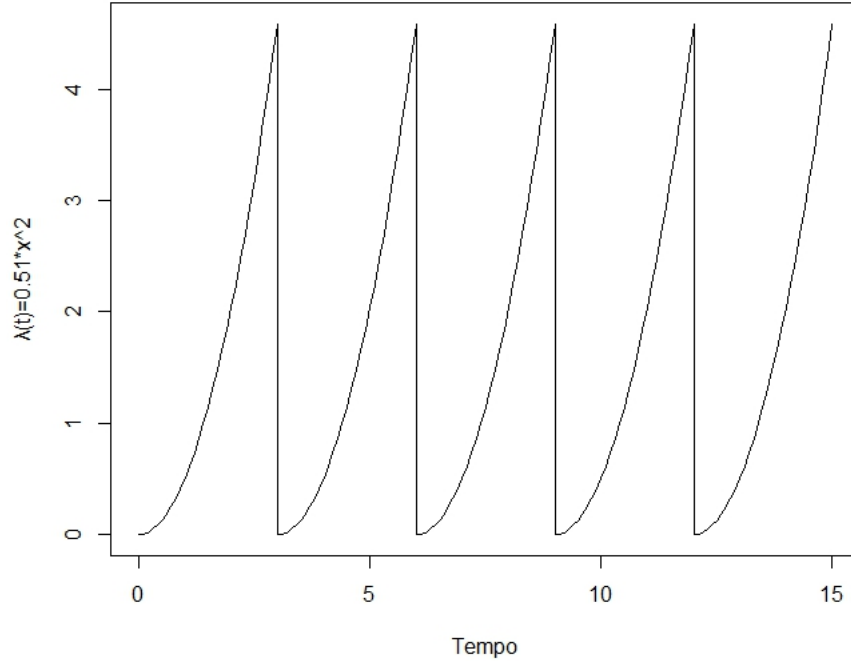


Figura 5.1: Função Intensidade com Manutenções  $\tau = 3$

Introduzimos o conceito de **custo**, que são os gastos para realizar a manutenção  $C_M$  ou o reparo  $C_R$  no sistema. Supomos em nosso modelo que a razão entre os custos  $Rc = \frac{C_M}{C_R}$  é constante no tempo. O custo total em um intervalo  $(a; b]$  é representado por  $C_T(a; b]$ . Supondo que este intervalo se trata de  $((i - 1)\tau; i\tau]$  para qualquer valor inteiro de  $i$  entre  $[1; m]$  temos que

$$C_T((i - 1)\tau; i\tau] = C_M + C_R \int_0^\tau \lambda(x) dx.$$

Ao custo total  $C_T(0; T]$ , acrescentamos também as possíveis falhas que ocorriam entre a última manutenção e o tempo final do estudo  $T$ ,

$$C_T(0; T] = m\{C_M + C_R \int_0^\tau \lambda(x) dx\} + C_R E(N(m\tau; T)).$$

O nosso interesse será não diretamente no custo total, já que o custo cresce quando o tempo cresce, mas no custo por unidade de tempo,  $H(\tau)$ ,

$$\begin{aligned}
H(\tau) &= \lim_{T \rightarrow \infty} \frac{C_{(0,T]}(\tau)}{T} \\
&= \frac{1}{\tau} \left[ C_M + C_R \int_0^\tau \lambda(u) du \right] \\
&= \frac{1}{\tau} [C_M + C_R \Lambda(\tau)].
\end{aligned} \tag{5.1}$$

Tem-se então a função a ser minimizada, para isto basta igualar a primeira derivada em relação a  $\tau$  a zero. Logo,  $\tau$  deve satisfazer a seguinte equação

$$\frac{C_M}{C_R} = \tau \lambda(\tau) - \Lambda(\tau).$$

Substituindo  $\lambda(t)$  pela função de intensidade PLP e isolando o valor  $\tau$ , encontramos

$$\tau = \theta \left[ \frac{C_M}{(\beta - 1)C_R} \right]^{\frac{1}{\beta}}. \tag{5.2}$$

Assim, podemos estimar  $\tau$ , que depende dos valores de  $\beta$ ,  $\theta$  e da Razão de Custos  $Rc = \frac{C_M}{C_R}$ , uma constante.

No Capítulo 2, chegamos às EMV's para  $\beta$  e  $\theta$ . Nesta seção, estamos interessados na estimação de  $\tau$  que é uma função de  $\beta$  e  $\theta$ . Então, substituindo  $\beta$  e  $\theta$  pelos valores  $\hat{\beta}$  e  $\hat{\theta}$  na função (5.2), temos uma estimativa pontual  $\hat{\tau}$ . Para a estimação intervalar podemos utilizar o Método Delta, onde o intervalo de confiança com  $100(1 - \alpha)\%$  para  $\tau$  é

$$\hat{\tau} - z_{1-\alpha/2} \hat{\sigma}_{\hat{\tau}} < \tau < \hat{\tau} + z_{\alpha/2} \hat{\sigma}_{\hat{\tau}},$$

com  $\hat{\sigma}_{\hat{\tau}} = \sqrt{[\nabla \tau(\hat{\beta}, \hat{\theta})]' \Sigma [\nabla \tau(\hat{\beta}, \hat{\theta})]}$ , onde  $\nabla \tau(\hat{\beta}, \hat{\theta})$  é o gradiente da função (5.2) avaliado em  $(\hat{\beta}, \hat{\theta})$ ,  $\Sigma$  é menos o inverso da matriz hessiana da função log-verossimilhança nos pontos  $\hat{\beta}$  e  $\hat{\theta}$  definida na seção (1.3) e  $z_{\alpha/2}$  é um ponto da curva gaussiana para um  $\alpha$ , grau de significância determinado.

Por definição  $\tau > 0$ , logo uma forma mais robusta de estimar  $\tau$  é utilizar a parametrização  $\tau = e^\vartheta$ . Colocando  $e^\vartheta$  em função de  $\eta$  e  $\zeta$ , parametrizações de  $\beta$  e

$\theta$ , como feito na seção (2.1), temos que

$$\begin{aligned}\tau &= \theta \left[ \frac{Rc}{(\beta - 1)} \right]^{\frac{1}{\beta}} \\ e^{\vartheta} &= e^{\zeta} \left[ \frac{Rc}{(e^{\eta} - 1)} \right]^{\frac{1}{e^{\eta}}} \\ \vartheta &= \zeta + \frac{\log Rc - \log(e^{\eta} - 1)}{e^{\eta}}.\end{aligned}$$

Pela propriedade de invariância do EMV,  $\hat{\tau} = e^{\hat{\vartheta}}$  e seu IC,

$$e^{\hat{\vartheta} - z_{\alpha/2} \hat{\sigma}_{\hat{\vartheta}}} \leq \hat{\tau} \leq e^{\hat{\vartheta} + z_{\alpha/2} \hat{\sigma}_{\hat{\vartheta}}}.$$

O método *bootstrap* também pode ser utilizado para estimação de um IC para  $\tau$  com os seguintes passos,

1. Geram-se  $B$  amostras com parâmetros  $\hat{\beta}$  e  $\hat{\theta}$ .
2. Calculam-se os Estimadores de Máxima Verossimilhança de  $\beta$  e  $\theta$  conforme seção (2.2), ou seja,  $\hat{\beta}_1^*, \hat{\beta}_2^*, \dots, \hat{\beta}_B^*$  e  $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*$ ;
3. Substituem-se os valores do item anterior na equação (5.2), obtendo  $\hat{\tau}_1^*, \hat{\tau}_2^*, \dots, \hat{\tau}_B^*$ ;
4. Os quantis da sequência  $\hat{\tau}_i^*$ 's são usados para estimar um Intervalo de Confiança IC para  $\tau$ .

No caso da Inferência Bayesiana, deveríamos encontrar uma distribuição a posteriori para  $\tau$ . Sabemos que  $\tau$  pode ser estimado através da função (5.2) e como a sua função de distribuição pode ser muito complicada, usaremos o método Monte Carlo, assim como foi feito para as funções  $\lambda(t)$  e  $\Lambda(t)$  na seção (4.2). Quantis do conjunto de estimativas  $\bar{\tau}_i$  produzidos na técnica, servirão para criar um Intervalo de Credibilidade  $100(1 - \alpha)$ . A Inferência Bayesiana é particularmente útil para casos onde os dados fornecem  $\hat{\beta} < 1$  e mesmo assim deseja-se estimar um  $\tau$ , porque podemos restringir o valor de  $\beta > 1$  através da distribuição a priori.

## 5.1 Estimação de $\tau$

A biblioteca *NHPPplp* possibilita a estimação de  $\tau$  com o uso do comando `plp.ml` para o caso da Inferência Clássica, desde que os dados observados e uma Razão de Custos ( $Rc$ ) sejam fornecidos.

Seguindo o Exemplo 3, temos  $Rc = C_M/C_R = 1/15$ . Obtemos o seguinte resultado,

```
> plp.ml(transformers, ratio.cust=1/15, conf=.90)
      Parameter Estimate StdDev CI.low  CI.up
1      shape      1.995   0.400  1.435  2.774
2      scale     24.366   2.798 20.172 29.431
3 Optimal.Maint(1/15)  6.286   0.722  5.203  7.593
```

Neste caso `log=TRUE` (padrão do comando). Através do Método Delta e com 90% de confiança temos que  $5,20 < \tau < 7,59$ . Observe que o IC não é simétrico em relação à estimação pontual e não representa o menor intervalo possível com a referida confiança. Para encontrar este intervalo, use `log=FALSE` e a rotina não realizará a parametrização do tipo  $\tau = e^\vartheta$ .

Para a utilização do método *bootstrap*, efetuamos o comando `plp.ml`, com a opção `bootstrap=TRUE`. Veja o Exemplo 3,

```
> plp.ml(transformers, ratio.cust=1/15, conf=.90, bootstrap=TRUE)
      Parameter Estimate StdDev CI.low  CI.up
1      shape      1.995   0.400  1.435  2.774
2      scale     24.366   2.798 20.172 29.431
3  shape.boot      1.995   0.527  1.550  2.830
4  scale.boot     24.366   3.014 21.461 29.163
5 Optimal.Maint(1/15)  6.772   1.033  1.349 15.272
```

No método bayesiano, utilizamos o comando `bayes.plp`. Com o mesmo exemplo e mantendo a mesma Razão de Custos  $Rc = 1/15$ , chegou-se ao seguinte resultado,

```
> bayes.plp(prior.beta=c(20,10), prior.eta=c(10,.5), data=transformers,
```

```

ratio.cust=1/15)
Parameter Estimate HPD.CI.low HPD.CI.up
1 beta'shape' 1.996 1.386 2.612
2 eta 20.667 13.654 28.059
3 theta'scale' 25.010 20.527 30.491
4 Optimal.Maint 6.617 5.221 7.969

```

Os três métodos ofereceram estimativas pontuais próximas umas das outras, sendo que o método *bootstrap* obteve um IC maior. No caso da Inferência bayesiana foram usadas prioris informativas.

# Capítulo 6

## Conclusão

Foi desenvolvida uma biblioteca robusta para o programa *R*, onde há várias opções de estimação dos parâmetros para a modelagem PPNH com função de intensidade PLP e de três funções dos parâmetros: função de intensidade  $\lambda(t)$ , média  $\Lambda(t)$  e tempo ótimo de manutenção  $\tau$ . Além disso, há comandos para elaboração de gráficos e testes de hipóteses que permitem diagnosticar a escolha do modelo. Estes métodos abrangem tanto análise para um único sistema quanto para  $k \geq 2$  sistemas. Ela também possui comandos que permitem simular casos PLP e casos PPNH com diferentes funções de intensidade  $\lambda(t)$ .

A presente dissertação serve como um manual de utilização da biblioteca. Ele possui exemplos com aplicações dos comandos a dados encontrados na literatura ou a dados simulados, onde é possível entender melhor como a teoria é aplicada na prática.

O PLP é bastante flexível e permite modelar situações diversas, ou seja, sistemas que melhoram, deterioram ou que nem melhoram, nem deterioram com o tempo. Valendo-se do princípio da parcimônia, situações similares também podem ter seus dados estudados pela nossa biblioteca. Entretanto, análise de confiabilidade de sistemas reparáveis é um assunto bastante amplo. A biblioteca *NHPPplp* está restrita à análise de PLP's mas outros tipos de  $\lambda(t)$  podem ser incluídas no futuro aumentando a capacidade de análise da biblioteca. A suposição de reparo mínimo nos restringe ao estudo de PPNH, todavia, em algumas situações práticas, observa-se que após o reparo o sistema não volta a mesma confiabilidade que estava

anteriormente. Isto abre um leque grande de diferentes formas de análise além do PPNH.

Um dos principais objetivos da análise de confiabilidade em sistemas reparáveis é encontrar um tempo ótimo de manutenção  $\tau$ . Nossa biblioteca oferece comandos para isto, mas temos algumas limitações, como  $\tau$  constante no tempo, assim como a razão de custos  $Rc$ . Se o estudo envolve grandes custos talvez seja adequado uma modelagem mais complexa, onde o tempo ótimo de manutenção  $\tau$  e a razão de custos  $Rc$  dependam do tempo  $t$  e/ou análises que não suponham reparo mínimo ou manutenção perfeita.

Enfim, este trabalho disponibiliza a estudantes, pesquisadores, engenheiros ou a qualquer um que venha a se interessar, uma biblioteca (*NHPPplp*) no programa *R* e um manual sobre ela (este relatório). Eles são uma ferramenta prática, livre e de fácil utilização para análise de confiabilidade em sistemas reparáveis.



# Bibliografia

- RD Baker. Some new tests of the power law process. *Technometrics*, 38(3):256–265, 1996.
- W. Baracho. Determinação de periodicidade Ótima de equipamentos reparáveis. *Instituto de Ciências Exatas da UFMG*, 2001.
- R.E. e R. Campo Barlow. *Total Time on Test Process and Applications to Failure Data Analysis. Reliability and Fault Tree Analysis*. Barlow, Fussel e Singpurwalla, Filadélfia, 1975.
- Ørnulf Borgan. Nelson–aalen estimator. *Encyclopedia of Biostatistics*, 2005.
- George Casella and Roger L Berger. Inferência estatística. *Centage Learning, ISBN13*, 291641363, 2010.
- Larry H Crow. Reliability analysis for complex systems. Technical report, DTIC Document, 1974.
- Larry H Crow. Reliability analysis for complex, repairable systems. Technical report, DTIC Document, 1975.
- Anthony Christopher Davison. *Bootstrap methods and their application*, volume 1. Cambridge university press, 1997.
- Marta A. e Neander F. Almeida Freitas. Estudo da tendência na intensidade de falhas em sistemas reparáveis. *A Pesquisa Operacional e o Desenvolvimento Sustentável*, 2007.

- Gustavo L Gilardoni and Enrico A Colosimo. Optimal maintenance time for repairable systems. *Journal of quality technology*, 39(1):48–53, 2007.
- Gustavo L Gilardoni and Enrico A Colosimo. On the superposition of overlapping poisson processes and nonparametric estimation of their intensity function. *Journal of Statistical Planning and Inference*, 141(9):3075–3083, 2011.
- Marcelo Nogueira Hodel. Análise comparativa entre ensaios controlados e aplicações reais de pastilhas de freio de caminhões. 2010.
- JT Molitor and SE Rigdon. A comparison of estimators of the shape parameter of the power law process. *Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association, Alexandria, VA*, pages 107–115, 1993.
- E. A. Motta, S. B. e Colosimo. Determination of preventive maintenance periodicities of standly devices by using. *Reliability Techniques: Reliability Engineering and System Safety*, (2):149–154, 2002.
- Maristela Dias de Oliveira, Enrico A Colosimo, and Gustavo L Gilardoni. Bayesian inference for power law processes with applications in repairable systems. *Journal of Statistical Planning and Inference*, 142(5):1151–1160, 2012.
- Gianpaolo Pulcini. A bounded intensity process for the reliability of repairable equipment. *Journal of Quality Technology*, 33(4):480–492, 2001.
- Steven E Rigdon and Asit P Basu. *Statistical methods for the reliability of repairable systems*. Wiley New York, 2000.
- Sidney Siegel. *Estatística não-paramétrica para as ciências do comportamento*. McGraw-Hill São Paulo, 1975.
- RDevelopment Core Team et al. Introdução ao r. *R foundation for Statistical Computing*, 2000. URL [cran.r-project.org/](http://cran.r-project.org/).