

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA
E COMPUTAÇÃO CIENTÍFICA

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Dissertação de Mestrado

ESTUDO DA EXTENSÃO DO MODELO BIVARIADO
EXPONENCIAL DE MARSHALL E OLKIN
PARA DADOS DE CONFIABILIDADE

Autora: Luzia Pedroso de Oliveira

Orientadora: Profa. Dra. Cicilia Yuko Wada

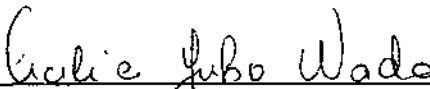
Campinas, SP
Janeiro, 2001

UNICAMP
BIBLIOTECA CENTRAL

ESTUDO DA EXTENSÃO DO MODELO BIVARIADO
EXPONENCIAL DE MARSHALL E OLKIN
PARA DADOS DE CONFIABILIDADE

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Luzia Pedroso de Oliveira e aprovada pela comissão julgadora.

Campinas, 15 de janeiro de 2001.


Profª. Dra. Círcia Yuko Wada
Orientadora

Banca Examinadora:

1. Profª. Dra. Círcia Yuko Wada
2. Prof. Dr. Francisco Louzada Neto
3. Prof. Dr. Ronaldo Dias

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Estatística.

UNIDADE	3e
N.º CHAMADA:	UNICAMP
	OL4e
V.	Ex.
TOMBO BC/	49044
PROC. / b.	892107
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREC. R\$	11,00
DATA	24/04/07
N.º CPD	

CM-00155146-7

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Oliveira, Luzia Pedroso de

OL4e Estudo da extensão do modelo bivariado exponencial de Marshall e Olkin para dados de confiabilidade / Luzia Pedroso de Oliveira – Campinas, [S.P. :s.n.], 2001.

Orientadora : Cicilia Yuko Wada

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

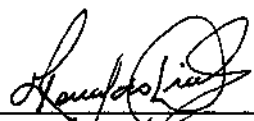
1. Confiabilidade – Métodos estatísticos. 2. Estimativa do parâmetro.
3. Testes de sobrevivência acelerados. I. Wada, Cicilia Yuko. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Dissertação de Mestrado defendida em 15 de dezembro de 2000 e aprovada

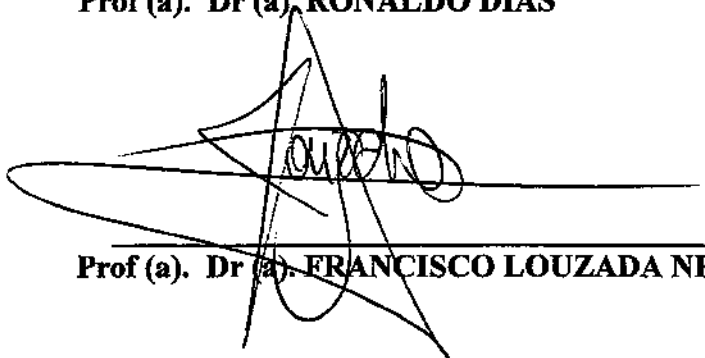
Pela Banca Examinadora composta pelos Profs. Drs.



Prof (a). Dr (a). CÍCILIA YUKO WADA



Prof (a). Dr (a). RONALDO DIAS



Prof (a). Dr (a). FRANCISCO LOUZADA NETO

*À toda a minha família,
em especial a minha filha Carla Vitória*

AGRADECIMENTOS

A Deus pelas minhas conquistas;

à professora Dra. Cicilia Yuko Wada pela orientação;

ao CNPQ pelo apoio financeiro;

ao professor Dr. Jorge Alberto Achcar pelas sugestões no início do trabalho;

aos meus pais pelo exemplo de perseverança e pelo apoio financeiro;

aos amigos prof. Dr. Carlos Alberto Diniz e Aline de Holanda Nunes Maia pelo conforto nos momentos difíceis;

aos companheiros da pós-graduação, em especial à Daniela Gonçalves Lenci e ao Mário Tarumoto, pela amizade;

aos funcionários da biblioteca, da secretaria, da limpeza e da manutenção do IMECC pelo cumprimento eficiente de seu trabalho;

enfim, gostaria de agradecer a todos que direta ou indiretamente contribuíram para a realização desse trabalho.

SUMÁRIO

LISTA DE FIGURAS	xiii
LISTA DE TABELAS	xv
RESUMO	xix
INTRODUÇÃO	1
CAPÍTULO I - REVISÃO BIBLIOGRÁFICA	5
1.1 A Distribuição Exponencial Bivariada de Marshall e Olkin e a sua Extensão.....	5
1.1.1 Derivação das Distribuições BVE e EBVE	5
1.1.2 Comparação entre as Distribuições BVE e EBVE.....	13
1.2 Modelo para Tempo Acelerado.....	17
1.3 Análise Clássica.....	21
1.3.1 Estimação dos Parâmetros usando o Método de Máxima Verossimilhança (MV).....	21
1.3.2 Propriedades dos Estimadores de MV.....	24
1.3.3 Diagnóstico de Normalidade Multivariada dos Estimadores de MV	25
1.4 Análise Bayesiana.....	26
1.4.1 Principais Conceitos.....	26
1.4.2 Métodos de Simulação Iterativa usados na Inferência Bayesiana.....	28
1.4.3 Critério de Convergência de Gelman e Rubin.....	30
CAPÍTULO II - ANÁLISE CLÁSSICA DA DISTRIBUIÇÃO EBVE	33
2.1 Geração de Dados da Distribuição EBVE.....	33
2.1.1 Geração de Dados da Distribuição EBVE usando o Método da Rejeição.....	34
2.1.2 Verificação da Validade das Amostras Geradas.....	35
2.2 Análise de Dados Bivariados sem Censura da Distribuição EBVE.....	36
2.2.1 Estimação dos Parâmetros usando o Método de MV e suas Propriedades.....	37
2.2.1.1 Exemplo de Aplicação.....	38
2.2.1.2 Simulações para Estudar as Propriedades dos Estimadores de MV.....	39

2.3	Análise de Dados Bivariados com Censura.....	47
2.3.1	Estimação dos Parâmetros usando o Método de MV: Simulações.....	47
CAPÍTULO III - ANÁLISE BAYESIANA DA DISTRIBUIÇÃO EBVE		53
3.1	Análise Bayesiana utilizando os Métodos de Gibbs e Metropolis Hastings.....	53
3.1.1	Exemplo de Aplicação.....	54
3.1.2	Estudo dos Parâmetros da Distribuição EBVE sob a Abordagem Bayes-Empírica..	60
CAPÍTULO IV - A DISTRIBUIÇÃO EBVE PARA TEMPOS ACELERADOS		69
4.1	Formulação de um Modelo para Testes Acelerados cujos Tempos Seguem a EBVE..	69
4.2	Estimação dos Parâmetros usando o Método de MV.....	75
4.2.1	Exemplo de Aplicação.....	75
4.2.2	Estudo de Simulação.....	79
CONCLUSÕES		91
APÊNDICES		93
A	- Funções de Sobrevivência da EBVE.....	93
A ₁	- Obtenção da Função de Sobrevivência de Z_i	93
A ₂	- Obtenção da Função de Sobrevivência Bivariada de (T_1, T_2)	95
A ₃	- Forma Geral da Distribuição EBVE e Três Casos Especiais.....	98
B	- Programa Computacional Implementado para a Geração de Dados da Distribuição EBVE pelo Método da Rejeição.....	101
C	- Programa Computacional Implementado para Estimação dos Parâmetros da Distribuição EBVE considerando Dados Completos.....	103
D	- Programa Computacional Implementado para Estimação dos Parâmetros da Distribuição EBVE considerando Dados com Censuras.....	125
E	- Programa Computacional Implementado para Estudar os Parâmetros da Distribuição EBVE sob a abordagem Bayes-Empírica.....	131
F	- Programa Computacional Implementado para Estimação dos Parâmetros da Distribuição EBVE considerando Tempos Acelerados	149
REFERÊNCIAS BIBLIOGRÁFICAS		163

LISTA DE FIGURAS

Figura 1.1 - Gráfico da função de densidade conjunta da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$	11
Figura 1.2 - Gráfico da função de sobrevivência conjunta da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$	11
Figura 1.3 - Gráficos das funções marginais de sobrevivência, de densidade e de risco dos tempos X_i , Z_i e T_i , da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$...	12
Figura 2.1 - Gráficos da densidade EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$ (interseções no plano $t_1=0$ e $t_2=0$).....	35
Figura 2.2 - Gráfico das médias das estimativas de MV e dos erros quadráticos médios dos estimadores dos parâmetros da EBVE com $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$, considerando dados sem censura.....	43
Figura 2.3 - Gráficos das coberturas dos intervalos de confiança aproximados, com coeficientes 90% e 95%, usando dados sem censura.....	43
Figura 2.4 - Histogramas das estimativas de MV ($n=30, 50$ e 100).....	44
Figura 2.5 - Histogramas das estimativas de MV ($n=200, 300$ e 500).....	45
Figura 2.6 - Gráficos normais probabilísticos das estimativas de MV ($n=30, 50$ e 500).....	46
Figura 2.7 - Gráficos das Médias das estimativas de MV dos parâmetros da EBVE para dados censurados.....	51
Figura 3.1 - Gráficos das densidades marginais <i>a priori</i> dos parâmetros assumidas no exemplo de aplicação.....	56
Figura 3.2 - Gráficos das trajetórias das cadeias, correspondentes ao exemplo de aplicação.....	57
Figura 3.3 - Gráficos das correlações dentro das cadeias, no exemplo de aplicação.....	58
Figura 3.4 - Gráfico das densidades marginais <i>a posteriori</i> dos parâmetros obtidas no exemplo de aplicação.....	59
Figura 3.5 - Gráficos das estimativas de MV, médias e medianas das densidades marginais <i>a posteriori</i> dos parâmetros juntamente com os intervalos de confiança e Bayesianos.....	63

- Figura 3.6 - Gráficos das médias e dos percentis 2,5 e 97,5 das densidades marginais *a priori* e *posteriori* dos parâmetros.....65
- Figura 4.1 – Gráficos das funções marginais de sobrevivência, de densidade e de risco, nos níveis de voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$74
- Figura 4.2 - Gráficos das médias das estimativas de MV dos parâmetros do modelo para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$ e na voltagem usual, $v_0=1,5$, onde $\lambda_1|v_0 = \lambda_2|v_0 = 0,1$, $\lambda_{12}|v_0 = 0,2$ e $s_1|v_0 = s_2|v_0 = 0,5$86
- Figura 4.3 - Gráficos das médias dos erros quadráticos médios dos estimadores dos parâmetros do modelo para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$ e na voltagem usual, $v_0=1,5$, onde $\lambda_1|v_0 = \lambda_2|v_0 = 0,1$, $\lambda_{12}|v_0 = 0,2$ e $s_1|v_0 = s_2|v_0 = 0,5$87
- Figura 4.4 - Cobertura dos intervalos aproximados, com 90% e 95% de confiança, dos parâmetros da distribuição EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$88
- Figura 4.5 - Cobertura dos intervalos aproximados, com 90% e 95% de confiança, dos parâmetros na voltagem usual, $v_0=1,5$89

LISTA DE TABELAS

Tabela 2.1 - Resumo da distribuição EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$	36
Tabela 2.2 - Média das médias, variâncias e correlações dos tempos (t_1 , t_2) das 500 amostras geradas da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$	36
Tabela 2.3 - Estimativas de MV e intervalos de confiança aproximados dos parâmetros da EBVE, no exemplo de aplicação.....	38
Tabela 2.4 - Estimativas de MV e intervalos de confiança aproximados das funções marginais, de confiabilidade e de risco, avaliadas no tempo $t=5$, correspondentes ao exemplo de aplicação	39
Tabela 2.5 - Estimativas dos quartis das distribuições marginais de T_1 e T_2 , correspondentes ao exemplo de aplicação	39
Tabela 2.6 - Descrição das amostras simuladas com dados sem censura.....	41
Tabela 2.7 - Médias das 500 estimativas de MV, dos desvios padrões (dp) e dos erros quadráticos médios (eqm) dos estimadores dos parâmetros da distribuição EBVE com $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$, considerando dados sem censura.....	41
Tabela 2.8 - Cobertura dos intervalos de confiança aproximados, com coeficientes 90% e 95%, considerando dados sem censura.....	42
Tabela 2.9 - Razões entre os erros quadráticos médios das estimativas de MV.....	42
Tabela 2.10 - Diagnóstico de normalidade multivariada dos estimadores de MV, considerando dados sem censura.....	42
Tabela 2.11 - Resultados do teste de Ryan-Joiner para verificar normalidade univariada.....	43
Tabela 2.12 - Descrição das amostras simuladas considerando dados censurados.....	48
Tabela 2.13 - Médias das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da distribuição EBVE com $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$, considerando dados censurados.....	49
Tabela 2.14 - Cobertura dos intervalos de confiança aproximados, com coeficientes 90% e 95%, usando dados censurados.....	50
Tabela 3.1 - Parâmetros das densidades <i>a priori</i> usados no exemplo de aplicação.....	55

Tabela 3.2 - Índices de convergência obtidos pelo critério de Gelman e Rubin	55
Tabela 3.3 - Resumo das densidades marginais <i>a posteriori</i> correspondentes ao exemplo de aplicação.....	60
Tabela 3.4 - Resumo das estimativas dos parâmetros da EBVE obtidas no Estudo de Simulação 3.1.....	62
Tabela 3.5 - Cobertura dos intervalos Bayesianos e de confiança aproximados, correspondentes ao Estudo de Simulação 3.1.....	62
Tabela 3.6 - Parâmetros das densidades <i>a priori</i> usados no Estudo de Simulação 3.2.....	67
Tabela 3.7 - Resumo das estimativas dos parâmetros da EBVE obtidas no Estudo de Simulação 3.2.....	67
Tabela 3.8 - Cobertura dos intervalos Bayesianos e de confiança aproximados, correspondentes ao Estudo de Simulação 3.2.....	67
Tabela 4.1 - Estimativas de MV e intervalos de confiança assintóticos dos parâmetros da distribuição EBVE para tempos acelerados, obtidas no exemplo de aplicação...76	
Tabela 4.2 - Estimativas de MV e intervalos de confiança dos parâmetros na voltagem $v_0=1,5$	77
Tabela 4.3 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$, $v_3=3$, correspondentes ao exemplo de aplicação.....	77
Tabela 4.4 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, da parte específica de cada componente, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$	78
Tabela 4.5 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, da parte comum de cada componentes, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$	78
Tabela 4.6 - Estimativas pontuais dos quartis da distribuição EBVE para tempos acelerados nos níveis de voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$	79
Tabela 4.7 - Descrição das amostras simuladas com tempos acelerados.....	81
Tabela 4.8 - Média das médias, variâncias e correlações dos tempos (t_1 , t_2) das 500 amostras geradas da EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$	82

Tabela 4.9 - Resumo dos valores teóricos da distribuição EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$	82
Tabela 4.10 - Média das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da distribuição EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$	83
Tabela 4.11 - Cobertura dos intervalos de confiança assintóticos dos parâmetros da distribuição EBVE para tempos acelerados.....	84
Tabela 4.12 - Média das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da distribuição EBVE, na voltagem $v_0 = 1,5$, dados por $\lambda_{1 v_0} = \lambda_{2 v_0} = 0,1$, $\lambda_{12 v_0} = 0,2$ e $s_{1 v_0} = s_{2 v_0} = 0,5$	84
Tabela 4.13 - Cobertura dos intervalos com 90% e 95% de confiança dos parâmetros da EBVE, na voltagem $v_0 = 1,5$	85

RESUMO

Dentre as várias distribuições exponenciais bivariadas apresentadas na literatura, a distribuição de Marshall e Olkin (BVE) recebe grande destaque. Devido a isso, Ryu (1993) propôs uma extensão da BVE que também possui propriedades com interpretações simples e úteis, com a vantagem de ser absolutamente contínua.

Neste trabalho estudamos a distribuição proposta por Ryu (EBVE) e formulamos um modelo para testes acelerados, onde os tempos até as falhas seguem essa distribuição, assumindo uma relação de potência inversa entre os tempos e a voltagem.

Foram gerados dados da EBVE e da distribuição formulada, usando o método da rejeição e feitas algumas simulações para verificar a validade das amostras obtidas.

Estudamos as propriedades assintóticas dos estimadores de máxima verossimilhança dos parâmetros da distribuição EBVE, considerando amostras com dados completos e censurados e também as dos parâmetros da distribuição EBVE para tempos acelerados.

Apresentamos uma análise Bayesiana do modelo, na qual assumimos densidades *a priori* informativas e encontramos as densidades marginais *a posteriori* dos parâmetros, utilizando os métodos de Gibbs e Metropolis Hastings.

INTRODUÇÃO

Os modelos de confiabilidade e de sobrevivência desempenham um papel importante em pesquisas de diversas áreas, especialmente em engenharia e ciências biomédicas, pois descrevem, respectivamente, o tempo até a falha de componentes e o tempo de sobrevivência de unidades biológicas.

Para modelar o tempo de um único componente ou de uma unidade biológica, são utilizados os modelos univariados, sendo as distribuições mais importantes, a exponencial, Weibull, gamma, valor extremo e log-normal. No estudo de dois ou mais componentes ou unidades, os modelos multivariados são mais apropriados do que os univariados, uma vez que conseguem explicar uma possível associação entre os tempos até as falhas ou de sobrevivência. Um caso particular dos modelos multivariados são os modelos bivariados, que descrevem dados pareados.

Métodos estatísticos têm sido extensamente desenvolvidos para os modelos univariados, enquanto que estudos sobre os modelos bivariados começam a aparecer na literatura somente nas últimas décadas.

Para modelar tempos pareados as distribuições exponenciais bivariadas são geralmente consideradas. Diferentemente da distribuição normal, não existe uma extensão natural única para a distribuição exponencial. Várias distribuições exponenciais bivariadas têm sido propostas. Gumbel (1960) propôs algumas distribuições com marginais exponenciais, mas não discutiu em que situações esses modelos se aplicam. Freund (1961) apresentou uma extensão exponencial específica para um sistema com dois componentes, onde o sistema pode continuar funcionando mesmo depois da falha de um dos componentes. Marshall e Olkin (1967) propuseram uma distribuição exponencial bivariada (BVE), para modelar a probabilidade de sobrevivência conjunta para tempos até as falhas de dois componentes

sujeitos a choques fatais, provenientes de três fontes independentes. Essa distribuição é bastante aceita, pois possui propriedades com interpretações físicas simples e úteis e satisfaz as propriedades de falta de memória marginal e conjunta, estendendo para o caso bivariado a propriedade de falta de memória da distribuição exponencial univariada. Contudo, ela apresenta a desvantagem de não ser absolutamente contínua, pois os componentes podem falhar simultaneamente com probabilidade positiva. Outras distribuições exponenciais bivariadas foram propostas por Downton (1970), Hawkes (1972) e Paulson (1973).

Devido à grande aceitação da BVE, várias distribuições exponenciais bivariadas têm sido propostas tentando preservar algumas das suas propriedades importantes, mas considerando que os componentes não falham simultaneamente. Block e Basu (1974) mostraram que a única distribuição bivariada absolutamente contínua que satisfaz as propriedades de falta de memória marginal e conjunta, é a distribuição bivariada com exponenciais independentes. Neste mesmo artigo, Block e Basu propuseram uma distribuição exponencial bivariada absolutamente contínua, a ACBVE, que conserva a propriedade de falta de memória conjunta e cujas distribuições marginais são médias ponderadas de exponenciais. Considerando os resultados de Block e Basu, as distribuições comparáveis com a BVE de Marshall e Olkin, propostas posteriormente, não consideram algumas das propriedades de falta de memória. Sarkar (1987) deriva uma distribuição exponencial bivariada absolutamente contínua, a $ACBVE_2$ com marginais exponenciais. Ryu (1993) propôs uma extensão da BVE, que chamaremos de EBVE. A EBVE apresenta funções de risco marginais crescentes e também não satisfaz a propriedade de falta de memória conjunta, entretanto, ela permite ter as propriedades de falta de memória marginal e conjunta para algum grau de aproximação.

O objetivo desta dissertação é estudar a distribuição EBVE sob os seguintes aspectos:

- a- comparação das suas propriedades com as da distribuição BVE;
- b- geração de dados através do método da rejeição e verificação da validade das amostras obtidas;
- c- estimação dos parâmetros e de suas funções usando o método de máxima verossimilhança (MV), para amostras com dados não censurados e com censura do tipo I;

- d- estimação dos parâmetros através da inferência Bayesiana, assumindo densidades *a priori* informativas;
- e- formulação de um modelo para tempos acelerados e estimação dos parâmetros do modelo proposto usando o método de MV.

No Capítulo I fazemos uma revisão das distribuições BVE e EBVE e apresentamos alguns conceitos básicos de inferência clássica e Bayesiana.

No Capítulo II geramos amostras da distribuição EBVE, utilizando o método da rejeição e fazemos um estudo de simulação para verificar a validade das amostras geradas. Ainda neste capítulo, obtemos as estimativas de MV dos parâmetros da EBVE e estudamos as propriedades assintóticas dos estimadores de MV, considerando tempos não censurados e com censura do tipo I.

No Capítulo III estudamos o modelo EBVE sob a abordagem Bayes-empírica. Assumimos densidades *a priori* informativas para os parâmetros e encontramos as densidades marginais *a posteriori*, utilizando os métodos de simulação iterativa, o amostrador de Gibbs e o algoritmo de Metropolis Hastings (M-H). Apresentamos um exemplo de aplicação e fazemos simulações para estudar o modelo.

No Capítulo IV formulamos um modelo para testes acelerados, onde os tempos até as falhas seguem a EBVE, utilizando a relação de potência inversa. Apresentamos um exemplo de aplicação e estudamos as propriedades dos estimadores de MV através de simulações.

No Apêndice A mostramos como foram obtidas as funções de sobrevivência da EBVE. Apresentamos os programas computacionais implementados, para a geração de dados da EBVE, pelo método da rejeição (Apêndice B), para estudo dos estimadores de MV dos parâmetros da EBVE, utilizando dados não censurados (Apêndice C) e dados com censura do tipo I (Apêndice D). No Apêndice E, encontra-se o programa implementado para estudar os parâmetros do modelo sob a abordagem Bayes-empírica e finalmente no Apêndice F o programa computacional para estudar os estimadores de MV, considerando amostras da EBVE para tempos acelerados.

CAPÍTULO I

REVISÃO BIBLIOGRÁFICA

1.1 A Distribuição Exponencial Bivariada de Marshall e Olkin e a sua Extensão

Existem várias distribuições exponenciais bivariadas propostas na literatura estatística, sendo que a distribuição exponencial bivariada de Marshall e Olkin (1967), BVE, recebe maior destaque, devido ao fato de ter propriedades com interpretações físicas simples e úteis. Entretanto, como consequência das condições sob as quais foi derivada, essa distribuição não é apropriada em situações onde dois componentes não falham simultaneamente ou quando não podemos garantir as propriedades de falta de memória, marginal e conjunta. Por esta razão, Ryu (1993), propôs um modelo exponencial bivariado, que é adequado para as situações não contempladas pela BVE e que também possui propriedades com interpretações simples. Ryu considerou o modelo proposto como uma extensão da BVE, que chamaremos de EBVE.

Nesta seção fazemos uma revisão das distribuições BVE e EBVE, onde mostramos como elas foram derivadas e comparamos as suas propriedades.

1.1.1 Derivação das Distribuições BVE e EBVE

a) A Distribuição BVE

Primeiramente descrevemos os modelos de choque fatal e os modelos de choques não fatais com os quais se obtém a função de sobrevivência, ou função de confiabilidade, conjunta, para tempos até a falha que seguem a distribuição BVE.

No modelo de choque fatal os componentes de um sistema com dois componentes falham depois de receber um choque que é sempre fatal. Assume-se que os choques são regidos por três processos de Poisson independentes $\{Z_1(t), t \geq 0; \lambda_1\}$, $\{Z_2(t), t \geq 0; \lambda_2\}$, $\{Z_{12}(t), t \geq 0; \lambda_{12}\}$ onde λ_1 , λ_2 e λ_{12} são as intensidades dos processos. Os eventos no processo $\{Z_i(t), t \geq 0; \lambda_i\}$, $i=1, 2$, são choques no componente i e os eventos no processo $\{Z_{12}(t), t \geq 0; \lambda_{12}\}$ são os choques que ocorrem nos dois componentes. Considerando as variáveis aleatórias T_1 e T_2 os tempos até as falhas dos componentes 1 e 2 respectivamente, a função de sobrevivência conjunta é dada por:

$$\begin{aligned} S_{T_1, T_2}(t_1, t_2) &= P(T_1 > t_1, T_2 > t_2) = P[Z_1(t_1; \lambda_1) = 0, Z_2(t_2; \lambda_2) = 0, Z_{12}(\max(t_1, t_2); \lambda_{12}) = 0] \\ &= \exp(-\lambda_1 t_1 - \lambda_2 t_2 - \lambda_{12} \max(t_1, t_2)). \end{aligned} \quad (1.1)$$

No modelo de choques não fatais os choques que ocorrem nos componentes não são necessariamente fatais. Os choques são regidos por três processos de Poisson independentes $\{Z_1(t), t \geq 0; \beta_1\}$, $\{Z_2(t), t \geq 0; \beta_2\}$, $\{Z_{12}(t), t \geq 0; \beta_{12}\}$ onde β_1 , β_2 e β_{12} são as intensidades dos processos.

Descrevendo as condições dos componentes pelos pares ordenados $(0, 0)$, $(0, 1)$, $(1, 0)$ e $(1, 1)$, onde as coordenadas representam o primeiro e o segundo componente sendo que 1 indica que o componente está funcionando e 0 que o componente falhou, os eventos no processo $\{Z_1(t), t \geq 0; \beta_1\}$ são choques que ocorrem no primeiro componente causando uma transição da condição $(1, 1)$ para $(0, 1)$ com probabilidade p_1 e para $(1, 0)$ com probabilidade $(1 - p_1)$. Similarmente, os eventos no processo $\{Z_2(t), t \geq 0; \beta_2\}$ são choques que ocorrem no segundo componente causando uma transição da condição $(1, 1)$ para $(1, 0)$ com probabilidade p_2 e para $(1, 1)$ com probabilidade $(1 - p_2)$. Os eventos no processo $\{Z_{12}(t), t \geq 0; \beta_{12}\}$ são choques que ocorrem em ambos os componentes e causam uma transição da condição $(1, 1)$ para $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ com as respectivas probabilidades: p_{00} , p_{01} , p_{10} , p_{11} .

Assume-se também que cada choque em um componente representa uma chance independente para falha. Considerando as variáveis aleatórias T_1 e T_2 , tempos até as falhas do primeiro e segundo componente, temos para $t_1 \leq t_2$ e $t_1 \geq t_2$ respectivamente:

$$\begin{aligned}
 S_{T_1, T_2}(t_1, t_2) &= P(T_1 > t_1, T_2 > t_2) = \left[\sum_{k=0}^{\infty} e^{-\beta_1 t_1} \frac{(\beta_1 t_1)^k}{k!} (1-p_1)^k \right] \left[\sum_{l=0}^{\infty} e^{-\beta_2 t_2} \frac{(\beta_2 t_2)^l}{l!} (1-p_2)^l \right] \\
 &\times \left[\sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \left(e^{-\beta_{12} t_1} \frac{(\beta_{12} t_1)^m}{m!} (p_{11})^m \right) \left(e^{-\beta_{12}(t_2-t_1)} \frac{(\beta_{12}(t_2-t_1))^n}{n!} (p_{11} + p_{01})^n \right) \right] \\
 &= \exp \left[-t_1(\beta_1 p_1 + \beta_{12} p_{01}) - t_2(\beta_2 p_2 + \beta_{12}(1-p_{11}-p_{01})) \right] \quad (1.2)
 \end{aligned}$$

e

$$S_{T_1, T_2}(t_1, t_2) = \exp \left[-t_1(\beta_1 p_1 + \beta_{12}(1-p_{11}-p_{10})) - t_2(\beta_2 p_2 + \beta_{12} p_{10}) \right]. \quad (1.3)$$

Consequentemente, combinando (1.2) e (1.3) a função de sobrevivência conjunta é dada por (1.1), onde $\lambda_1 = \beta_1 p_1 + \beta_{12} p_{01}$, $\lambda_2 = \beta_2 p_2 + \beta_{12} p_{10}$, $\lambda_{12} = \beta_{12} p_{00}$.

A distribuição BVE também pode ser derivada da relação de independência entre os tempos restantes até a falha dos componentes que não falharam até um certo tempo e o tempo de funcionamento dos mesmos, ver Marshall e Olkin (1967).

b) A Distribuição EBVE

Da mesma forma que na BVE, na distribuição EBVE assume-se que os choques, que ocorrem nos dois componentes, são regidos por três processos de Poisson independentes, $\{N_1(t), t \geq 0; \lambda_1\}$, $\{N_2(t), t \geq 0; \lambda_2\}$ e $\{N_{12}(t), t \geq 0; \lambda_{12}\}$, onde as variáveis aleatórias $N_i(t)$, $i=1, 2$, medem o número de choques que ocorrem na parte específica do componente i no intervalo $(0, t]$ e $N_{12}(t)$ mede o número de choques que ocorrem nas partes comuns dos componentes 1 e 2 em $(0, t]$. Os parâmetros λ_i , $i=1, 2$, e λ_{12} representam respectivamente, o número médio de choques que ocorrem na parte específica do componente i e nas partes comuns dos componentes 1 e 2, durante um intervalo de tempo unitário.

Assume-se que a função de risco condicional aos processos de Poisson, no tempo t para falha do componente i , $i=1, 2$, é dada por:

$$h_i(t) = d_i N_i(t) + s_i N_{12}(t), \quad (1.4)$$

onde d_i refere-se ao aumento na função de risco, devido a choques na parte específica do componente i e s_i refere-se ao aumento na função de risco, devido a choques na parte comum

do componente $i, i=1, 2$. Para que a distribuição seja realmente comparável com a BVE e para uma simplificação dos cálculos é assumido que $d_1=d_2=\infty$. Dessa forma, tem-se que a ocorrência de um choque fatal, na parte específica do componente $i, i=1, 2$ resulta na falha imediata do componente, enquanto que os choques não fatais, que ocorrem nas partes específicas, não afetam as condições físicas dos componentes. Por outro lado, os choques não fatais na parte comum causam um desgaste importante nos componentes, aumentando a função de risco para a falha do componente i pela quantidade s_i . Observemos que se $s_1=s_2=\infty$ então o modelo se reduz a BVE, onde os choques não fatais que ocorrem tanto nas partes específicas quanto nas partes comuns não são acumulativos.

A função de sobrevivência conjunta para o modelo geral, juntamente com alguns casos particulares são apresentados no Apêndice A₃. Neste capítulo apresentamos a derivação da EBVE considerando $d_1=d_2=\infty$.

Representemos por $X_i, i=1, 2$, a variável aleatória que mede o tempo até o primeiro salto no processo $\{N_i(t), t \geq 0; \lambda_i\}$, ou seja, o tempo até a ocorrência de um choque na parte específica do componente i , por $Z_i, i=1, 2$, o tempo até a ocorrência de um choque fatal nas partes comuns do componente i e por $T_i, i=1, 2$, o tempo até a falha do componente i , onde $T_i = \min(X_i, Z_i)$.

Das considerações anteriores tem-se que $X_i \sim \text{Exp}(\lambda_i)$ e portanto a função de sobrevivência de X_i é dada por:

$$S_{X_i}(t) = P(X_i > t) = \exp(-\lambda_i t), \tag{1.5}$$

logo, a função de risco de X_i que é definida como:

$$h_{X_i}(t) = \lim_{\Delta t \rightarrow \infty} \frac{P(t \leq X_i < t + \Delta t | X_i \geq t)}{\Delta t} = \frac{f_{X_i}(t)}{S_{X_i}(t)} = \frac{-\frac{\partial}{\partial t} S_{X_i}(t)}{S_{X_i}(t)} = -\frac{\partial}{\partial t} \log S_{X_i}(t)$$

é constante em t , ou seja,

$$h_{X_i}(t) = \lambda_i. \tag{1.6}$$

A função de sobrevivência de Z_i (Apêndice A₁) é dada por:

$$S_{Z_i}(t) = P(Z_i > t) = \exp\left(-\lambda_{12} t + \frac{\lambda_{12}}{s_i} (1 - e^{-s_i t})\right) \tag{1.7}$$

e a função de risco de Z_i é dada por:

$$h_{Z_i}(t) = \lambda_{12} (1 - e^{-s_i t}). \quad (1.8)$$

Observemos que h_{Z_i} é uma função crescente em t a menos que $s_i = \infty$

Como X_i e Z_i são independentes, a função de risco de T_i é dada pela soma das funções de riscos, ou seja,

$$h_{T_i}(t) = \lambda_i + \lambda_{12} (1 - e^{-s_i t}). \quad (1.9)$$

Da última equação obtém-se a função de sobrevivência de T_i , uma vez que

$$S_{T_i}(t) = \exp\left(-\int_0^t h_{T_i}(u) du\right),$$

dada por:

$$S_{T_i}(t) = P(T_i > t) = \exp\left(-\lambda_i t - \lambda_{12} t + \frac{\lambda_{12}}{s_i} (1 - e^{-s_i t})\right). \quad (1.10)$$

A função de densidade de T_i é obtida através da relação

$$f_{T_i}(t) = -\frac{\partial}{\partial t} S_{T_i}(t)$$

pois $S_{T_i}(t)$ é absolutamente contínua, e resulta em

$$f_{T_i}(t) = \left[\lambda_i + \lambda_{12} (1 - e^{-s_i t})\right] \exp\left(-\lambda_i t - \lambda_{12} t + \frac{\lambda_{12}}{s_i} (1 - e^{-s_i t})\right). \quad (1.11)$$

Remarcamos que se $s_i = \infty$, a distribuição se reduz a uma distribuição exponencial com função de risco constante $\lambda_i + \lambda_{12}$ e assim satisfaz a propriedade de falta de memória marginal.

A função de sobrevivência conjunta de (T_1, T_2) é obtida (Apêndice A₂) de forma similar à função de sobrevivência de Z_i e apresenta a seguinte forma:

$$S_{T_1, T_2}(t_1, t_2) = \begin{cases} \exp\left(-(\lambda_1 + \lambda_{12})t_1 - \lambda_2 t_2 + \frac{\lambda_{12}}{s_1} (1 - e^{-s_1(t_1-t_2)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_1(t_1-t_2)} - e^{-s_1 t_1 - s_2 t_2})\right) & \text{se } t_1 > t_2 \\ \exp\left(-\lambda_1 t_1 - (\lambda_2 + \lambda_{12})t_2 + \frac{\lambda_{12}}{s_2} (1 - e^{-s_2(t_2-t_1)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2})\right) & \text{se } t_1 \leq t_2. \end{cases} \quad (1.12)$$

Como $f_{T_1, T_2}(t_1, t_2) = \frac{\partial^2}{\partial t_1 \partial t_2} S_{T_1, T_2}(t_1, t_2)$, pois $S_{T_1, T_2}(t_1, t_2)$ é absolutamente contínua, a densidade conjunta de (T_1, T_2) é dada por:

$$f_{T_1, T_2}(t_1, t_2) = \begin{cases} S_{T_1, T_2}(t_1, t_2) \left(\lambda_2 + \lambda_{12} e^{-s_1(t_1-t_2)} - \frac{\lambda_{12}}{s_1 + s_2} (s_1 e^{-s_1(t_1-t_2)} + s_2 e^{-s_1 t_1 - s_2 t_2}) \right) \\ \times \left(\lambda_1 + \lambda_{12} - \lambda_{12} e^{-s_1(t_1-t_2)} + \frac{s_1 \lambda_{12}}{s_1 + s_2} (e^{-s_1(t_1-t_2)} - e^{-s_1 t_1 - s_2 t_2}) \right) + S_{T_1, T_2}(t_1, t_2) \\ \times \left(s_1 \lambda_{12} e^{-s_1(t_1-t_2)} - \frac{s_1 \lambda_{12}}{s_1 + s_2} (s_1 e^{-s_1(t_1-t_2)} + s_2 e^{-s_1 t_1 - s_2 t_2}) \right) & \text{se } t_1 > t_2 \\ S_{T_1, T_2}(t_1, t_2) \left(\lambda_1 + \lambda_{12} e^{-s_2(t_2-t_1)} - \frac{\lambda_{12}}{s_1 + s_2} (s_2 e^{-s_2(t_2-t_1)} + s_1 e^{-s_1 t_1 - s_2 t_2}) \right) \\ \times \left(\lambda_2 + \lambda_{12} - \lambda_{12} e^{-s_2(t_2-t_1)} + \frac{s_2 \lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2}) \right) + S_{T_1, T_2}(t_1, t_2) \\ \times \left(s_2 \lambda_{12} e^{-s_2(t_2-t_1)} - \frac{s_2 \lambda_{12}}{s_1 + s_2} (s_2 e^{-s_2(t_2-t_1)} + s_1 e^{-s_1 t_1 - s_2 t_2}) \right) & \text{se } t_1 \leq t_2. \end{cases} \quad (1.13)$$

Mudando a distribuição de X_i de exponencial para Weibull obtém-se uma generalização da distribuição EBVE e neste caso a função de sobrevivência conjunta é dada por:

$$S_{T_1, T_2}(t_1, t_2) = \begin{cases} \exp \left(-\lambda_1 t_1^{\alpha_1} - \lambda_{12} t_1 - \lambda_2 t_2^{\alpha_2} + \frac{\lambda_{12}}{s_2} (1 - e^{-s_1(t_1-t_2)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_1(t_1-t_2)} - e^{-s_1 t_1 - s_2 t_2}) \right) & \text{se } t_1 > t_2 \\ \exp \left(-\lambda_1 t_1^{\alpha_1} - \lambda_2 t_2^{\alpha_2} - \lambda_{12} t_2 + \frac{\lambda_{12}}{s_2} (1 - e^{-s_2(t_2-t_1)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2}) \right) & \text{se } t_1 \leq t_2 \end{cases} \quad (1.14)$$

Note que se $\alpha_1 \rightarrow 1$ e $\alpha_2 \rightarrow 1$ a função de sobrevivência em (1.14) se reduz a (1.12).

A seguir apresentamos os gráficos das funções conjuntas de sobrevivência e de densidade, Figuras 1.1 e 1.2 respectivamente e também das funções marginais de sobrevivência, de densidade e de risco, Figura 1.3 da distribuição EBVE com parâmetros $\lambda_1 = \lambda_2 = 0,1$, $\lambda_{12} = 0,2$ e $s_1 = s_2 = 0,5$.

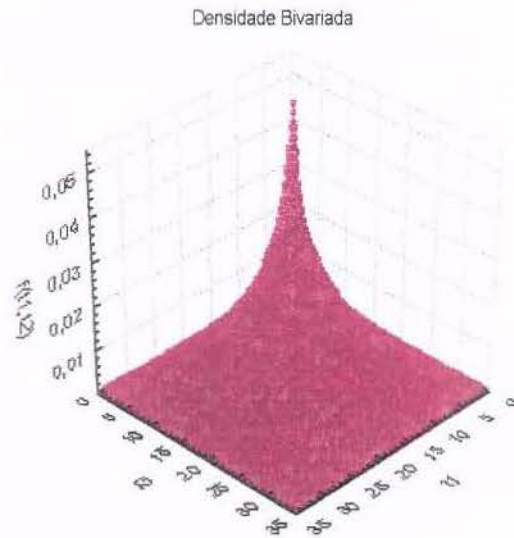


Figura 1.1 Gráfico da função de densidade conjunta da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1 = s_2=0,5$.

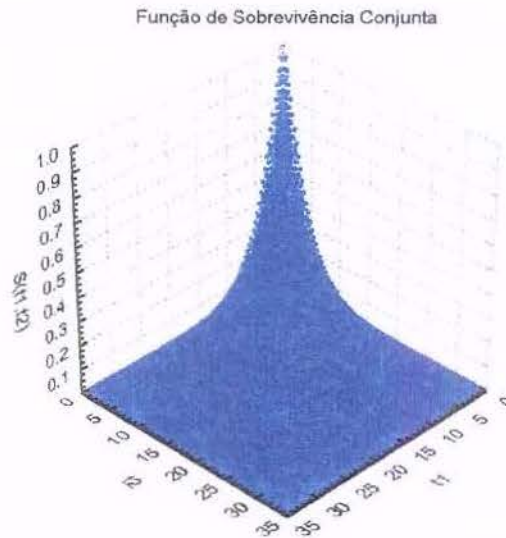


Figura 1.2 Gráfico da função de sobrevivência conjunta da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1 = s_2=0,5$.

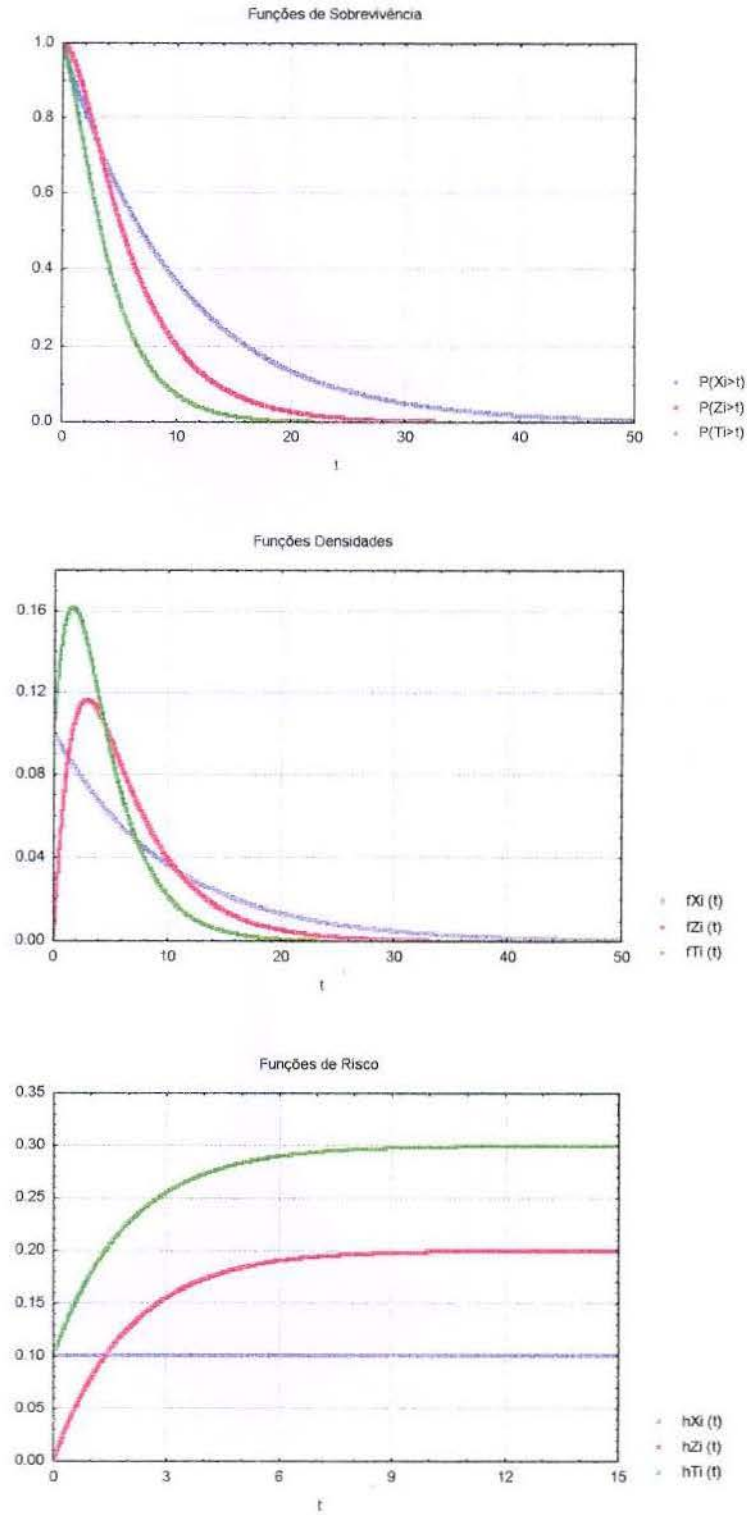


Figura 1.3 Gráficos das funções marginais de sobrevivência, de densidade e de risco dos tempos X_i , Z_i e T_i da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$.

1.1.2 Comparação entre as Distribuições BVE e EBVE

A seguir fazemos uma comparação entre as distribuições BVE e EBVE com relação aos componentes que elas podem modelar (a) e às suas propriedades (b), (Ryu, 1993).

a) Situações Físicas Modeladas pelas Distribuições

Os modelos BVE e EBVE são apropriados para descrever a confiabilidade conjunta de dois componentes satisfazendo as seguintes condições:

- i) Os componentes possuem partes que apresentam o mesmo risco de falha (parte comum) e partes que são específicas (parte específica).
- ii) Os componentes estão sujeitos a choques não fatais e fatal. Os choques que ocorrem na parte comum são independentes dos choques que ocorrem na parte específica e os choques que ocorrem na parte específica de um componente são independentes dos choques que ocorrem na parte específica do outro componente.
- iii) Cada um dos componentes apresenta uma estrutura em série, ou seja, um componente falha quando ocorre um choque fatal em qualquer parte, comum ou específica.
- iv) Na BVE os choques não fatais que ocorrem nas partes comum e específicas não afetam as condições físicas dos componentes. Por outro lado, na EBVE os choques não fatais podem afetar as condições físicas dos componentes.
- iv) Na BVE a ocorrência de um choque fatal na parte comum acarreta falha nos dois componentes o que necessariamente não ocorre na EBVE.

b) Propriedades das Distribuições

- i) A BVE não é uma distribuição absolutamente contínua como é a distribuição EBVE, pois na BVE se um choque fatal ocorre na parte comum precedendo um choque nas partes específicas então o tempo até a falha dos dois componentes será o mesmo. Se denotarmos por X_i , $i=1,2$, a variável aleatória que representa o tempo até a ocorrência de um choque fatal nas partes específicas do componente i e denotarmos por Z a variável aleatória que

representa o tempo até a ocorrência de um choque fatal na parte comum dos componentes então:

$$\begin{aligned}
 P(T_1 = T_2) &= P(Z < X_1, Z < X_2) = P(Z < \min(X_1, X_2)) \\
 &= \int_0^{\infty} \int_z^{\infty} f_{Z, \min(X_1, X_2)}(z, x) dx dz \\
 &= \int_0^{\infty} \int_z^{\infty} f_Z(z) f_{\min(X_1, X_2)}(x) dx dz \\
 &= \int_0^{\infty} \int_z^{\infty} \lambda_{12} e^{-\lambda_{12}z} (\lambda_1 + \lambda_2) e^{-(\lambda_1 + \lambda_2)x} dx dz \\
 &= \int_0^{\infty} \lambda_{12} e^{-\lambda_{12}z} \left[\int_z^{\infty} (\lambda_1 + \lambda_2) e^{-(\lambda_1 + \lambda_2)x} dx \right] dz \\
 &= \int_0^{\infty} \lambda_{12} e^{-\lambda_{12}z} e^{-(\lambda_1 + \lambda_2)z} dz \\
 &= \lambda_{12} \int_0^{\infty} e^{-(\lambda_1 + \lambda_2 + \lambda_{12})z} dz \\
 &= \frac{\lambda_{12}}{\lambda_1 + \lambda_2 + \lambda_{12}},
 \end{aligned}$$

e portanto, na BVE a probabilidade que os componentes falhem simultaneamente é dada por:

$$P(T_1 = T_2) = \frac{\lambda_{12}}{\lambda_1 + \lambda_2 + \lambda_{12}}.$$

- ii) A distribuição EBVE é a BVE quando $s_1 = s_2 = \infty$.
- iii) Como a BVE, a EBVE foi derivada baseada em uma situação física e possui propriedades com interpretações físicas simples, o que difere das outras distribuições que também são comparáveis com a BVE como por exemplo a ACBVE e a ACBVE₂.
- iv) A BVE apresenta a propriedade de falta de memória marginal sendo as distribuições marginais exponenciais com parâmetros $(\lambda_1 + \lambda_{12})$. Por outro lado, as distribuições

marginais da EBVE não são exponenciais apresentando dessa forma riscos de falha crescentes, o que a torna mais apropriada em algumas situações práticas.

- v) A BVE apresenta a propriedade de falta de memória conjunta, ou seja, a probabilidade de sobrevivência de dois componentes que já tem um certo tempo de funcionamento é a mesma do que a de dois componentes novos, ou seja,

$$P(T_1 > t_1 + t, T_2 > t_2 + t | T_1 > t, T_2 > t) = P(T_1 > t_1, T_2 > t_2)$$

para todo $t_1, t_2, t \geq 0$. Na EBVE a probabilidade que os componentes não falhem até um certo tempo é menor para os componentes com menor tempo de uso. Assim sendo,

$$P(T_1 > t_1 + t, T_2 > t_2 + t | T_1 > t, T_2 > t) \leq P(T_1 > t_1, T_2 > t_2)$$

para todo t_1, t_2 e $t \geq 0$, pois

$$\begin{aligned} P(T_1 > t_1 + t, T_2 > t_2 + t | T_1 > t, T_2 > t) &\leq P(T_1 > t_1, T_2 > t_2) \\ \Leftrightarrow \frac{P(T_1 > t_1 + t, T_2 > t_2 + t, T_1 > t, T_2 > t)}{P(T_1 > t, T_2 > t)} &\leq P(T_1 > t_1, T_2 > t_2) \\ \Leftrightarrow \frac{P(T_1 > t_1 + t, T_2 > t_2 + t)}{P(T_1 > t, T_2 > t)} &\leq P(T_1 > t_1, T_2 > t_2) \\ \Leftrightarrow \frac{P(T_1 > t_1, T_2 > t_2)P(T_1 > t, T_2 > t)}{P(T_1 > t_1 + t, T_2 > t_2 + t)} &\geq 1 \end{aligned}$$

e como para todo $t_1, t_2 \geq 0$,

$$\frac{P(T_1 > t_1, T_2 > t_2)P(T_1 > t, T_2 > t)}{P(T_1 > t_1 + t, T_2 > t_2 + t)} = \exp\left(\frac{\lambda_{12}}{s_1 + s_2} \left(1 - e^{-(s_1 + s_2)t}\right) \left(1 - e^{-s_1 t_1 - s_2 t_2}\right)\right)$$

e sendo $\lambda_{12}, s_1, s_2 \geq 0$, tem-se que,

$$\left(\frac{\lambda_{12}}{s_1 + s_2} \left(1 - e^{-(s_1 + s_2)t}\right) \left(1 - e^{-s_1 t_1 - s_2 t_2}\right)\right) \geq 0$$

e portanto,

$$\exp\left(\frac{\lambda_{12}}{s_1 + s_2} \left(1 - e^{-(s_1 + s_2)t}\right) \left(1 - e^{-s_1 t_1 - s_2 t_2}\right)\right) \geq 1.$$

vi) Na BVE a variável $\min(T_1, T_2)$ tem distribuição exponencial com parâmetro $(\lambda_1 + \lambda_{12})$, enquanto que na EBVE a função de sobrevivência do $\min(T_1, T_2)$ é dada por:

$$\begin{aligned} S_{\min(T_1, T_2)}(t) &= P(\min(T_1, T_2) > t) = P(T_1 > t, T_2 > t) \\ &= P(\min(X_1, Z_1) > t, \min(X_2, Z_2) > t) \\ &= P(X_1 > t, Z_1 > t, X_2 > t, Z_2 > t) \\ &= P(X_1 > t)P(X_2 > t)E(P(Z_1 > t, Z_2 > t | N_{12})) \\ &= \exp(-(\lambda_1 + \lambda_2))t E\left(\exp\left(- (s_1 + s_2) \int_0^t N_{12}(u) du\right)\right) \\ &= \exp\left[-(\lambda_1 + \lambda_2 + \lambda_{12})t + \frac{\lambda_{12}}{s_1 + s_2} (1 - e^{-(s_1 + s_2)t})\right]. \end{aligned}$$

vii) Na BVE o $\min(T_1, T_2)$ é independente de $T_1 - T_2$, ou seja,

$$P(\min(T_1, T_2) > t, T_1 - T_2 > u) = P(\min(T_1, T_2) > t)P(T_1 - T_2 > u)$$

enquanto que na EBVE a igualdade acima não ocorre a não ser que $\lambda_{12} = 0$ ou $s_1 + s_2 = \infty$. Isto significa que na distribuição EBVE o tempo até a falha de um componente fornece informação útil sobre o tempo restante do outro componente, a menos que não ocorram choques nas partes comuns ($\lambda_{12} = 0$) ou que um choque nas partes comuns seja fatal para pelo menos um dos componentes ($s_1 = \infty$ ou $s_2 = \infty$ ou $s_1 = s_2 = \infty$).

viii) Na EBVE a probabilidade de que os componentes não falhem até um certo tempo é maior do que na BVE, pois

$$\frac{S_{T_1, T_2}(t_1, t_2)}{S_{T_1, T_2, BVE}(t_1, t_2)} = \exp\left(\frac{\lambda_{12}}{\bar{s}} (1 - e^{-\bar{s}|t_1 - t_2|}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-\bar{s}|t_1 - t_2|} - e^{-s_1 t_1 - s_2 t_2})\right) \quad (1.15)$$

onde $\bar{s} = s_1$ se $t_1 > t_2$ e $\bar{s} = s_2$ se $t_1 \leq t_2$. Como

$$\left(\frac{\lambda_{12}}{\bar{s}} (1 - e^{-\bar{s}|t_1 - t_2|}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-\bar{s}|t_1 - t_2|} - e^{-s_1 t_1 - s_2 t_2})\right) \geq 0$$

então, a razão entre as funções de sobrevivência é maior ou igual a 1. Mais ainda,

$$\max_{T_1, T_2} \frac{S_{T_1, T_2}(t_1, t_2)}{S_{T_1, T_2, BVE}(t_1, t_2)} \leq \exp\left(\frac{\lambda_{12}}{\min(s_1, s_2)}\right).$$

De fato, observando que a razão dada em (1.15) está aumentando em t_1 para $t_1 > t_2$ e em t_2 para $t_1 \leq t_2$, temos que

$$\frac{S_{T_1, T_2}(t_1, t_2)}{S_{T_1, T_2, BVE}(t_1, t_2)} = \begin{cases} \leq \exp\left(\frac{\lambda_{12}}{s_1}\right) & p / t_1 > t_2 \\ \leq \exp\left(\frac{\lambda_{12}}{s_2}\right) & p / t_1 \leq t_2 \end{cases}$$

logo,

$$\frac{S_{T_1, T_2}(t_1, t_2)}{S_{T_1, T_2, BVE}(t_1, t_2)} \leq \exp\left(\frac{\lambda_{12}}{\min(s_1, s_2)}\right).$$

Note que, nos casos onde $\lambda_{12}=0$, ou seja, quando não ocorrem choques comuns e quando os choques agem de forma não acumulativa $s_1=s_2=\infty$,

$$\frac{S_{T_1, T_2}(t_1, t_2)}{S_{T_1, T_2, BVE}(t_1, t_2)} = 1$$

1.2 Modelo para Tempo Acelerado

Para estimar a confiabilidade de componentes manufaturados é necessário analisar uma amostra de seus tempos até as falhas. Entretanto, devido a alta confiabilidade de muitos componentes torna-se impraticável esperar que eles falhem naturalmente, ou seja, sob condições usuais de operação. Assim sendo, visando reduzir o tempo e o custo para a realização desses experimentos os componentes são colocados em teste sob alguns níveis de *stress* mais severos do que o usual de forma a acelerar os seus tempos até as falhas. Os *stress* de aceleração são escolhidos de acordo com as características dos componentes sendo os mais usuais temperatura, voltagem, cargas mecânicas, umidade e vibração e podem ser aplicados de várias formas incluindo *stress* constante, cíclico, progressivo, aleatório sendo o primeiro o mais utilizado, ver por exemplo, Nelson (1990).

Através do ajuste de um modelo para tempos acelerados que inclui além da distribuição dos tempos até a falha uma relação entre esses tempos e os *stress* de aceleração, é

possível obter informações a respeito da confiabilidade dos componentes sob condições usuais de operação.

Representemos por T a variável aleatória tempo até a falha de um componente submetido a condições usuais de operação e por $T | \underline{v}$, a variável aleatória tempo até a falha, quando o componente é submetido à aceleração através de variáveis de *stress* $\underline{v} = (v_1, v_2, \dots, v_p)'$. Assumindo que o efeito dessa aceleração é multiplicativo sobre o tempo até a falha do componente, o tempo até a falha acelerada é dado por:

$$T | \underline{v} = \exp\left(\varphi\left(\underline{v}, \underline{\beta}\right)\right)T, \quad (1.16)$$

onde $\varphi\left(\underline{v}, \underline{\beta}\right)$ é uma função linear em $\underline{\beta}$, $\underline{v} = (v_1, v_2, \dots, v_p)'$ é um vetor de p covariáveis de aceleração (temperatura, voltagem, etc.), $\underline{\beta} = (\beta_1, \beta_2, \dots, \beta_p)'$ é um vetor dos correspondentes coeficientes. A função exponencial em (1.16) garante a positividade de $T | \underline{v}$.

A função do *stress*, dada por, $\exp\left(\varphi\left(\underline{v}, \underline{\beta}\right)\right)$ é incorporada na função de sobrevivência do tempo em condições usuais de modo que a função de sobrevivência dos tempos acelerados é dada por:

$$\begin{aligned} S_{T|\underline{v}}(t) &= P\left(T | \underline{v} > t\right) = P\left(\exp\left(\varphi\left(\underline{v}, \underline{\beta}\right)\right)T > t\right) \\ &= P\left(T > \exp\left(-\varphi\left(\underline{v}, \underline{\beta}\right)\right)t\right) = S_T\left(\exp\left(-\varphi\left(\underline{v}, \underline{\beta}\right)\right)t\right). \end{aligned} \quad (1.17)$$

Note que a função de sobrevivência de $T | \underline{v}$ é a função de sobrevivência de T avaliada no tempo acelerado pelo *stress*.

A relação entre as funções de risco de $T | \underline{v}$ e T é dada por:

$$\begin{aligned}
h_{T|v}(t) &= \frac{f_{T|v}(t)}{S_{T|v}(t)} = \frac{-\frac{\partial}{\partial t} S_{T|v}(t)}{S_{T|v}(t)} = \frac{-\frac{\partial}{\partial t} S_T \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right) t \right)}{S_T \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right) t \right)} \\
&= \frac{f_T \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right) t \right) \exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right)}{S_T \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right) t \right)} \\
&= h_T \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right) t \right) \exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta} \right) \right). \tag{1.18}
\end{aligned}$$

Estendendo (1.16) para o caso bivariado temos:

$$T_1 | \underset{\sim}{v} = \exp \left(\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_1 \right) \right) T_1 \quad \text{e} \quad T_2 | \underset{\sim}{v} = \exp \left(\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_2 \right) \right) T_2, \tag{1.19}$$

onde T_1 e T_2 são os tempos até as falhas dos componentes 1 e 2 respectivamente nas condições usuais. A relação entre as funções de sobrevivência dos tempos até as falhas em condições usuais e com aceleração é dada por,

$$\begin{aligned}
S_{T_1, T_2 | \underset{\sim}{v}}(t_1, t_2) &= P(T_1 | \underset{\sim}{v} > t_1, T_2 | \underset{\sim}{v} > t_2) \\
&= P \left(\exp \left(\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_1 \right) \right) T_1 > t_1, \exp \left(\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_2 \right) \right) T_2 > t_2 \right) \\
&= P \left(T_1 > \exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_1 \right) \right) t_1, T_2 > \exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_2 \right) \right) t_2 \right) \\
&= S_{T_1, T_2} \left(\exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_1 \right) \right) t_1, \exp \left(-\varphi \left(\underset{\sim}{v}, \underset{\sim}{\beta}_2 \right) \right) t_2 \right). \tag{1.20}
\end{aligned}$$

A função do *stress* pode assumir várias formas. Em particular, quando se considera um único fator de aceleração, geralmente são utilizadas as relações de potência inversa para voltagens e de Arrhenius para temperaturas, ou ainda, generalizações dessas relações.

Descrevemos a seguir, as relações de potência inversa (i) e de Arrhenius (ii) (Nelson, 1990).

i) Relação de Potência Inversa

A relação de potência inversa entre o tempo até a falha t_j do componente e a voltagem v_j é dada por:

$$t_j = \frac{A}{v_j^P} \quad A > 0, \quad (1.21)$$

onde A e P são parâmetros que caracterizam a geometria, tamanho, fabricação, método de teste entre outros fatores e dependem do modo de falha, sendo que os componentes com mais de um modo de falha apresentam diferentes valores de A e P para cada modo. Esta relação se verifica, por exemplo, em isolamentos elétricos e dielétricos, lâmpadas incandescentes e lâmpadas para flash. Em termos de variáveis aleatórias, usando (1.16) temos a seguinte relação para cada nível de voltagem:

$$T | v_j = \exp\left(\varphi\left(v_j, \beta\right)\right)T = \exp\left(\ln\beta_0 + \beta_1(-\ln v_j)\right)T = \beta_0 v_j^{-\beta_1} T, \quad (1.22)$$

onde β_0 e β_1 correspondem aos parâmetros A e P respectivamente em (1.21). Portanto, a função do *stress* para o modelo de potência inversa é dada por:

$$\exp\left(\varphi\left(v_j, \beta\right)\right) = \exp\left(\ln\beta_0 + \beta_1(-\ln v_j)\right) = \beta_0 v_j^{-\beta_1}, \quad (1.23)$$

onde $\varphi\left(v_j, \beta\right) = \ln\beta_0 + \beta_1(-\ln v_j)$.

ii) Relação de Arrhenius

O modelo de Arrhenius relaciona o tempo até a falha t_j com a temperatura τ_j da seguinte forma:

$$t_j = \exp\left(\frac{E}{k\tau_j}\right), \quad (1.24)$$

onde τ_j é a temperatura *Kelvin* absoluta, equivalente a temperatura *Celsius*+273°C e a temperatura *Fahrenheit*+459.7°F; k é a constante de *Boltzmann*, 8.6171×10^{-5} *elétron-volts* por °C; E é a energia de ativação da reação, geralmente em *elétron-volts*. Esta relação é

aplicada em testes para plásticos, lubrificantes, filamentos de lâmpadas incandescentes, entre outros. Para este modelo a função do *stress* apresenta a seguinte forma:

$$\exp(\varphi(\tau_j, \alpha)) = \exp(\alpha \tau_j^{-1}), \quad (1.25)$$

onde α corresponde a $\frac{E}{k}$ em (1.24).

Vejamos um caso univariado, onde T tem distribuição exponencial com parâmetro λ , $T \sim \text{Exp}(\lambda)$, e a relação entre o tempo até a falha e o *stress* é dada pelo modelo de potência inversa.

Exemplo. Suponha que a função de sobrevivência de T é dada por $S_T(t) = \exp(-\lambda t)$ e a função de sobrevivência de $T | v_j$, $j=1, 2, \dots, k$ obtida usando a relação em (1.17) é dada por:

$$\begin{aligned} S_{T|v_j}(t) &= S_T(\exp(-\varphi(v_j, \beta)))t \\ &= S_T(\exp(-\ln \beta_0 - \beta_1(-\ln v_j)))t \\ &= S_T(e^{-\ln \beta_0} v_j^{\beta_1} t) \\ &= \exp(\lambda \beta_0^{-1} v_j^{\beta_1} t) \\ &= \exp(\beta_0^* v_j^{\beta_1} t) \end{aligned}$$

onde $\beta_0^* = \lambda \beta_0^{-1}$, logo $T | v_j \sim \text{Exp}(\beta_0^* v_j^{\beta_1})$. Então, usando (1.18) obtemos a função de risco de $T | v_j$ dada por $h_{T|v_j}(t) = \beta_0^* v_j^{\beta_1}$.

Este exemplo mostra, portanto, que a forma da distribuição de $T | v_j$ para cada nível $j=1, 2, \dots, k$ de *stress* é a mesma de T .

1.3 Análise Clássica

1.3.1 Estimação dos Parâmetros usando o Método de Máxima Verossimilhança

Consideremos Y_1, Y_2, \dots, Y_n variáveis aleatórias independentes identicamente distribuídas com função densidade $f_{Y_i}(y; \theta)$ onde $\theta = (\theta_1, \dots, \theta_p)$ é um vetor de parâmetros desconhecidos assumindo valores em um conjunto $\Theta \subset \mathbb{R}^p$.

Obtidos os valores amostrais y_1, y_2, \dots, y_n a função $f_{Y_i}(y; \theta)$ pode ser considerada uma função dos parâmetros, denominada função de verossimilhança de θ dado y_1, y_2, \dots, y_n e apresenta a seguinte forma (veja, Lawless, 1982):

$$L(\theta | y) = f_{\underline{y}}(\underline{y}; \theta) = \prod_{i=1}^n f_{Y_i}(y_i; \theta). \quad (1.26)$$

No caso de variáveis aleatórias bivariadas $(Y_{11}, Y_{21}), \dots, (Y_{1n}, Y_{2n})$, independentes e identicamente distribuídas com função densidade conjunta $f_{\underline{y}_1, \underline{y}_2}(\underline{y}_1, \underline{y}_2; \theta)$ a função de verossimilhança é dada por:

$$L(\theta | \underline{y}_1, \underline{y}_2) = f_{\underline{y}_1, \underline{y}_2}(\underline{y}_1, \underline{y}_2; \theta) = \prod_{i=1}^n f_{Y_{1i}, Y_{2i}}(y_{1i}, y_{2i}; \theta). \quad (1.27)$$

Em análise de sobrevivência ou confiabilidade, o tempo até a falha pode ser censurado e neste caso, a função de verossimilhança é dada por:

$$L(\theta | t) = \prod_{i=1}^n f_{T_i}(t_i; \theta)^{\delta_i} S_{T_i}(t_i; \theta)^{1-\delta_i}, \quad (1.28)$$

onde $\delta_i = \begin{cases} 1, & \text{se } t_i \text{ é o tempo até a falha} \\ 0, & \text{se } t_i \text{ é o tempo da censura} \end{cases}$.

A análise de dados bivariados censurados já apresenta mais dificuldades. Consideremos por exemplo, que ambos os tempos possam ser censurados. Neste caso, os valores amostrais pertence a uma das 4 classes descritas a seguir:

C₁: t_{1i} e t_{2i} são tempos até as falhas;

C₂: t_{1i} é um tempo até a falha e t_{2i} é um tempo censurado (conhecemos apenas que $T_{2i} \geq t_{2i}$);

C₃: t_{2i} é um tempo até a falha e t_{1i} é um tempo censurado;

C₄: t_{1i} e t_{2i} são tempos censurados, $t_{1i} = t_{2i}$.

Dessa forma, a função de verossimilhança é dada por:

$$L\left(\underline{\theta} \mid \underline{t}_1, \underline{t}_2\right) = \left\{ \prod_{i \in C_1} f_{T_{1i}, T_{2i}}(t_{1i}, t_{2i}) \right\} \left\{ \prod_{i \in C_2} \frac{-\partial}{\partial t_{1i}} S_{T_{1i}, T_{2i}}(t_{1i}, t_{2i}) \right\} \\ \times \left\{ \prod_{i \in C_3} \frac{-\partial}{\partial t_{2i}} S_{T_{1i}, T_{2i}}(t_{1i}, t_{2i}) \right\} \left\{ \prod_{i \in C_4} S_{T_{1i}, T_{2i}}(t_{1i}, t_{2i}) \right\}. \quad (1.29)$$

Consideremos $(T_{1j} \mid v_j, T_{2j} \mid v_j), \dots, (T_{1n} \mid v_j, T_{2n} \mid v_j), j=1, 2, \dots, k$ os tempos bivariados até as falhas aceleradas por cargas de *stress* constantes, v_j , mais elevadas do que a usual. Assumindo que os tempos em cada nível j de *stress* são independentes, com densidade $f_{T_{1j}, T_{2j}}(t_{1i}, t_{2i}; \underline{\theta})$, onde $\underline{\theta} \in \Theta \subset R^p$, a função de verossimilhança é dada por:

$$L\left(\underline{\theta} \mid \underline{t}_1, \underline{t}_2\right) = \prod_{j=1}^k \prod_{i=1}^{n_j} f_{T_{1j}, T_{2j}}(t_{1i}, t_{2i}; \underline{\theta}), \quad (1.30)$$

O princípio da estimação de máxima verossimilhança (MV) consiste em obter os valores de $\underline{\theta}$ que maximizam a função de verossimilhança ou equivalentemente os valores que maximizam o logaritmo natural da função de verossimilhança, uma vez que a função log é crescente.

Em casos gerais, os estimadores de MV de $\underline{\theta}$, $\hat{\underline{\theta}}$ pode ser encontrado resolvendo as chamadas equações de verossimilhança, dadas por:

$$U_i(\underline{\theta})=0, \quad i=1, \dots, p, \quad (1.31)$$

onde

$$U_i(\underline{\theta}) = \frac{\partial}{\partial \theta_i} \log L\left(\underline{\theta} \mid \underline{y}\right), \quad i=1, \dots, p.$$

A função $U_i(\underline{\theta})$ é chamada *score* e o vetor $U(\underline{\theta})=(U_1(\underline{\theta}), \dots, U_p(\underline{\theta}))$ de vetor *score*.

Em algumas situações não é possível encontrar $\hat{\underline{\theta}}$ analiticamente sendo necessária a utilização de métodos numéricos. Também há casos onde o máximo não existe.

1.3.2 Propriedades dos Estimadores de MV

Sejam Y_1, Y_2, \dots, Y_n variáveis aleatórias independentes, com função de densidade $f_{Y_i}(y; \underline{\theta})$, $\underline{\theta} \in \Theta \subset R^p$. Sob as condições de regularidade, descritas em (i) e (ii) a seguir, o estimador de MV de $\underline{\theta}$, $\hat{\underline{\theta}}$ apresenta as seguintes propriedades assintóticas (veja Leite e Singer, 1990):

- 1) $\hat{\underline{\theta}}$ é um estimador consistente de $\underline{\theta}$.
- 2) $\sqrt{n}(\hat{\underline{\theta}} - \underline{\theta}) \xrightarrow{D} N_p(\underline{\theta}, I(\underline{\theta})^{-1})$, onde $I(\underline{\theta})^{-1}$ é a inversa da matriz de informação de Fisher definida na condição (ii) a) abaixo.

As condições de regularidade são:

- (i) Para $i, j = 1, \dots, p$, $\frac{\partial}{\partial \theta_i} f(y; \underline{\theta})$ e $\frac{\partial^2}{\partial \theta_i \partial \theta_j} f(y; \underline{\theta})$ existem em quase toda parte e são tais que $\left| \frac{\partial}{\partial \theta_i} f(y; \underline{\theta}) \right| \leq H_i(y)$ e $\left| \frac{\partial^2}{\partial \theta_i \partial \theta_j} f(y; \underline{\theta}) \right| \leq G_{ij}(y)$ onde $\int_R H_i(y) dy < \infty$ e $\int_R G_{ij}(y) dy < \infty$;
- (ii) Para $i, j = 1, \dots, p$, $\frac{\partial}{\partial \theta_i} f(y; \underline{\theta})$ e $\frac{\partial^2}{\partial \theta_i \partial \theta_j} f(y; \underline{\theta})$ existem em quase toda parte e são tais que:

- a) a matriz de informação de Fisher,

$$I(\underline{\theta}) = E \left\{ \left(\frac{\partial}{\partial \underline{\theta}} \log f(Y_1; \underline{\theta}) \right) \left(\frac{\partial}{\partial \underline{\theta}} \log f(Y_1; \underline{\theta}) \right)' \right\}, \quad (1.32)$$

onde $\left(\frac{\partial}{\partial \underline{\theta}} \log f(Y_1; \underline{\theta}) \right) = \left(\frac{\partial}{\partial \theta_1} \log f(Y_1; \underline{\theta}), \dots, \frac{\partial}{\partial \theta_p} \log f(Y_1; \underline{\theta}) \right)$ é finita e positiva definida.

$$b) E_{\theta} \left\{ \sup_{\{h: \|h\| \leq \delta\}} \left\| \frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y_i; \theta + h) - \frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y_i; \theta) \right\| \right\} = \Psi_{\delta}, \text{ onde}$$

$$\frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y_i; \theta) = \left[\frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y_i; \theta) \right]_{ij}, \text{ converge para zero com } \delta \rightarrow 0.$$

Um estimador consistente da matriz de informação de Fisher é dado pela matriz de informação observada, $I_o(\theta)$, com elementos

$$I_{o,ij} = \frac{1}{n} \frac{-\partial^2}{\partial \theta_i \partial \theta_j} \log L(\theta | t_1, t_2) \Bigg|_{\theta = \hat{\theta}}. \tag{1.33}$$

Portanto, usando as propriedades assintóticas dos estimadores de MV, os intervalos de confiança para θ_i são dados por:

$$IC(\theta_i) = \left(\hat{\theta}_i - z_{\alpha/2} \frac{I_{o,ii}^{-1}}{\sqrt{n}}; \hat{\theta}_i + z_{\alpha/2} \frac{I_{o,ii}^{-1}}{\sqrt{n}} \right), \tag{1.34}$$

e os intervalos de confiança aproximados para funções dos parâmetros $g(\theta)$, obtidos usando o método DELTA (veja, Leite e Singer, 1990), são dados por:

$$IC(g(\theta)) = \left(g(\hat{\theta}) - z_{\alpha/2} \frac{\sqrt{\hat{\text{var}} g(\theta)}}{\sqrt{n}}; g(\hat{\theta}) + z_{\alpha/2} \frac{\sqrt{\hat{\text{var}} g(\theta)}}{\sqrt{n}} \right), \tag{1.35}$$

onde

$$\hat{\text{var}}(g(\theta)) = \left(\frac{\partial}{\partial \theta_1} g(\theta), \frac{\partial}{\partial \theta_2} g(\theta), \dots, \frac{\partial}{\partial \theta_p} g(\theta) \right)' \Bigg|_{\theta = \hat{\theta}} I_o^{-1} \left(\frac{\partial}{\partial \theta_1} g(\theta), \frac{\partial}{\partial \theta_2} g(\theta), \dots, \frac{\partial}{\partial \theta_p} g(\theta) \right) \Bigg|_{\theta = \hat{\theta}}. \tag{1.36}$$

1.3.3 Diagnóstico de Normalidade Multivariada dos Estimadores de MV

Existem vários métodos que podem ser utilizados para verificar se a distribuição dos estimadores de MV de $\theta = (\theta_1, \theta_2, \dots, \theta_p)$ se aproxima da distribuição normal multivariada, os

quais, em geral, produzem medidas que são difíceis de calcular e interpretar. Devido a isto, Kass e Slate (1992) propuseram algumas medidas, baseadas nas terceiras derivadas da função de verossimilhança, avaliadas nas estimativas de MV que são fáceis de serem obtidas. Uma dessas medidas é dada por:

$$STD = \sum_{ijklmn} b_{ij} b_{im} b_{kn} d_{ijk} d_{lmn}, \quad (1.37)$$

onde, b_{ij} , $i, j=1, 2, \dots, p$, são os elementos da inversa da matriz de informação observada e

$$d_{ijk} = \frac{\partial^3}{\partial \theta_i \partial \theta_j \partial \theta_k} \log L(\underline{\theta} | \underline{y}) \Big|_{\underline{\theta} = \hat{\underline{\theta}}}$$

O valor de STD deve ser pequeno para que a distribuição dos estimadores seja normal. Para julgar o quanto pequeno deve ser esse valor, Kass e Slate (1992) acharam razoável considerar $STD < p \times (0,36)^2$, onde p é a dimensão do vetor de parâmetros.

1.4 Análise Bayesiana

A inferência Bayesiana é uma abordagem da Estatística que além de fazer uso de toda a informação contida nos dados, também leva em conta o grau de conhecimento do pesquisador, a respeito dos parâmetros antes da observação dos dados.

Nesta seção fazemos uma revisão dos principais conceitos de inferência Bayesiana e dos dois métodos de simulação iterativa que são úteis para esta abordagem, o amostrador de Gibbs e o algoritmo de Metropolis-Hastings (M-H), juntamente com o critério de convergência de Gelman e Rubin.

1.4.1 Principais Conceitos

Consideremos uma amostra aleatória de tamanho n , $\underline{Y} = (Y_1, Y_2, \dots, Y_n)'$, com função de densidade $f_{\underline{Y}}(\underline{y}; \underline{\theta})$, onde $\underline{\theta} = (\theta_1, \dots, \theta_p)'$ é um vetor de parâmetros.

Seja $\pi_i(\theta_i)$, $i=1, 2, \dots, p$, uma função de densidade *a priori* para θ_i , que representa as informações sobre θ_i antes da coleta dos dados. Assumindo independência das densidades *a priori*, a densidade *a priori* conjunta para $\underline{\theta}$ é dada por:

$$\pi(\underline{\theta}) = \pi_1(\theta_1) \pi_2(\theta_2) \dots \pi_p(\theta_p). \quad (1.38)$$

A densidade *a posteriori* de $\underline{\theta}$, que representa o conhecimento a respeito de $\underline{\theta}$ após a observação dos dados $\underline{y}=(y_1, y_2, \dots, y_n)'$, é obtida da fórmula de Bayes,

$$\pi(\underline{\theta} | \underline{y}) = \frac{\pi(\underline{\theta}) f_{\underline{Y}}(\underline{y} | \underline{\theta})}{f_{\underline{Y}}(\underline{y})}, \quad (1.39)$$

$$\text{onde } f_{\underline{Y}}(\underline{y}) = \begin{cases} \int f_{\underline{Y}}(\underline{y} | \underline{\theta}) \pi(\underline{\theta}) d\underline{\theta}, & \text{p/ } \underline{\theta} \text{ contínuo,} \\ \sum f_{\underline{Y}}(\underline{y} | \underline{\theta}) \pi(\underline{\theta}), & \text{p/ } \underline{\theta} \text{ discreto,} \end{cases} \quad (1.40)$$

e a soma ou a integral em (1.40) é calculada para o intervalo admissível de $\underline{\theta}$.

Como $f_{\underline{Y}}(\underline{y})$ é constante em relação a $\underline{\theta}$, então tem-se que,

$$\pi(\underline{\theta} | \underline{y}) = c \pi(\underline{\theta}) f_{\underline{Y}}(\underline{y} | \underline{\theta}),$$

onde c é uma constante necessária para que a soma ou integral da densidade *a posteriori* em (1.39) seja igual a 1.

A informação de $\underline{\theta}$ proveniente dos dados é contida na função de verossimilhança de $\underline{\theta}$. Dessa forma, a densidade *a posteriori* conjunta é proporcional ao produto entre a densidade *a priori* conjunta e a função de verossimilhança, ou seja,

$$\pi(\underline{\theta} | \underline{y}) \propto \pi(\underline{\theta}) L(\underline{\theta} | \underline{y}). \quad (1.41)$$

A partir da densidade *a posteriori* conjunta são obtidas as densidades marginais *a posteriori* de θ_i , dadas por:

$$\pi(\theta_i | \underline{y}) = \int \dots \int \pi(\underline{\theta} | \underline{y}) d\theta_{\sim i}, \quad (1.42)$$

onde $\theta_{\sim i}$ é o vetor $\underline{\theta}$ sem o i -ésimo elemento.

As inferências sobre os parâmetros são feitas através das densidades marginais *a posteriori*. As informações nelas contidas podem ser resumidas através de medidas de locação, como a média e mediana, medidas de dispersão, como a variância e desvio padrão e também pelos intervalos Bayesianos (veja Box e Tiao, 1973).

Os conceitos apresentados para o caso univariado podem ser estendidos para mais de uma variável. Por exemplo, para o caso bivariado tem-se que

$$\pi(\underline{\theta} | (y_1, y_2)) \propto \pi(\underline{\theta}) L(\underline{\theta} | (y_1, y_2)), \quad (1.43)$$

onde $(y_1, y_2) = ((y_{11}, y_{21}), \dots, (y_{1n}, y_{2n}))$.

Em alguns modelos as integrais em (1.42) podem não apresentar solução analítica, sendo necessário o uso de métodos de aproximações ou numéricos, que por sua vez, em alguns modelos, são de difícil implementação. Nestes casos, métodos de simulação iterativa podem ser usados, entre eles os algoritmos de Gibbs e de M-H.

1.4.2 Métodos de Simulação Iterativa usados na Inferência Bayesiana

Utilizando os métodos de simulação iterativa é possível gerar observações de uma distribuição marginal indiretamente, sem precisar calcular a sua densidade.

Os métodos de simulação iterativa mais utilizados na inferência Bayesiana são, o amostrador de Gibbs e o algoritmo M-H. O amostrador de Gibbs foi proposto por Geman e Geman (1984) e o algoritmo M-H foi desenvolvido por Metropolis, Rosenbluth, Rosenbluth, Teller e Teller (1953) e generalizado por Hastings (1970). Posteriormente, os dois métodos foram discutidos com mais detalhes por Casella e George (1992), que estudaram o amostrador de Gibbs e por Chib e Greenberg (1995) que descreveram o algoritmo M-H.

O amostrador de Gibbs consiste em amostrar de uma cadeia de Markov, cujo núcleo de transição é dado pelas distribuições condicionais completas. O algoritmo do amostrador de Gibbs é apresentado a seguir (veja Gamerman, 1996):

i) inicializar o contador de iterações da cadeia, $j=1$, e arbitrar valores iniciais

$$\underline{\theta}^{(0)} = (\theta_1^{(0)}, \dots, \theta_p^{(0)});$$

ii) obter um novo valor $\underline{\theta}^{(j)} = (\theta_1^{(j)}, \dots, \theta_p^{(j)})$ gerado das distribuições condicionais

$$\theta_1^{(j)} \sim f(\theta_1 | \theta_2^{(j-1)}, \dots, \theta_p^{(j-1)})$$

$$\theta_2^{(j)} \sim f(\theta_2 | \theta_1^{(j-1)}, \theta_3^{(j-1)}, \dots, \theta_p^{(j-1)})$$

⋮

$$\theta_p^{(j)} \sim f(\theta_p | \theta_1^{(j-1)}, \dots, \theta_{p-1}^{(j-1)});$$

(1.44)

iii) mudar o contador j para $j+1$ até obter o tamanho de amostra desejado.

No caso das distribuições condicionais completas serem desconhecidas, associa-se ao amostrador de Gibbs, o algoritmo M-H.

Para implementar o algoritmo M-H é necessário especificar uma densidade que gere os candidatos à distribuição de interesse. Uma possibilidade seria escrever as distribuições condicionais da seguinte forma: (veja Chib e Greenberg, 1995)

$$f(\theta_1 | \theta_2, \theta_3, \dots, \theta_p) \propto \psi_1(\theta_1, \theta_2, \dots, \theta_p) h_i(\theta_1)$$

$$f(\theta_2 | \theta_1, \theta_3, \dots, \theta_p) \propto \psi_2(\theta_1, \theta_2, \dots, \theta_p) h_i(\theta_2)$$

⋮

$$f(\theta_p | \theta_1, \theta_2, \dots, \theta_{p-1}) \propto \psi_p(\theta_1, \theta_2, \dots, \theta_p) h_i(\theta_p),$$

(1.45)

onde $\psi_i(\theta_1, \theta_2, \dots, \theta_p)$ é uniformemente limitada e $h_i(\theta_i)$ é uma densidade que pode ser amostrada, denominada núcleo de transição.

Através de $h_i(\theta_i)$ são gerados os candidatos que serão aceitos com uma certa probabilidade de movimento, dada por:

$$\alpha_i(\theta_i^{(j)}, \theta_i^{(cand)}) = \min\left(\frac{\psi_i(\theta_i^{(cand)}, \underline{\theta}_{-i})}{\psi_i(\theta_i^{(j)}, \underline{\theta}_{-i})}, 1\right) \quad (1.46)$$

onde $\underline{\theta}_{-i}$ é o vetor $\underline{\theta}$ sem o i -ésimo elemento.

O algoritmo de M-H (veja Gamerman, 1996) é apresentado a seguir:

i) inicializar o contador de iterações da cadeia, $j=1$, e arbitrar valores iniciais,

$$\underline{\theta}^{(0)} = (\theta_1^{(0)}, \dots, \theta_k^{(0)});$$

- ii) inicializar o contador de elementos de θ , $i=1$;
- iii) gerar da distribuição $h_i(\theta_i)$ um candidato para θ_i , dado por $\theta_i^{(cand)}$;
- iv) calcular a probabilidade de movimento $\alpha_i(\theta_i^{(j-1)}, \theta_i^{(cand)})$;
- v) gerar $u \sim U(0,1)$. Aceitar o candidato se $u \leq \alpha_i$, $\theta_i^{(j)} = \theta_i^{(cand)}$, caso contrário $\theta_i^{(j)} = \theta_i^{(j-1)}$;
- vi) repetir iii), iv), v) para $i=2$ até p ;
- vii) mudar o contador de j para $j+1$ e retornar a ii) até obter o tamanho de amostra desejado.

1.3.3 Critério de Convergência de Gelman e Rubin

Para que as amostras, geradas pelos métodos de simulação iterativa, possam ser utilizadas para inferência é necessário verificar se as mesmas estão convergindo para a distribuição de equilíbrio. Existem alguns procedimentos gráficos para monitorar a convergência, por exemplo, a similaridade da trajetória da cadeia, ao longo das iterações, é um forte indício de convergência. Uma outra indicação de convergência é dada pela baixa autocorrelação entre os elementos da cadeia. Existem também inúmeros critérios para diagnosticar convergência, sendo os mais utilizados, os propostos por Gelman e Rubin (1992), Geweke (1992), Raftery e Lewis (1992) e Heidelberger e Welch (1983).

Nesta seção apresentamos o método proposto por Gelman e Rubin (1992). Os autores sugerem utilizar várias cadeias paralelas, inicializadas em pontos distintos. O método é baseado na comparação das variâncias dentro e entre as cadeias para cada parâmetro. Esta comparação é utilizada para estimar o quanto o parâmetro de escala da distribuição de interesse pode ser reduzido quando a cadeia vai para o infinito. Para calcular esse fator de redução deve-se obter para cada variável:

1. A variância entre as m médias das cadeias $\{\theta_{i1}, \theta_{i2}, \dots, \theta_{im}\}$, $i=1, 2, \dots, m$, dada por:

$$\frac{E}{n} = \sum_{i=1}^m \frac{(\bar{\theta}_i - \bar{\theta}_{..})^2}{(m-1)},$$

$$\text{onde } \bar{\theta}_i = \sum_{j=1}^n \theta_{ij} \text{ e } \bar{\theta}_{..} = \sum_{i=1}^m \sum_{j=1}^n \theta_{ij}.$$

2. A média das m variâncias dentro das cadeias, s_i^2 , cada uma delas baseada em $n-1$ graus de liberdade, dada por:

$$D = \sum_{i=1}^m \frac{s_i^2}{m}.$$

3. Uma estimativa da média da distribuição de interesse, dada pela média amostral dos $m \times n$ valores simulados de θ_i , $\hat{\mu} = \bar{\theta}_{..}$.
4. Uma estimativa não viciada da variância da distribuição de interesse, dada pela média ponderada entre as variâncias dentro e entre as cadeias, dada por:

$$\hat{\sigma}^2 = \left(\frac{(n-1)}{n} \right) D + \frac{1}{n} E.$$

5. Os parâmetros de escala $\sqrt{\hat{V}} = \sqrt{\hat{\sigma}^2 + \frac{E}{mn}}$ e graus de liberdade $gl = \frac{2\hat{V}^2}{\hat{\text{var}}(\hat{V})}$ da distribuição t de Student aproximada para θ_i , onde

$$\begin{aligned} \hat{\text{var}}(\hat{V}) = & \left(\frac{(n-1)}{n} \right)^2 \frac{1}{m} \hat{\text{var}}(s_i^2) + \left(\frac{(m+1)}{mn} \right)^2 \frac{2}{(m-1)} E^2 + 2 \left(\frac{(m+1)(n-1)}{mn^2} \right) \\ & \times \frac{n}{m} (\text{cov}(s_i^2, \bar{\theta}_{..}^2) - 2\bar{\theta}_{..} (\text{cov}(s_i^2, \bar{\theta}_{..}))), \end{aligned}$$

e as variâncias e covariâncias estimadas são obtidas dos m valores de $\bar{\theta}_{..}$ e s_i^2 .

6. Finalmente, o fator de redução do parâmetro de escala é dado por:

$$\sqrt{\hat{R}} = \sqrt{\left(\frac{\hat{V}}{\hat{W}} \frac{gl}{gl-2} \right)} \quad (1.47)$$

que converge para 1 quando $n \rightarrow \infty$.

Uma vez que \hat{R} está próximo de 1 para todas as variáveis de interesse, os autores sugerem utilizar a segunda metade das observações de cada cadeia para as inferências.

CAPÍTULO II

ANÁLISE CLÁSSICA DA DISTRIBUIÇÃO EBVE

Este capítulo é dividido em duas seções. Na seção 2.1 geramos amostras da distribuição EBVE, usando o método da rejeição e fazemos um estudo de simulação para verificar a validade das amostras geradas. Na seção 2.2 estimamos os parâmetros da distribuição EBVE, utilizando o método de MV, considerando amostras com dados não censurados e com censura do tipo I, ou seja, quando o tempo para realização do experimento é fixado previamente. Em ambos os casos, fazemos simulações, considerando amostras de vários tamanhos, visando estudar as propriedades assintóticas dos estimadores de MV.

2.1 Geração de Dados da Distribuição EBVE

Existem vários métodos de simulação estocástica que são usados para gerar quantidades de uma distribuição de interesse. Entre eles, o método da rejeição tem sido bastante utilizado. Este método foi descrito inicialmente por von Neumann (1951) e algumas modificações deste método apareceram posteriormente na literatura.

Nesta seção geramos amostras da distribuição EBVE, usando a versão do método da rejeição descrita por Kennedy (1980). A seção é dividida em duas partes, sendo que na primeira mostramos como obter as amostras e na segunda apresentamos um estudo de simulação para verificar a qualidade das amostras geradas.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

2.1.1 Geração de Dados da Distribuição EBVE usando o Método da Rejeição

O método da rejeição (veja Kennedy, 1980) consiste em:

- 1) gerar x de uma distribuição $q(x)$;
- 2) gerar u de uma $U(0, 1)$;
- 3) aceitar x se $uM(x) \leq f(x)$, onde $M(x)$ é uma função integrável, conhecida como envelope superior, tal que, $M(x) \geq f(x)$; caso contrário retornar ao passo 1).

Para gerar dados da densidade da EBVE, dada por $f_{T_1, T_2}(t_1, t_2)$ em (1.13), consideramos x no passo 1) acima, um vetor com dois elementos (x_1, x_2) e q uma distribuição uniforme em $G = [a_1, b_1] \times [a_2, b_2]$. Uma vez que $(X_1, X_2) \sim U(G)$ então X_1 e X_2 são independentes (James, 1981). Dessa forma, geramos x_1 de $U=[a_1, b_1]$ e x_2 de $U=[a_2, b_2]$.

A probabilidade de rejeição dos candidatos é menor quando as funções M e f em 3) estão próximas. Entretanto, atualmente com a rapidez dos computadores este fato torna-se menos importante e assim sendo, escolhemos $M(x_1, x_2)$ igual a uma constante, garantindo facilmente o envelope para f . Essa constante foi escolhida baseada na análise dos gráficos da densidade bivariada, apresentados na Figura 2.1. Portanto, para os parâmetros considerados, $\lambda_1 = \lambda_2 = 0,1$, $\lambda_{12} = 0,2$, $s_1 = s_2 = 0,5$ utilizamos $M(x_1, x_2) = 0,046$. Os valores $a_1 = a_2 = 0$ e $b_1 = b_2 = 35$ também foram escolhidos baseados nos gráficos da Figura 2.1.

Como a densidade da EBVE é definida de forma diferente para $t_1 > t_2$ e $t_1 \leq t_2$, utilizamos a variável $ber \sim \text{bernoulli}(0,5)$ para classificar os pares.

Resumindo, o programa para a geração de (T_1, T_2) , apresentado no Apêndice B, é feito do algoritmo mostrado a seguir.

1. Gerar $x_1 \sim U(0, 35)$ e $x_2 \sim U(0, 35)$.
 2. Gerar $ber \sim \text{bernoulli}(0,5)$
 - se $ber = 0$ usar o par $(\min(x_1, x_2), \max(x_1, x_2)) = (c_1, c_2)$
 - se $ber = 1$ usar o par $(\max(x_1, x_2), \min(x_1, x_2)) = (c_1, c_2)$ como candidato.
 3. Gerar $u \sim U(0, 0,046)$.
- Se $u < f(c_1, c_2)$ aceitar o candidato, caso contrário retornar ao passo 1.

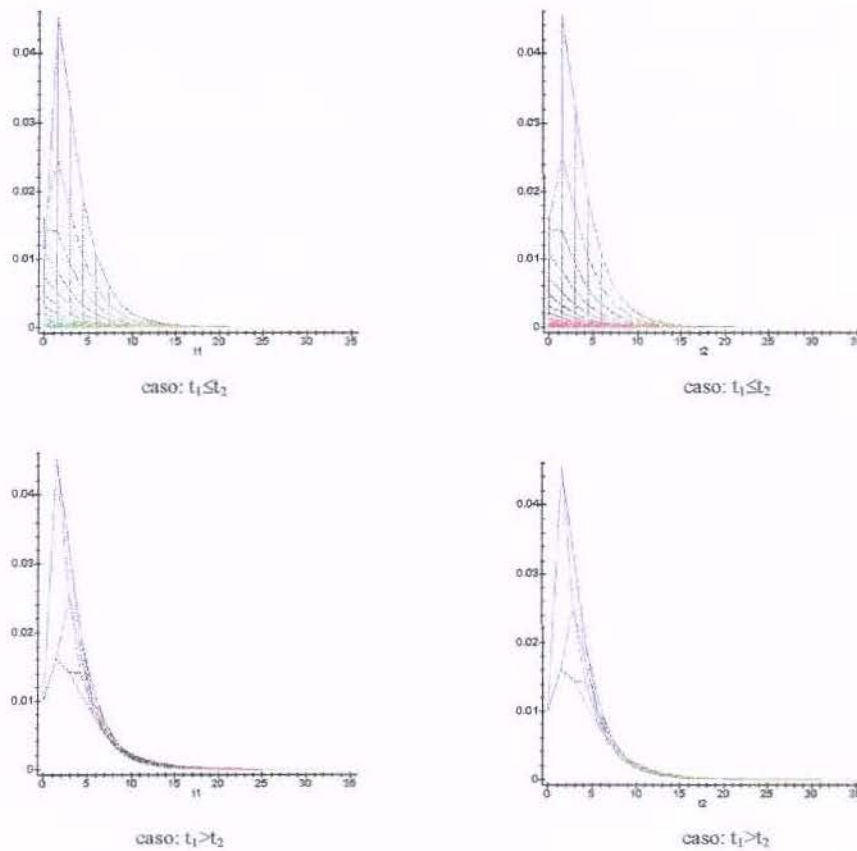


Figura 2.1 - Gráficos da densidade EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$ (interseções no plano $t_1=0$ e $t_2=0$).

2.1.2 Verificação da Validade das Amostras Geradas

Nesta seção apresentamos um estudo de simulação para verificarmos a qualidade das amostras geradas pelo método da rejeição, descrito anteriormente.

Inicialmente calculamos os valores teóricos, esperança, variância e correlação da distribuição, considerando os parâmetros utilizados na geração, ou seja, $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$, $s_1=s_2=0,5$. Os resultados foram obtidos numericamente e se encontram na Tabela 2.1.

Para verificar a eficiência do método da rejeição, geramos 500 amostras considerando vários tamanhos, $n=20, 30, 50, 100, 200, 300$ e 500 e para cada uma delas calculamos as médias de t_1 , t_2 , $\min(t_1, t_2)$, as variâncias e a correlação entre t_1 e t_2 . Na Tabela 2.2 são apresentadas as médias dos valores obtidos nas 500 amostras e o tempo gasto nas simulações, usando um processador *Pentium II* – 350 MHz com 128 MB de RAM.

Tabela 2.1 - Resumo da distribuição EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$

$E(T_1)$	$E(T_2)$	$E \min(T_1, T_2)$	$Var(T_1)$	$Var(T_2)$	$Corr(T_1, T_2)$
4,31	4,31	2,89	12,83	12,83	0,32

Tabela 2.2 – Média das médias, variâncias e correlações dos tempos (t_1, t_2) das 500 amostras geradas da EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$

n	médias t_1 média (dp)	médias t_2 média (dp)	médias $\min(t_1, t_2)$ média (dp)	var t_1 média (dp)	var t_2 média (dp)	corr(t_1, t_2) média (dp)	tempo
20	4,28 (0,77)	4,29 (0,80)	2,89 (0,57)	11,70 (6,43)	12,08 (7,15)	0,32 (0,26)	1'01
30	4,28 (0,61)	4,30 (0,65)	2,89 (0,47)	12,09 (5,43)	12,40 (5,89)	0,32 (0,21)	1'32
50	4,28 (0,47)	4,29 (0,51)	2,88 (0,35)	12,27 (4,18)	12,64 (4,88)	0,32 (0,16)	2'33
100	4,29 (0,34)	4,30 (0,36)	2,88 (0,24)	12,53 (3,21)	12,69 (3,47)	0,33 (0,12)	5'05
200	4,30 (0,25)	4,31 (0,24)	2,88 (0,18)	12,75 (2,34)	12,71 (2,30)	0,32 (0,09)	10'11
300	4,31 (0,20)	4,31 (0,20)	2,89 (0,15)	12,73 (1,92)	12,73 (1,84)	0,32 (0,08)	15'15
500	4,30 (0,16)	4,30 (0,16)	2,89 (0,12)	12,71 (1,48)	12,73 (1,43)	0,32 (0,06)	25'22

Observamos que as médias das médias, variâncias e correlações dos dados gerados estão próximas dos valores da distribuição e que os desvios padrões dessas medidas parecem pequenos, principalmente para $n \geq 100$.

Mais precisamente, considerando que a distribuição das médias amostrais se aproxima da distribuição normal, podemos interpretar os resultados da Tabela 2.2 como segue: por exemplo, para $n=200$, aproximadamente 68% das médias de t_1 , das 500 amostras geradas, estão dentro do intervalo $(\bar{t}_1 - s, \bar{t}_1 + s) = (4,05, 4,55)$, ou ainda, que 95% das médias de t_1 estão dentro do intervalo $(\bar{t}_1 - 2s, \bar{t}_1 + 2s) = (3,80, 4,80)$ onde \bar{t}_1 e s são respectivamente, a média e o desvio padrão das 500 médias de t_1 .

2.2 Análise de Dados Bivariados sem Censura da Distribuição EBVE

Nesta seção encontramos as estimativas de MV dos parâmetros da distribuição EBVE, usando dados sem censura e estudamos as propriedades assintóticas desses estimadores através de simulações, considerando amostras de vários tamanhos.

2.2.1 Estimação dos Parâmetros Usando o Método de MV e suas Propriedades

Seja $\left(\underset{\sim}{T}_1, \underset{\sim}{T}_2\right) = ((T_{11}, T_{21}), \dots, (T_{1n}, T_{2n}))$, uma amostra aleatória dos tempos até as falhas de n pares de componentes, com função de distribuição EBVE.

Obtidos os valores amostrais, o logaritmo da função de verossimilhança dos parâmetros $(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2)$ é dado por:

$$\log L\left(\underset{\sim}{\theta} \mid \underset{\sim}{t}_1, \underset{\sim}{t}_2\right) = \sum_{i: t_{1i} \leq t_{2i}} \left\{ \begin{aligned} & \exp\left(-(\lambda_1 + \lambda_{12})t_{1i} - \lambda_2 t_{2i} + \frac{\lambda_{12}}{s_1}(1 - e^{-s_1(t_{1i} - t_{2i})}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_1(t_{1i} - t_{2i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(\lambda_1 + \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} - \frac{\lambda_{12}}{s_1 + s_2}(s_2 e^{-s_2(t_{2i} - t_{1i})} + s_1 e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(\lambda_2 + \lambda_{12} - \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} + \frac{s_2 \lambda_{12}}{s_1 + s_2}(e^{-s_2(t_{2i} - t_{1i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & + \exp\left(-(\lambda_1 + \lambda_{12})t_{1i} - \lambda_2 t_{2i} + \frac{\lambda_{12}}{s_1}(1 - e^{-s_1(t_{1i} - t_{2i})}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_1(t_{1i} - t_{2i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(s_2 \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} - \frac{s_2 \lambda_{12}}{s_1 + s_2}(s_2 e^{-s_2(t_{2i} - t_{1i})} + s_1 e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \end{aligned} \right\} + \sum_{i: t_{1i} > t_{2i}} \left\{ \begin{aligned} & \exp\left(-\lambda_1 t_{1i} - (\lambda_2 + \lambda_{12})t_{2i} + \frac{\lambda_{12}}{s_2}(1 - e^{-s_2(t_{2i} - t_{1i})}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_2(t_{2i} - t_{1i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(\lambda_1 + \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} - \frac{\lambda_{12}}{s_1 + s_2}(s_2 e^{-s_2(t_{2i} - t_{1i})} + s_1 e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(s_2 \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} - \frac{s_2 \lambda_{12}}{s_1 + s_2}(s_2 e^{-s_2(t_{2i} - t_{1i})} + s_1 e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & + \exp\left(-\lambda_1 t_{1i} - (\lambda_2 + \lambda_{12})t_{2i} + \frac{\lambda_{12}}{s_2}(1 - e^{-s_2(t_{2i} - t_{1i})}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_2(t_{2i} - t_{1i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \\ & \times \left(\lambda_2 + \lambda_{12} - \lambda_{12} e^{-s_2(t_{2i} - t_{1i})} + \frac{s_2 \lambda_{12}}{s_1 + s_2}(e^{-s_2(t_{2i} - t_{1i})} - e^{-s_1 t_{1i} - s_2 t_{2i}})\right) \end{aligned} \right\}. \tag{2.1}$$

Os estimadores de MV, neste caso, não podem ser obtidos analiticamente e dessa forma, maximizamos a função (2.1) através do método numérico de *quasi-Newton* BFGS, veja Fletcher (1987), já implementado em Ox, (Doornik, 1999).

2.2.2.1 Exemplo de Aplicação

Primeiramente geramos uma amostra de tamanho $n=500$, da distribuição EBVE com parâmetros $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$ e $s_1=s_2=0,5$, usando o método da rejeição, apresentado na seção 2.1.

As estimativas de MV e os intervalos de confiança aproximados, obtidos usando (1.34) são apresentados na Tabela 2.3, abaixo.

Tabela 2.3 Estimativas de MV e intervalos de confiança aproximados dos parâmetros da EBVE, no exemplo de aplicação

	EMV	IC 95%
λ_1	0,092	(0,067; 0,118)
λ_2	0,099	(0,074; 0,125)
λ_{12}	0,201	(0,164; 0,238)
s_1	0,567	(0,356; 0,778)
s_2	0,540	(0,333; 0,748)

Calculamos as estimativas de MV das funções marginais de confiabilidade e de risco, avaliadas no tempo $t=5$, para cada uma das partes, específica e comum dos componentes. Para isso, substituímos respectivamente, nas expressões (1.5), (1.6), (1.7) e (1.8) as estimativas de MV dos parâmetros, apresentadas na Tabela 2.3, acima. As estimativas encontradas e os intervalos de confiança obtidos usando (1.35) são mostrados na Tabela 2.4.

Também obtivemos as estimativas dos percentis (t_p) , $p \times 100$, das distribuições marginais dos tempos, igualando-as a p e resolvendo, numericamente, essas equações em t . Na Tabela 2.5, apresentamos as estimativas dos quartis das distribuições marginais de T_1 e T_2 .

Tabela 2.4 Estimativas de MV e intervalos de confiança aproximados, das funções marginais de confiabilidade e de risco, avaliadas no tempo $t=5$, correspondentes ao exemplo de aplicação

<i>tempo</i>	$S(t)$	IC 95%	$h(t)$	IC 95%
X_1	0,630	(0,551; 0,710)	0,092	(0,067; 0,118)
Z_1	0,511	(0,450; 0,572)	0,189	(0,158; 0,221)
T_1	0,332	(0,290; 0,354)	0,282	(0,256; 0,307)
X_2	0,609	(0,530; 0,687)	0,099	(0,074; 0,125)
Z_2	0,518	(0,457; 0,579)	0,187	(0,157; 0,218)
T_2	0,315	(0,283; 0,348)	0,287	(0,261; 0,313)

Tabela 2.5 Estimativas dos quartis das distribuições marginais de T_1 e T_2 , correspondentes ao exemplo de aplicação

componente i	$100 \times p$	t_p
1	25	1,738
	50	3,395
	75	5,892
2	25	1,703
	50	3,344
	75	5,801

2.2.1.2 Simulações para Estudar as Propriedades dos Estimadores de MV

Nesta seção apresentamos os resultados obtidos em um estudo de simulação, feito com 500 amostras de tamanhos $n=30, 50, 100, 200, 300, 500$, da distribuição EBVE com parâmetros $\lambda_1 = \lambda_2 = 0,1$, $\lambda_{12} = 0,2$ e $s_1 = s_2 = 0,5$, geradas pelo método da rejeição, descrito na seção 2.1.1. Para cada uma das amostras foram obtidas as estimativas de MV, os desvios padrões, os erros quadráticos médios dos estimadores, a medida *STD* (seção 1.3.3) e os

intervalos de confiança aproximados, com coeficientes nominais 90% e 95%. Neste estudo, foram consideradas as amostras cujas estimativas pertenceram aos intervalos: $\hat{\lambda}_1 < 1$, $\hat{\lambda}_2 < 1$, $\hat{\lambda}_{12} < 2$, $\hat{s}_1 < 5$ e $\hat{s}_2 < 5$. O total de amostras que não convergiram ou não satisfizeram as condições acima, para os tamanhos de amostras considerados, são apresentados na Tabela 2.6. O programa utilizado encontra-se no Apêndice C.

Na Tabela 2.7 apresentamos as médias das 500 estimativas de MV. Os valores encontrados estão próximos dos apresentados por Ryu (1993), Tabela 1. Ainda na Tabela 2.7, se encontram as médias dos desvios padrões e dos erros quadráticos médios dos estimadores e nas Figuras 2.2 e 2.3 mostramos os gráficos referentes a essa tabela. Observamos que os erros quadráticos médios de λ_1 e λ_2 estão próximos de zero, mesmo para amostras com $n=30$, enquanto que o estimadores de λ_{12} parecem apresentar vícios e variabilidade pequenos para amostras maiores do que 100 e os de s_1 e s_2 para amostras com $n>300$.

Na Tabela 2.8 se encontram as coberturas dos intervalos com 90% e 95% de confiança, as quais não parecem próximas das coberturas nominais, ainda para $n=500$.

Na Tabela 2.9 apresentamos algumas razões entre os erros quadráticos médios e verificamos que a eficiência dos estimadores aumenta consideravelmente, com o aumento do tamanho das amostras.

As médias e os desvios padrões das medidas *STD*, apresentadas na Tabela 2.10, estão diminuindo a medida que o tamanho das amostras aumenta, como era esperado. Entretanto, mesmo para amostras de tamanho 500, a média de *STD* $> 0,648$, indica que a distribuição dos estimadores de MV parece não se aproximar da normal multivariada (veja seção 1.3.3). Estudamos também as distribuições univariadas dos estimadores. Nas Figuras 2.4 e 2.5 estão os histogramas e na Figura 2.6, os gráficos normais probabilísticos das estimativas de MV. A Tabela 2.11 contém os resultados do teste de Ryan Joiner, que é similar ao teste de Shapiro Wilk (veja Lawless, 1982), para verificar a normalidade univariada. Mesmo considerando amostras de tamanho 500, o *p-value* para s_1 e s_2 , foi menor do que 0,01 e portanto, ainda para um nível de significância $\alpha=0,01$ a hipótese de normalidade é rejeitada (α é um valor fixado previamente que indica a probabilidade de rejeitar a hipótese de normalidade quando a distribuição é de fato normal).

Tabela 2.6 – Descrição das amostras simuladas com dados sem censura

condições	tamanho das amostras					
	30	50	100	200	300	500
$\lambda_1 > 1$	0*	0	0	0	0	0
$\lambda_2 > 1$	0	0	0	0	0	0
$\lambda_{12} > 2$	8	3	0	0	0	0
$s_1 > 5$	23	9	1	0	0	0
$s_2 > 5$	24	11	2	0	0	0
$\lambda_1, \lambda_2 > 1$ ou $\lambda_{12} > 2$ ou $s_1, s_2 > 5$	51	22	3	0	0	0
não convergiram	318	107	23	1	0	0
covariância negativa	0	0	0	0	0	0

*Número de amostras

Tabela 2.7 Médias das 500 estimativas de MV, dos desvios padrões (dp) e dos erros quadráticos médios (eqm) dos estimadores dos parâmetros da distribuição EBVE com $\lambda_1 = \lambda_2 = 0.1$, $\lambda_{12} = 0.2$, $s_1 = s_2 = 0.5$, considerando dados sem censura.

média (dp)	tamanho das amostras					
	30	50	100	200	300	500
$\hat{\lambda}_1$	0,095 (0,046)	0,097 (0,041)	0,095 (0,033)	0,098 (0,024)	0,097 (0,019)	0,099 (0,014)
$\hat{\lambda}_2$	0,092 (0,049)	0,090 (0,041)	0,096 (0,033)	0,098 (0,023)	0,099 (0,019)	0,099 (0,014)
$\hat{\lambda}_{12}$	0,268 (0,191)	0,244 (0,108)	0,225 (0,063)	0,211 (0,038)	0,208 (0,030)	0,206 (0,022)
\hat{s}_1	0,762 (0,767)	0,629 (0,584)	0,583 (0,385)	0,537 (0,215)	0,530 (0,161)	0,510 (0,116)
\hat{s}_2	0,794 (0,814)	0,683 (0,606)	0,580 (0,398)	0,530 (0,214)	0,519 (0,166)	0,508 (0,113)
dp λ_1	0,056 (0,018)	0,044 (0,013)	0,032 (0,008)	0,022 (0,004)	0,018 (0,003)	0,014 (0,001)
dp λ_2	0,058 (0,028)	0,046 (0,017)	0,032 (0,007)	0,022 (0,004)	0,018 (0,002)	0,014 (0,001)
dp λ_{12}	0,255 (1,23)	0,101 (0,238)	0,056 (0,036)	0,035 (0,008)	0,028 (0,005)	0,021 (0,003)
dp s_1	0,667 (0,815)	0,465 (0,592)	0,303 (0,286)	0,188 (0,090)	0,149 (0,054)	0,110 (0,029)
dp s_2	0,720 (1,10)	0,500 (0,591)	0,316 (0,532)	0,185 (0,099)	0,146 (0,056)	0,109 (0,027)
eqm λ_1	0,006 (0,004)	0,004 (0,003)	0,002 (0,002)	0,001 (0,001)	0,001 (0,001)	4×10^{-4} (3×10^{-4})
eqm λ_2	0,007 (0,009)	0,004 (0,004)	0,002 (0,002)	0,001 (0,001)	0,001 (0,001)	4×10^{-4} (3×10^{-4})
eqm λ_{12}	1,62 (23,96)	0,081 (1,06)	0,009 (0,027)	0,003 (0,003)	0,002 (0,002)	0,001 (0,001)
eqm s_1	1,77 (5,22)	0,924 (3,74)	0,329 (1,23)	0,091 (0,165)	0,052 (0,074)	0,026 (0,031)
eqm s_2	2,47 (13,87)	1,00 (3,42)	0,548 (5,68)	0,091 (0,213)	0,052 (0,082)	0,025 (0,029)
tempo simulações.	2:01'25''	1:29'51''	2:26'09''	4:27'52''	6:33'17''	15:27'03''

Tabela 2.8 Cobertura dos intervalos de confiança aproximados, com coeficientes 90% e 95%, usando dados sem censura

n	λ_1	λ_2	λ_{12}	s_1	s_2
30	93,0 ^a	91,8	95,2	85,6	86,4
	96,8 ^b	95,0	96,4	88,4	89,6
50	89,4	88,8	92,8	83,0	88,6
	95,0	94,4	96,6	85,8	91,2
100	86,8	88,4	90,0	85,8	87,2
	93,0	93,8	95,2	89,0	91,0
200	89,4	87,4	89,6	86,4	86,6
	94,4	92,4	95,4	92,0	90,0
300	88,2	87,2	89,6	87,8	86,6
	93,0	91,6	94,4	92,6	91,8
500	88,4	90,6	88,8	88,6	89,4
	94,2	95,0	95,6	92,0	93,4

^{a,b} Coeficientes de confiança 90% e 95%, respectivamente

Tabela 2.9 Razões entre os erros quadráticos médios das estimativas de MV

	$\frac{eqm(n=30)}{eqm(n=50)}$	$\frac{eqm(n=30)}{eqm(n=100)}$	$\frac{eqm(n=30)}{eqm(n=500)}$
λ_1	1,44	2,46	14,12
λ_2	1,58	3,03	17,00
λ_{12}	20,04	180,6	1655,0
s_1	1,91	5,37	66,74
s_2	2,47	4,50	96,87

Tabela 2.10 Diagnóstico de normalidade multivariada dos estimadores de MV, considerando dados sem censura

n	STD média (dp)
30	3345,5 (49642,0)
50	167,78 (2964,8)
100	13,1 (71,6)
200	3,32 (2,09)
300	2,06 (1,17)
500	1,12 (0,28)

Tabela 2.11 Resultados do teste de Ryan-Joiner para verificar normalidade univariada

	<i>n</i> =30		<i>n</i> =50		<i>n</i> =500	
	R	<i>p</i> -value	R	<i>p</i> -value	R	<i>p</i> -value
$\hat{\lambda}_1$	0,992	<0,01	0,996	0,032	0,999	0,048
$\hat{\lambda}_2$	0,969	<0,01	0,992	<0,01	0,999	0,047
$\hat{\lambda}_{12}$	0,780	<0,01	0,882	<0,01	0,994	0,021
\hat{s}_1	0,861	<0,01	0,829	<0,01	0,984	<0,01
\hat{s}_2	0,840	<0,01	0,845	<0,01	0,983	<0,01

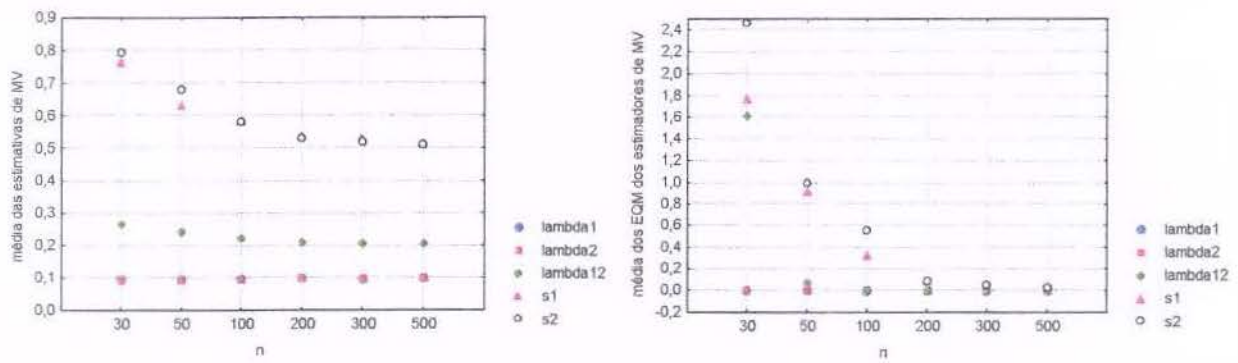


Figura 2.2 – Gráficos das médias das estimativas de MV e dos erros quadráticos médios dos estimadores dos parâmetros da EBVE com $\lambda_1 = \lambda_2 = 0,1$, $\lambda_{12} = 0,2$, $s_1 = s_2 = 0,5$, considerando dados sem censura.

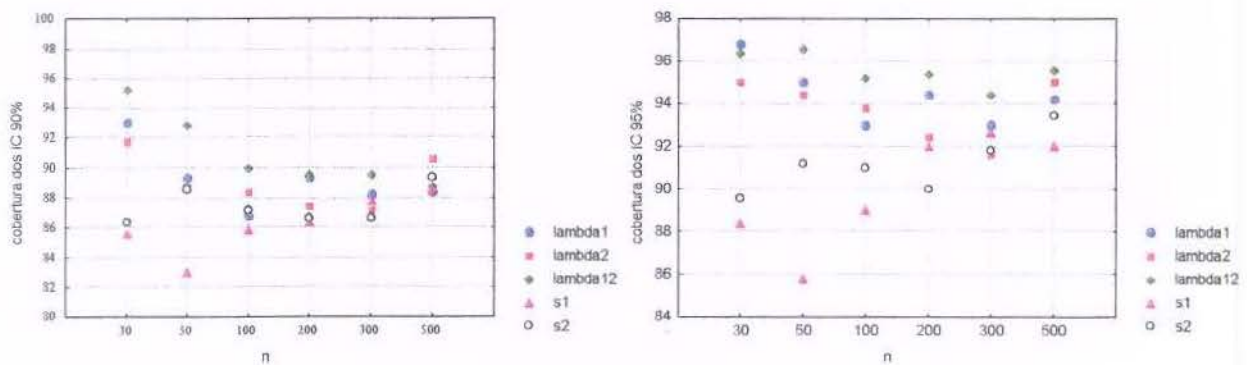


Figura 2.3 – Gráficos das coberturas dos intervalos de confiança aproximados, com coeficientes 90% e 95%, usando dados sem censura.

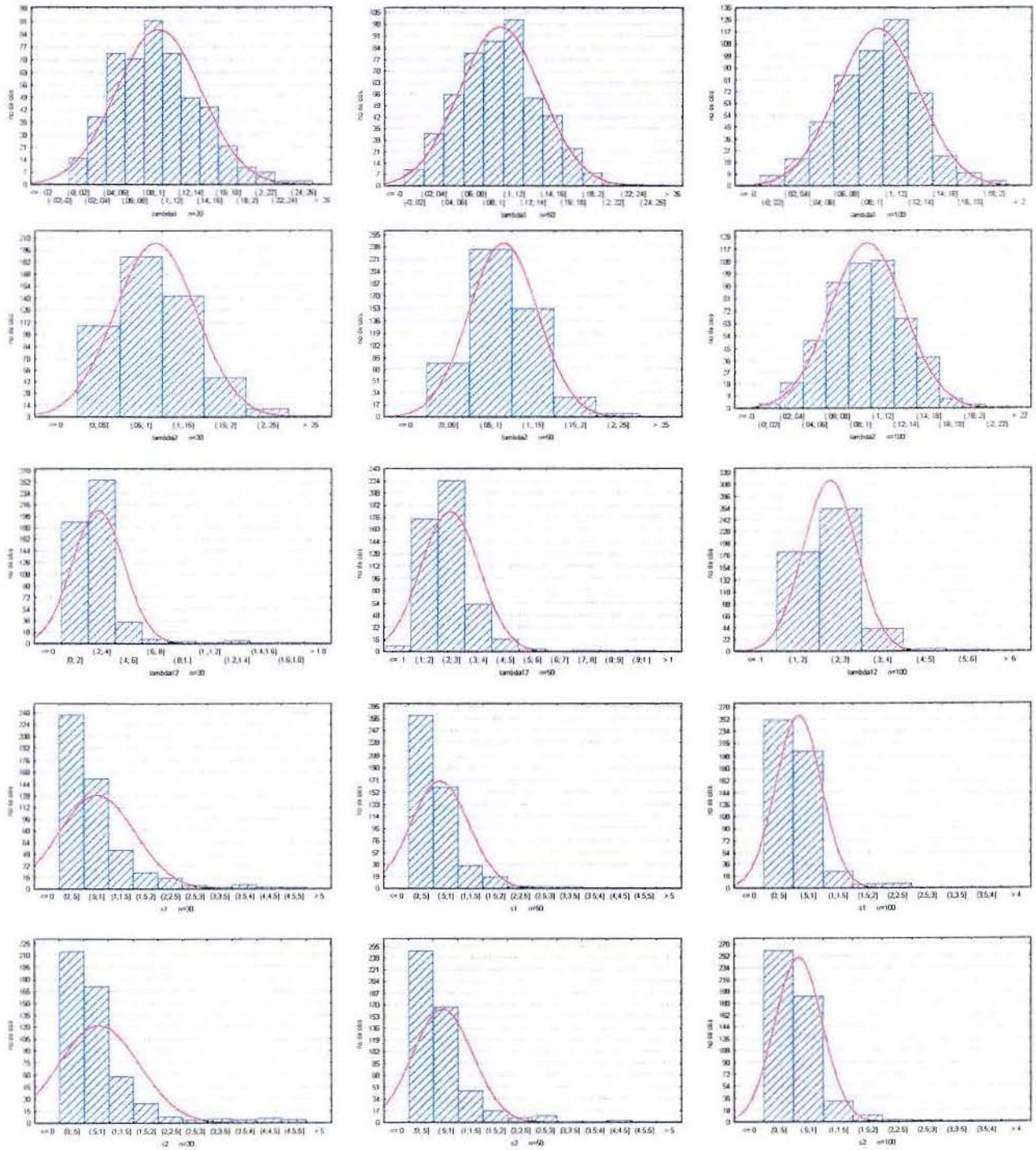


Figura 2.4 Histogramas das estimativas de MV ($n=30, 50$ e 100).

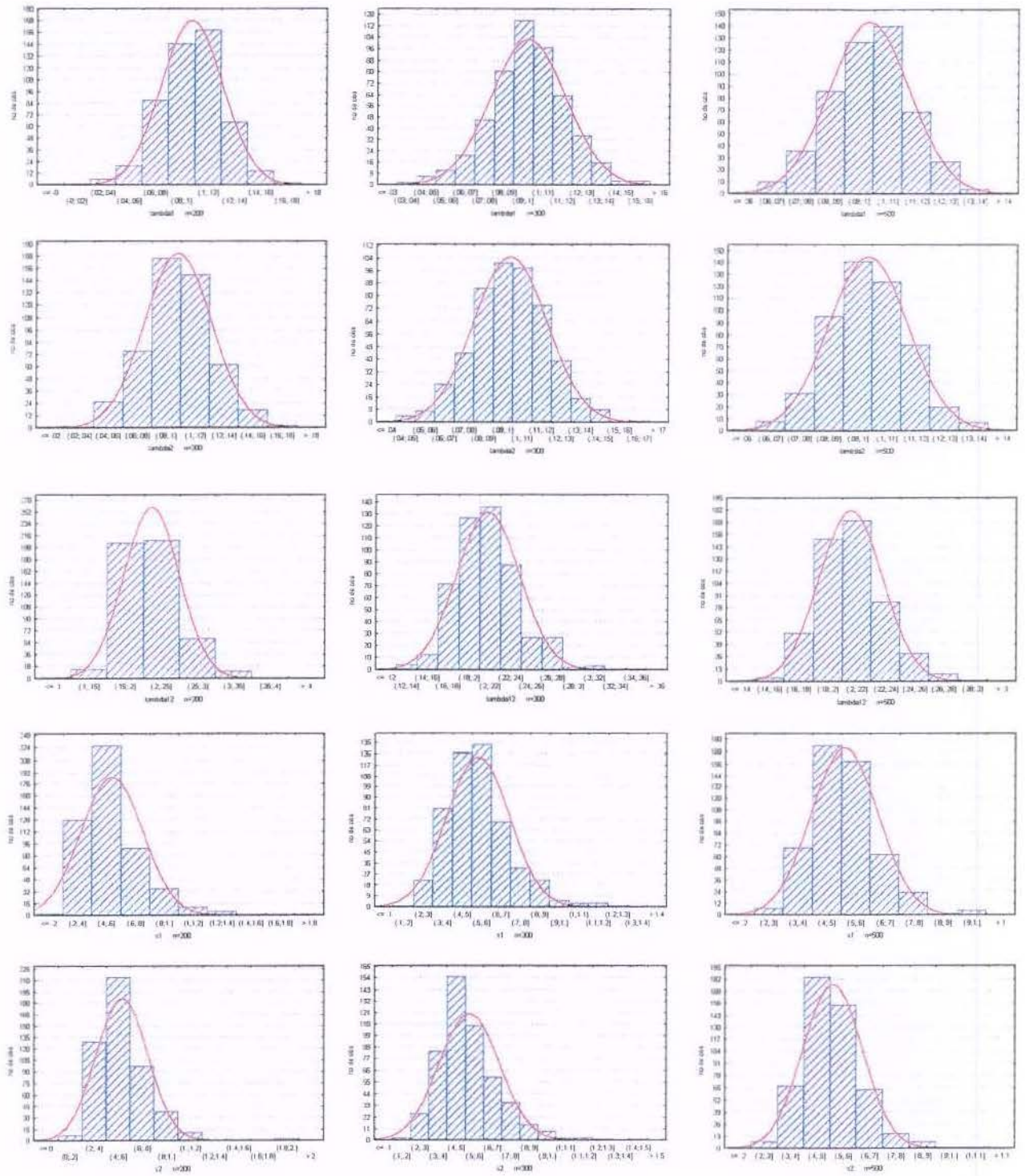


Figura 2.5 Histogramas das estimativas de MV ($n=200, 300$ e 500).

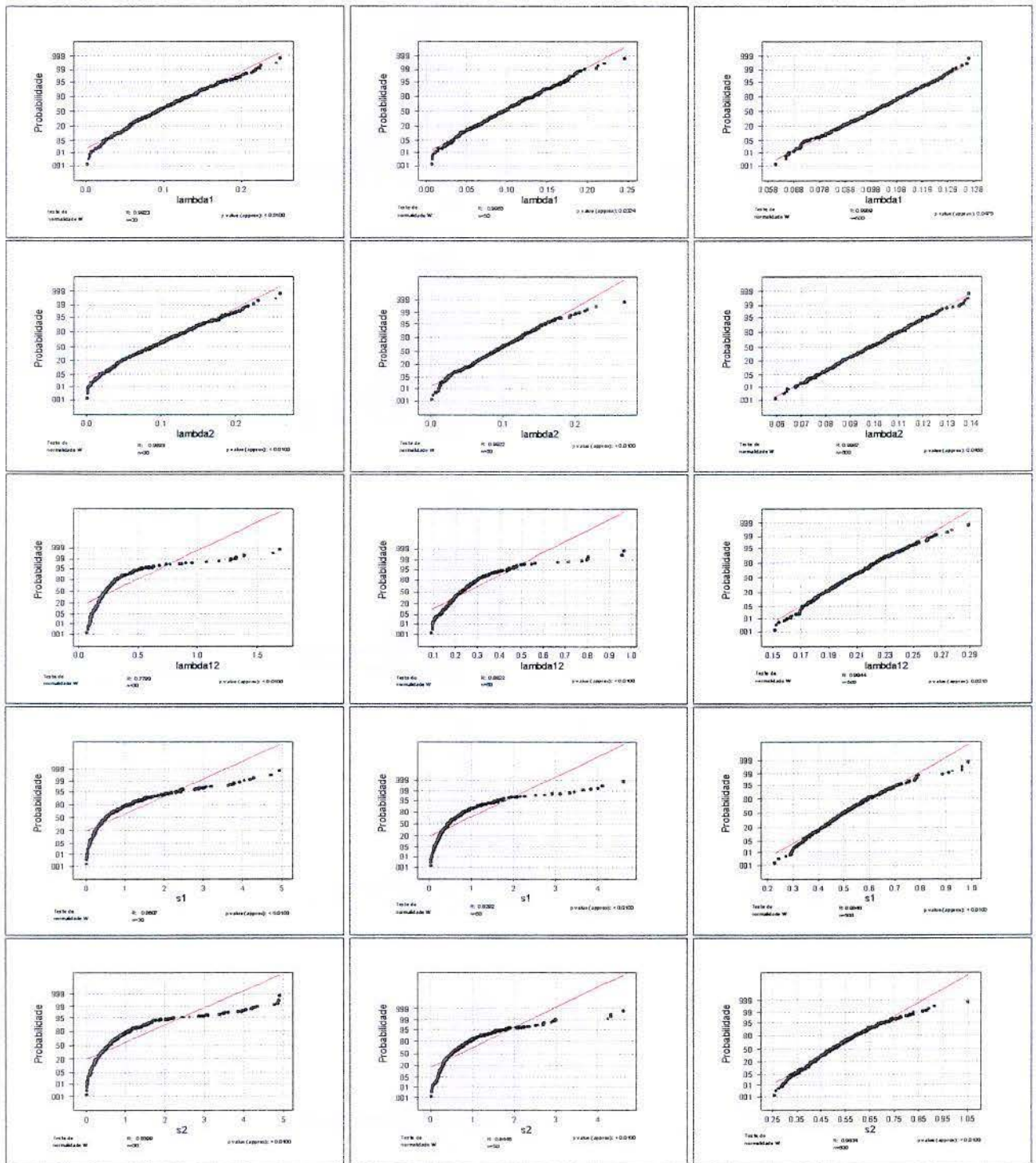


Figura 2.6 Gráficos normais probabilísticos das estimativas de MV ($n=30, 50, 500$).

2.3 Análise de Dados Bivariados Censurados

Nesta seção obtemos as estimativas de máxima verossimilhança dos parâmetros da EBVE considerando dados censurados e fazemos um estudo de simulação fixando alguns tempos de censuras.

2.3.1 Estimação dos Parâmetros usando o Método de MV: simulações

Para obter dados da distribuição EBVE considerando censura do tipo I, geramos amostras completas, utilizando o método da rejeição, apresentado na seção 2.1.1 e igualamos a τ , fixado previamente, todos os tempos excedendo esse valor. Foram usados os seguintes valores para os parâmetros, $\lambda_1 = \lambda_2 = 0,1$, $\lambda_{12} = 0,2$ e $s_1 = s_2 = 0,5$.

O estudo de simulação foi feito com 500 amostras de tamanhos 30, 50, 100 e 500, utilizando $\tau = 7, 8, 10$ e 12, correspondendo as probabilidades de censura 0,29, 0,22, 0,13 e 0,07, respectivamente.

As estimativas de MV dos parâmetros da EBVE foram obtidas numericamente, maximizando o log da verossimilhança em (1.29), onde f e S são dadas, respectivamente em (1.13), (1.12) e

$$\begin{aligned} \frac{-\partial}{\partial t_{1i}} S(t_{1i}, t_{2i}) &= \exp\left(-\lambda_1 t_1 - (\lambda_2 + \lambda_{12}) t_2 + \frac{\lambda_{12}}{s_2} (1 - e^{-s_2(t_2-t_1)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2})\right) \\ &\quad \times \left(\lambda_1 + \lambda_{12} e^{-s_2(t_2-t_1)} - \frac{\lambda_{12}}{s_1 + s_2} (s_2 e^{-s_2(t_2-t_1)} + s_1 e^{-s_1 t_1 - s_2 t_2})\right) \\ \frac{-\partial}{\partial t_{2i}} S(t_{1i}, t_{2i}) &= \exp\left(-(\lambda_1 + \lambda_{12}) t_1 - \lambda_2 t_2 + \frac{\lambda_{12}}{s_1} (1 - e^{-s_1(t_1-t_2)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_1(t_1-t_2)} - e^{-s_1 t_1 - s_2 t_2})\right) \\ &\quad \times \left(\lambda_2 + \lambda_{12} e^{-s_1(t_1-t_2)} - \frac{\lambda_{12}}{s_1 + s_2} (s_1 e^{-s_1(t_1-t_2)} + s_2 e^{-s_1 t_1 - s_2 t_2})\right). \end{aligned}$$

As amostras cujas estimativas não convergiram ou não pertenceram aos intervalos : $\hat{\lambda}_1 < 1$, $\hat{\lambda}_2 < 1$, $\hat{\lambda}_{12} < 2$, $\hat{s}_1 < 5$ e $\hat{s}_2 < 5$ foram descartadas. O total de amostras nessas condições, para os tamanhos de amostras e tempos de censuras considerados são apresentados na Tabela 2.11. O programa computacional implementado está incluído no Apêndice D. O tempo gasto

nas simulações utilizando um processador *Pentium* II-350MHz com 128 MB de RAM são apresentados na Tabela 2.12. As médias das estimativas de MV obtidas são apresentadas na Figura 2.7. Na Tabela 2.12 também estão as médias dos desvios padrões e dos erros quadráticos médios dos estimadores.

Com os resultados obtidos, observamos que: para n fixo, diminuindo as censuras as médias das estimativas vão convergindo para os valores dos parâmetros, mas muito lentamente; para t fixo aumentando o tamanho das amostras as estimativas dos parâmetros não melhoram, uma vez que o número de amostras censuras também aumenta. As estimativas de MV para dados com censuras parecem apresentar grandes vícios.

As coberturas dos intervalos de confiança assintóticos (Tabela 2.14) não estão próximas da cobertura nominal, uma vez que as variâncias dos estimadores parecem convergir para zero mais rapidamente do que os seus vícios.

Tabela 2.12 – Descrição das amostras simuladas considerando dados censurados

n	tempo censura	$\lambda_1 > 1$	$\lambda_2 > 1$	$\lambda_{12} > 2$	$s_1 > 5$	$s_2 > 5$	$\lambda_1, \lambda_2 > 1$ ou $\lambda_{12} > 2$ ou $s_1, s_2 > 5$	não conver- giram.	cov negativa
30	7	0*	0	90	26	27	129	842	0
	8	0	0	56	31	29	104	579	0
	10	0	0	26	22	21	63	443	0
	12	0	0	11	21	21	50	416	2
50	7	0	0	52	7	6	64	328	2
	8	0	0	21	4	7	30	200	1
	10	0	0	7	7	7	20	137	0
100	12	0	0	5	5	7	17	135	0
	7	0	0	35	2	2	37	78	0
	8	0	0	9	1	2	11	40	0
	10	0	0	2	0	0	2	28	0
500	12	0	0	1	0	1	2	28	1
	7	0	0	2	0	0	2	0	0
	8	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0

* Número de amostras

Tabela 2.13 - Médias das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da distribuição EBVE com $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$, $s_1=s_2=0,5$, considerando dados censurados

n	τ	$\hat{\lambda}_1$	dp λ_1	$\hat{\lambda}_2$	dp λ_2	$\hat{\lambda}_{12}$	dp λ_{12}	\hat{s}_1	dp s_1	\hat{s}_2	dp s_2	tempo
30	7	0,070	(0,058) ^a (0,041) ^b	0,07	(0,056) (0,043)	0,532	(0,539) (0,353)	0,686	(0,585) (0,805)	0,672	(0,544) (0,833)	2:45'58''
	8	0,075	(0,058) (0,042)	0,075	(0,057) (0,045)	0,499	(0,569) (0,336)	0,604	(0,514) (0,743)	0,575	(0,477) (0,666)	1:37'55''
	10	0,084	(0,058) (0,045)	0,085	(0,058) (0,046)	0,375	(0,383) (0,239)	0,633	(0,558) (0,694)	0,622	(0,550) (0,690)	1:12'25''
	12	0,088	(0,058) (0,046)	0,087	(0,057) (0,047)	0,312	(0,218) (0,193)	0,691	(0,590) (0,703)	0,692	(0,635) (0,721)	1:28'51''
50	7	0,067	(0,045) (0,035)	0,063	(0,044) (0,035)	0,594	(0,497) (0,350)	0,463	(0,343) (0,593)	0,451	(0,327) (0,573)	2:37'45
	8	0,073	(0,045) (0,036)	0,070	(0,044) (0,036)	0,492	(0,352) (0,277)	0,441	(0,311) (0,534)	0,445	(0,312) (0,496)	1:34'47''
	10	0,084	(0,046) (0,040)	0,080	(0,046) (0,039)	0,365	(0,219) (0,183)	0,488	(0,355) (0,521)	0,511	(0,367) (0,520)	1:03'00''
	12	0,091	(0,046) (0,041)	0,085	(0,047) (0,040)	0,296	(0,147) (0,126)	0,558	(0,410) (0,560)	0,582	(0,432) (0,542)	59'23''
100	7	0,062	(0,031) (0,028)	0,063	(0,031) (0,028)	0,626	(0,390) (0,311)	0,314	(0,166) (0,391)	0,293	(0,153) (0,256)	2:31'05''
	8	0,070	(0,032) (0,031)	0,072	(0,032) (0,031)	0,490	(0,259) (0,222)	0,344	(0,184) (0,356)	0,328	(0,171) (0,289)	1:36'19''
	10	0,082	(0,034) (0,033)	0,083	(0,033) (0,032)	0,341	(0,120) (0,120)	0,424	(0,222) (0,395)	0,409	(0,213) (0,327)	1:20'11''
	12	0,088	(0,034) (0,034)	0,089	(0,033) (0,033)	0,278	(0,079) (0,080)	0,494	(0,267) (0,408)	0,490	(0,259) (0,407)	1:16'34''
500	7	0,063	(0,014) (0,014)	0,063	(0,014) (0,013)	0,581	(0,119) (0,164)	0,227	(0,054) (0,075)	0,228	(0,054) (0,075)	5:26'50''
	8	0,071	(0,014) (0,014)	0,071	(0,014) (0,014)	0,437	(0,065) (0,076)	0,269	(0,058) (0,074)	0,270	(0,058) (0,075)	5:13'44''
	10	0,084	(0,015) (0,015)	0,084	(0,015) (0,015)	0,310	(0,038) (0,036)	0,348	(0,076) (0,092)	0,346	(0,075) (0,088)	4:53'00''
	12	0,091	(0,015) (0,015)	0,091	(0,015) (0,015)	0,255	(0,029) (0,027)	0,414	(0,093) (0,106)	0,412	(0,092) (0,102)	4:44'14''

^a Média das estimativas dos desvios padrões dos estimadores

^b Desvio padrão das estimativas

Tabela 2.14 Cobertura dos intervalos de confiança aproximados, com coeficientes 90% e 95%, usando dados censurados

n	tempo censura	λ_1	λ_2	λ_{12}	s_1	s_2	
30	7	86,2 ^a	86,2	97,2	75,0	75,4	
		92,0 ^b	92,6	99,8	82,0	80,0	
	8	88,8	88,8	98,2	69,8	71,0	
		93,8	92,2	99,8	75,0	77,6	
	10	90,8	91,8	99,4	75,8	76,2	
		94,6	95,0	99,6	79,8	81,2	
	12	92,2	90,8	99,0	80,0	81,0	
		95,8	94,2	99,2	82,2	85,6	
	50	7	78,8	76,6	86,0	58,8	59,2
			86,6	83,6	97,4	65,6	65,8
8		86,4	82,4	90,6	59,6	60,4	
		91,2	89,0	98,6	63,8	67,8	
10		89,0	85,8	91,8	64,8	69,8	
		92,6	91,8	99,2	70,4	74,4	
12		88,8	88,6	94,6	74,4	79,4	
		93,6	93,6	99,0	79,4	83,2	
100		7	63,0	65,6	46,0	38,6	38,2
			72,4	73,0	68,0	44,2	44,8
	8	71,6	72,0	49,2	43,2	44,0	
		80,4	80,8	73,8	50,4	50,4	
	10	82,0	83,4	67,6	59,2	59,8	
		88,0	90,4	88,2	67,2	64,8	
	12	85,6	86,6	83,2	73,2	72,2	
		92,4	92,8	93,6	78,6	78,6	
	500	7	16,6	15,6	1,40	2,80	3,40
			25,4	24,0	2,00	4,60	4,80
8		37,0	35,2	0,00	8,00	7,40	
		48,6	45,8	0,40	10,4	11,4	
10		68,6	70,0	2,20	34,8	37,0	
		76,8	78,6	6,60	43,4	43,2	
12		81,4	83,4	38,2	65,4	62,8	
		89,8	91,2	55,6	73,6	70,8	

^{a,b} Intervalos de 90% e 95% de confiança respectivamente.

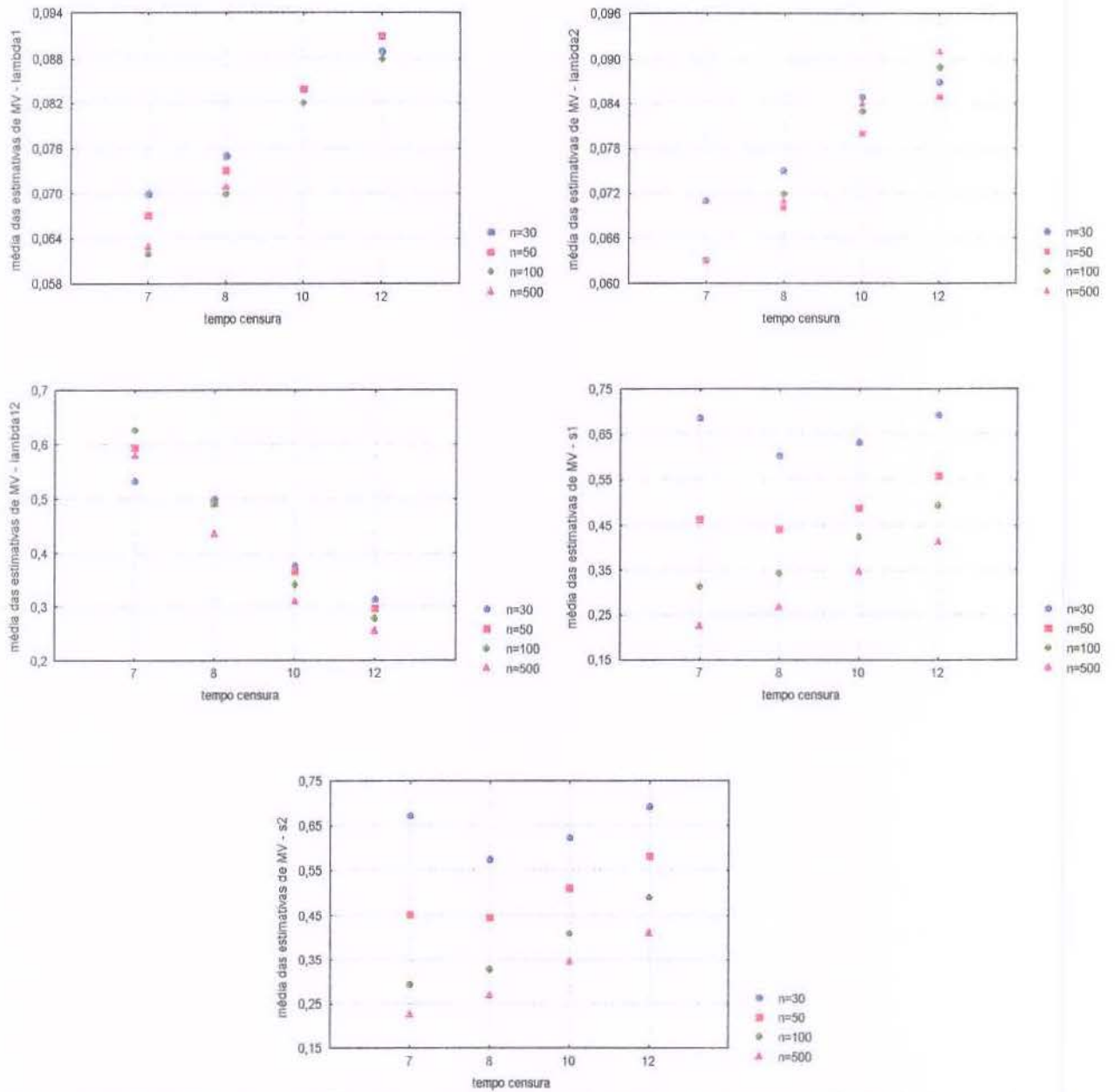


Figura 2.7 - Gráficos das médias das estimativas de MV dos parâmetros da EBVE para dados censurados

CAPÍTULO III

ANÁLISE BAYESIANA DA DISTRIBUIÇÃO EBVE

Neste capítulo encontramos as densidades marginais *a posteriori* dos parâmetros da EBVE assumindo densidades *a priori* informativas e usando os algoritmos de Gibbs com Metropolis Hastings (M-H). Apresentamos um exemplo de aplicação e fazemos simulações para estudar o modelo sob a abordagem Bayes-empírica.

3.1 Análise Bayesiana utilizando os métodos de Gibbs e Metropolis Hastings

Consideremos uma amostra aleatória, $\left(\underset{\sim}{T}_1, \underset{\sim}{T}_2\right) = (T_{11}, T_{21}), \dots, (T_{1n}, T_{2n})$, dos tempos até as falhas de n pares de componentes, com função de densidade EBVE, dada em (1.13).

Assumimos densidades marginais *a priori* gama $\Gamma(a_i, b_i)$, $i=1, 2, \dots, 5$, independentes para os parâmetros, uma vez que essa distribuição é definida para valores positivos sendo bastante flexível através das escolhas de a_i e b_i . Dessa forma, a densidade *a priori* conjunta é dada por:

$$\begin{aligned} \pi(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2) &= \pi_1(\lambda_1) \pi_2(\lambda_2) \pi_3(\lambda_{12}) \pi_4(s_1) \pi_5(s_2) \\ &\propto \lambda_1^{b_1-1} \lambda_2^{b_2-1} \lambda_{12}^{b_3-1} s_1^{b_4-1} s_2^{b_5-1} \exp\left(-\left(a_1\lambda_1 + a_2\lambda_2 + a_3\lambda_{12} + a_4s_1 + a_5s_2\right)\right) \end{aligned}$$

Os parâmetros das densidades *a priori* devem ser escolhidos de acordo com a experiência do pesquisador. Neste trabalho, os valores de (a_i, b_i) , $i=1, 2, \dots, 5$, foram obtidos usando informações contidas nos dados. Assim sendo, estudamos os parâmetros da distribuição EBVE sob a abordagem Bayes-empírica.

A densidade *a posteriori* conjunta de $(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2)$ é dada por:

$$\pi\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \propto \pi(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \quad (3.1)$$

onde $L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right)$ é a função de verossimilhança.

Para obter as densidades marginais *a posteriori* dos parâmetros, utilizamos o amostrador de Gibbs associado com o algoritmo M-H, apresentados na seção 1.4.2. As densidades condicionais usadas são dadas por:

$$\begin{aligned} \pi\left(\lambda_1 \mid \lambda_2, \lambda_{12}, s_1, s_2, (t_1, t_2)\right) &\propto \lambda_1^{b_1-1} \exp(-a_1 \lambda_1) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \\ \pi\left(\lambda_2 \mid \lambda_1, \lambda_{12}, s_1, s_2, (t_1, t_2)\right) &\propto \lambda_2^{b_2-1} \exp(-a_2 \lambda_2) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \\ \pi\left(\lambda_{12} \mid \lambda_1, \lambda_2, s_1, s_2, (t_1, t_2)\right) &\propto \lambda_{12}^{b_3-1} \exp(-a_3 \lambda_{12}) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \\ \pi\left(s_1 \mid \lambda_1, \lambda_2, \lambda_{12}, s_2, (t_1, t_2)\right) &\propto s_1^{b_4-1} \exp(-a_4 s_1) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \\ \pi\left(s_2 \mid \lambda_1, \lambda_2, \lambda_{12}, s_1, (t_1, t_2)\right) &\propto s_2^{b_5-1} \exp(-a_5 s_2) L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right) \end{aligned} \quad (3.2)$$

onde $L\left(\lambda_1, \lambda_2, \lambda_{12}, s_1, s_2 \mid (t_1, t_2)\right)$ é a função de verossimilhança.

3.1.1 Exemplo de Aplicação

Nesta seção encontramos as densidades marginais *a posteriori* dos parâmetros da EBVE, usando uma amostra de tamanho $n=50$, gerada pelo método da rejeição, apresentado na seção 2.1.1.

Utilizamos o seguinte critério na escolha dos parâmetros das densidades *a priori*: igualamos a estimativa de MV à média da distribuição gama, dada por $\frac{a_i}{b_i}$ e a estimativa da variância do estimador de MV à variância da gama, dada por $\frac{a_i}{b_i^2}$ e resolvemos o sistema

formado por essas duas equações. Os valores de (a_i, b_i) obtidos e os gráficos das densidades marginais *a priori* estão apresentados na Tabela 3.1 e Figura 3.1.

As densidades marginais *a posteriori* foram obtidas gerando observações das distribuições condicionais completas, dadas em (3.2), através do algoritmo M-H, com núcleo de transição dado pelas densidades marginais *a priori*. Para cada variável (parâmetro), foram geradas 5 cadeias com 6000 iterações inicializadas em pontos arbitrários e desprezadas as 1000 primeiras iterações (*burn-in*). Das iterações restantes, foram selecionadas uma a cada dez, totalizando 500 iterações para cada cadeia.

Os gráficos das trajetórias e das autocorrelações dentro das cadeias são apresentados nas Figuras 3.2 e 3.3, respectivamente, e ambos deram indicações de convergência, uma vez que as cadeias apresentaram repetidamente o mesmo comportamento e que a autocorrelação dentro das cadeias parece ser pequena. Aplicamos o critério de Gelman e Rubin, apresentado na seção (1.4.3) e os resultados obtidos, Tabela 3.2, foram bastante próximos de 1. Assim sendo, selecionamos a segunda metade das iterações de cada cadeia para constituir as amostras das densidades marginais *a posteriori*. Foram feitos os histogramas dessas amostras e para cada uma delas, ajustamos uma distribuição gama, Figura 3.4. Um resumo das densidades marginais *a posteriori* são apresentados na Tabela 3.3.

Tabela 3.1 - Parâmetros das densidades *a priori* usados no exemplo de aplicação

i	a_i	b_i
1	8,45	73,7
2	3,59	48,5
3	11,5	55,5
4	4,20	13,2
5	2,40	4,07

Tabela 3.2 - Índices de convergência obtidos pelo critério de Gelman e Rubin

	λ_1	λ_2	λ_{12}	s_1	s_2
$\sqrt{\hat{R}}$	1,000	0,999	1,000	1,001	1,001

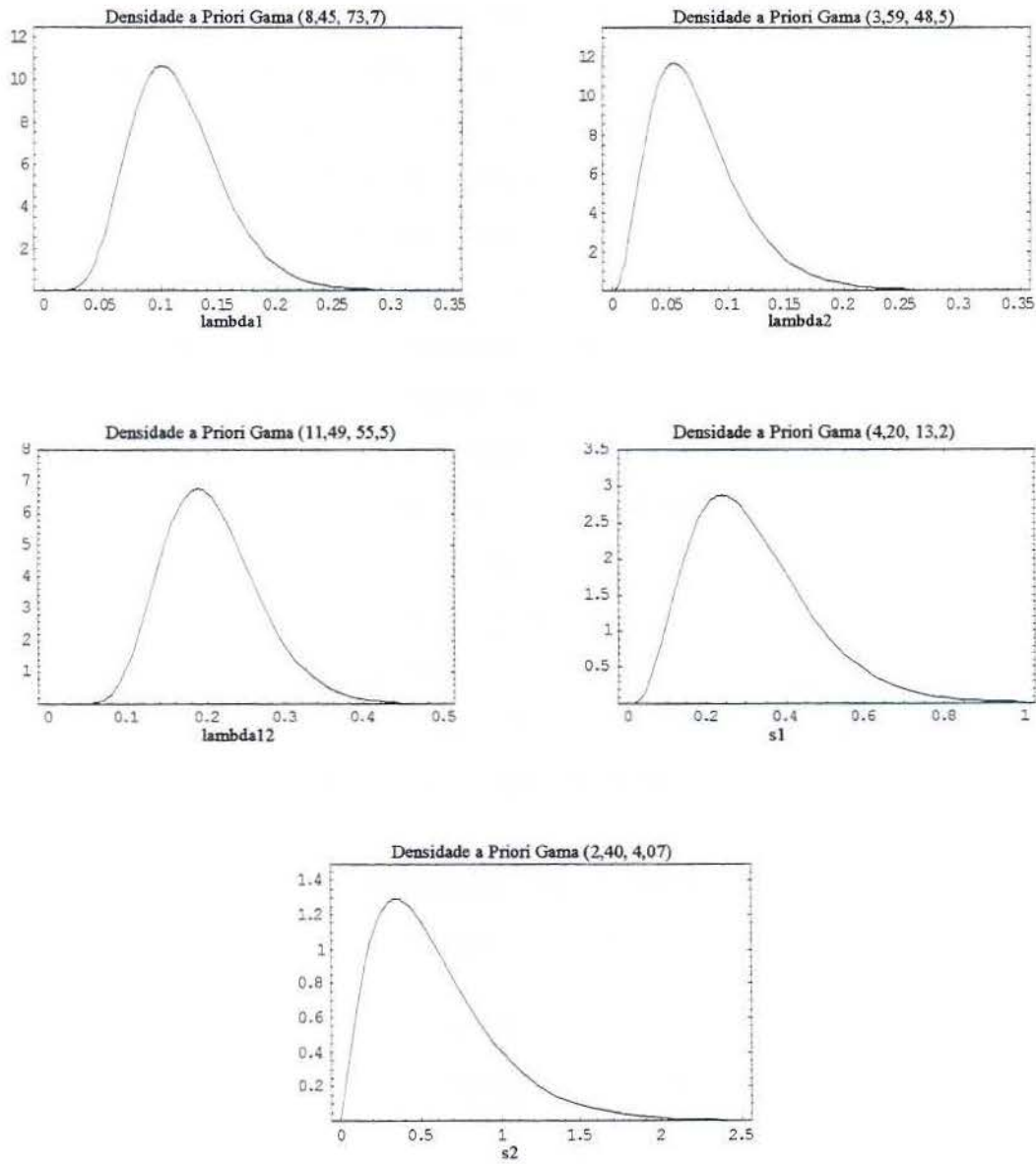


Figura 3.1 - Gráficos das densidades marginais *a priori* dos parâmetros assumidas no exemplo de aplicação

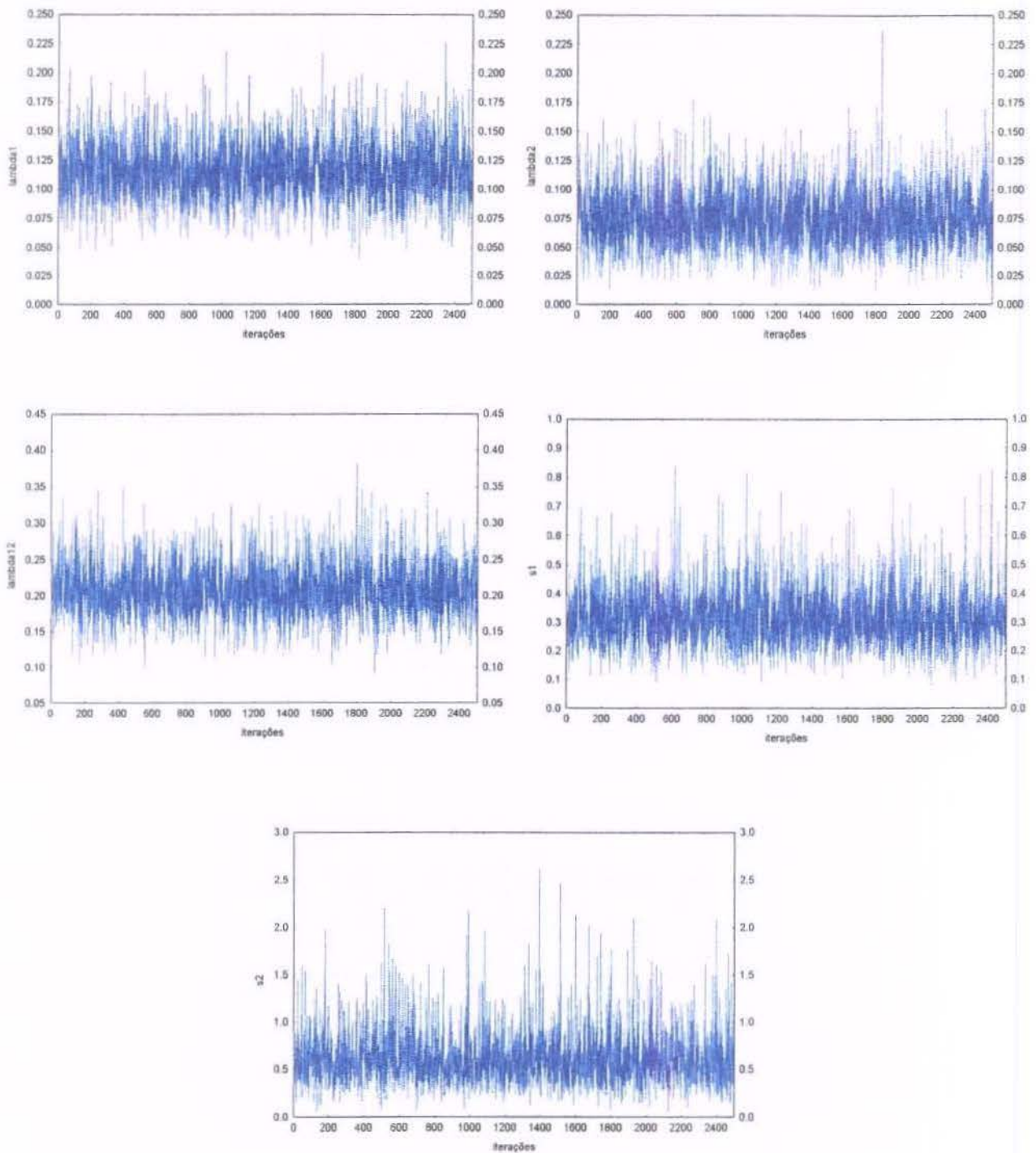


Figura 3.2 - Gráficos das trajetórias das cadeias, correspondentes ao exemplo de aplicação

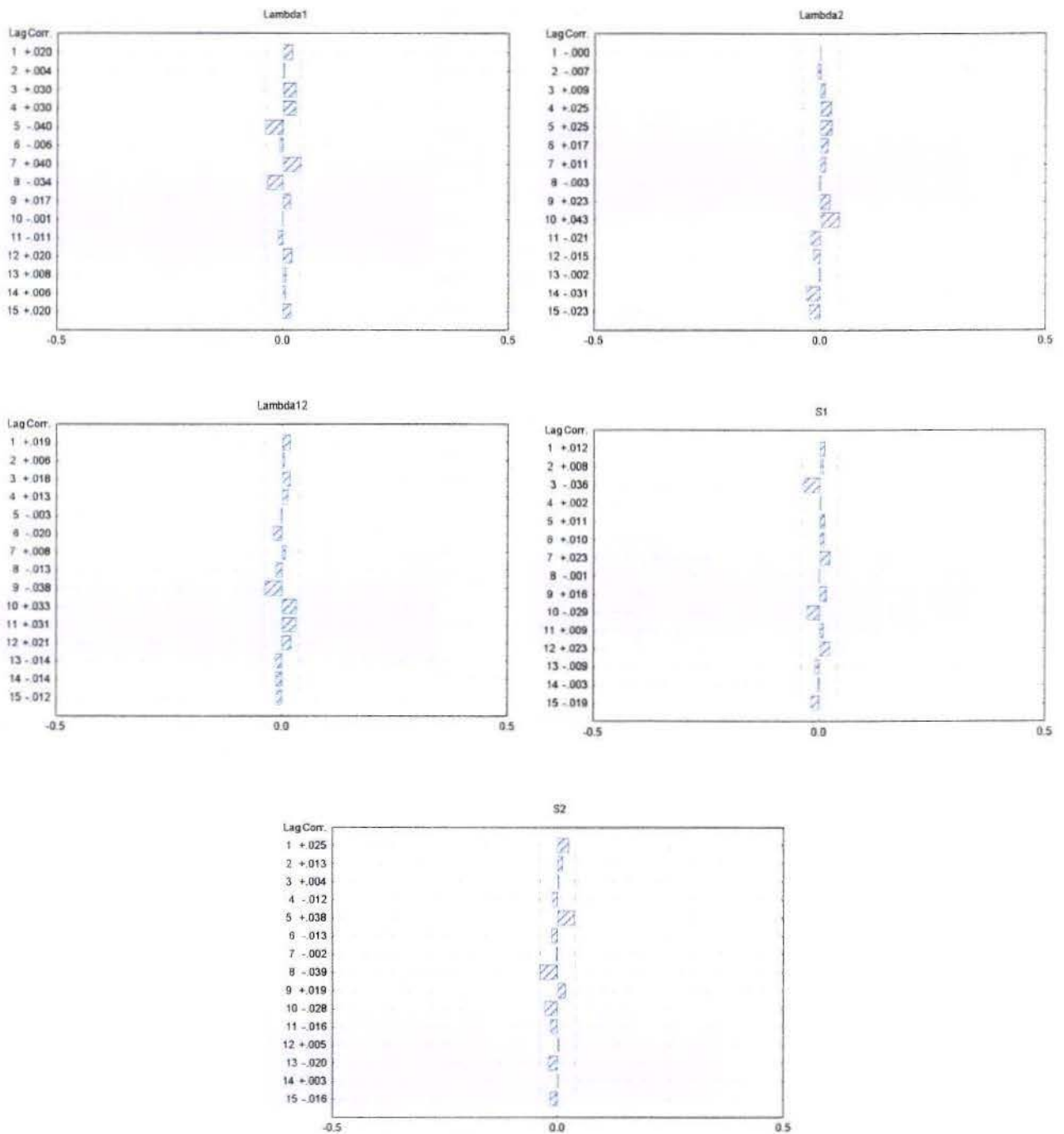


Figura 3.3 - Gráficos das correlações dentro das cadeias, no exemplo de aplicação

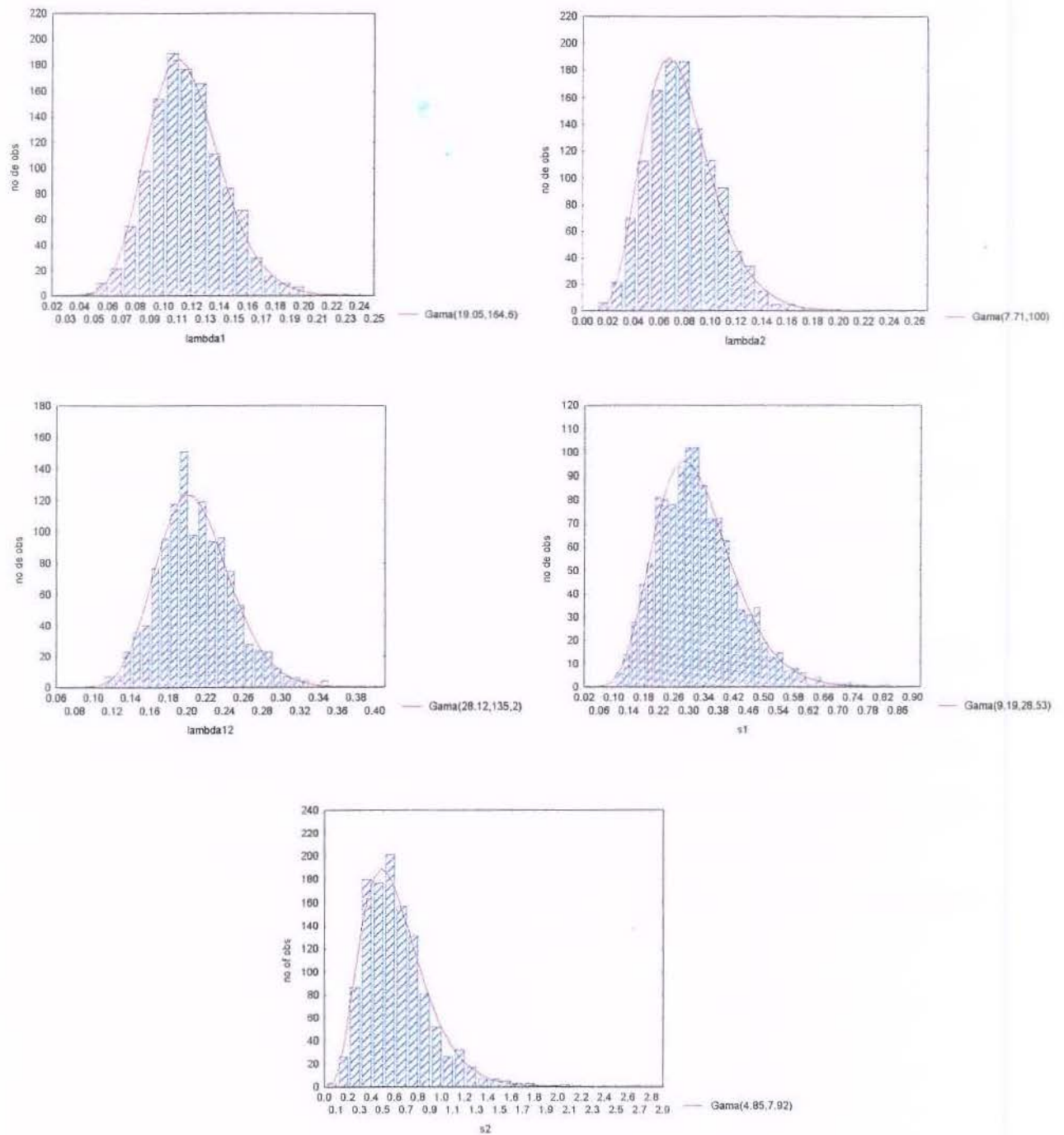


Figura 3.4 - Gráficos das densidades marginais *a posteriori* dos parâmetros obtidas no exemplo de aplicação

Tabela 3.3 - Resumo das densidades marginais *a posteriori* correspondentes ao exemplo de aplicação

	λ_1	λ_2	λ_{12}	s_1	s_2
média	0,12	0,08	0,21	0,32	0,61
mediana	0,11	0,07	0,21	0,31	0,56
mínimo	0,041	0,01	0,09	0,10	0,08
máximo	0,23	0,24	0,38	0,83	2,6
variância	0,00069	0,00073	0,0015	0,012	0,085
d. padrão	0,026	0,027	0,039	0,107	0,292
IC 90%	(0,076;0,16)	(0,037;0,13)	(0,15;0,28)	(0,17;0,52)	(0,25;1,17)
IC 95%	(0,069;0,17)	(0,031;0,13)	(0,14;0,29)	(0,15;0,56)	(0,20;1,34)

3.1.2 Estudo dos Parâmetros da Distribuição EBVE sob a Abordagem Bayes-Empírica

Nesta seção apresentamos dois estudos de simulação dos parâmetros da distribuição EBVE, feitos com 500 amostras de tamanho 50. As amostras foram geradas pelo método da rejeição (2.1.1) e para cada uma delas encontramos as densidades marginais *a posteriori*, assumindo densidades *a priori* $\Gamma(a_{ij}, b_{ij})$, $i=1, 2, \dots, 5$, para os parâmetros, onde a_{ij} e b_{ij} foram obtidos das informações contidas nos dados.

No primeiro estudo (Estudo de Simulação 3.1) os a_{ij} e b_{ij} , $i=1, 2, \dots, 5$ foram escolhidos a partir das relações:

$$\begin{aligned}
 a_{1j} &= \frac{\hat{\lambda}_{1j}^2}{\hat{\text{var}}(\lambda_{1j})} & b_{1j} &= \frac{\hat{\lambda}_{1j}}{\hat{\text{var}}(\lambda_{1j})} \\
 a_{2j} &= \frac{\hat{\lambda}_{2j}^2}{\hat{\text{var}}(\lambda_{2j})} & b_{2j} &= \frac{\hat{\lambda}_{2j}}{\hat{\text{var}}(\lambda_{2j})} \\
 a_{3j} &= \frac{\hat{\lambda}_{12j}^2}{\hat{\text{var}}(\lambda_{12j})} & b_{3j} &= \frac{\hat{\lambda}_{12j}}{\hat{\text{var}}(\lambda_{12j})} \\
 a_{4j} &= \frac{\hat{s}_{1j}^2}{\hat{\text{var}}(s_{1j})} & b_{4j} &= \frac{\hat{s}_{1j}}{\hat{\text{var}}(s_{1j})} \\
 a_{5j} &= \frac{\hat{s}_{2j}^2}{\hat{\text{var}}(s_{2j})} & b_{5j} &= \frac{\hat{s}_{2j}}{\hat{\text{var}}(s_{2j})}
 \end{aligned} \tag{3.3}$$

onde $\hat{\lambda}_{1j}$, $\hat{\lambda}_{2j}$, $\hat{\lambda}_{12j}$, \hat{s}_{1j} , \hat{s}_{2j} , são as estimativas de MV e $\hat{v}\hat{a}r(\lambda_{1j})$, $\hat{v}\hat{a}r(\lambda_{2j})$, $\hat{v}\hat{a}r(\lambda_{12j})$, $\hat{v}\hat{a}r(s_{1j})$, $\hat{v}\hat{a}r(s_{2j})$ são as estimativas das variâncias dos estimadores de MV da amostra j .

Para obter as densidades marginais *a posteriori* utilizamos o método de simulação iterativa de Gibbs (1.4.2) e para amostrar das distribuições condicionais completas desconhecidas (3.2), usamos o algoritmo M-H, com núcleo de transição dado pelas densidades *a priori*. Para cada parâmetro (variável) foram geradas 5 cadeias com 6000 iterações, inicializadas em pontos arbitrários e desprezadas as 1000 primeiras iterações (*burn-in*). Das iterações restantes, foram selecionadas uma a cada dez, totalizando 500 iterações para cada cadeia. Para verificação da convergência utilizamos o critério de Gelman e Rubin e foram incluídas no estudo somente as amostras com $0,995 \leq \sqrt{\hat{R}} \leq 1,005$ para todos os parâmetros, onde $\sqrt{\hat{R}}$ é dada em (1.47). Das amostras selecionadas tomamos as últimas 240 iterações de cada cadeia, totalizando 1200 iterações, para constituir as amostras das densidades marginais *a posteriori*.

Para cada uma das 500 amostras que convergiram obtivemos:

1. As estimativas de MV pelo método numérico BFGS e os intervalos aproximados com 90% e 95% de confiança.
2. Um resumo das densidades marginais *a posteriori* incluindo média, mediana, desvio padrão e intervalos Bayesianos.
3. Os erros quadráticos médios das estimativas de MV e das médias e medianas das densidades *a posteriori*.

Na Tabela 3.4 são apresentadas as médias e os desvios padrões dos resultados obtidos, referente aos itens 1, 2 e 3 acima, e na Tabela 3.5 estão as coberturas dos intervalos de confiança e Bayesianos. O tempo gasto nas simulações foi 296 horas e 47', utilizando um processador *Pentium II* – 350 MHz com 128 MB de RAM. O programa computacional implementado em Ox (Doornik, 1999) encontra-se no Apêndice E.

Com os resultados mostrados na Tabela 3.4 observamos que as médias e medianas das densidades *a posteriori* de λ_1 , λ_2 e λ_{12} parecem estar bem próximas dos valores dos parâmetros, enquanto que as de s_1 e s_2 não estão tão próximas e apresentam uma maior variabilidade. Ainda observamos que os erros quadráticos médios das estimativas de λ_1 , λ_2 e λ_{12} estão próximos de zero, enquanto que as estimativas de s_1 e s_2 parecem ser menos

eficientes. De acordo com a Tabela 3.5, a porcentagem de intervalos de confiança que contém o verdadeiro valor do parâmetro é maior do que a dos intervalos Bayesianos.

Tabela 3.4 - Resumo das estimativas dos parâmetros da EBVE obtidas no Estudo de Simulação 3.1

média×10 (dp×10)	λ_1	λ_2	λ_{12}	s_1	s_2
estimativas	0,91 ^a (0,39)	0,91(0,38)	2,30 (0,72)	6,42 (4,94)	6,83 (5,68)
	0,92 ^b (0,41)	0,91 (0,40)	2,36 (0,75)	6,84 (5,71)	7,25 (6,32)
	0,90 ^c (0,42)	0,89 (0,42)	2,32 (0,73)	6,20 (4,91)	6,57 (5,40)
dp	0,42 ^a (0,10)	0,43 (0,11)	0,73 (0,35)	4,26 (4,33)	4,57 (5,03)
	0,27 ^{b,c} (0,06)	0,28 (0,06)	0,46 (0,21)	3,22 (3,80)	3,46 (4,50)
eqm	0,04 ^a (0,02)	0,04 (0,02)	0,13 (0,22)	6,32 (21,85)	8,17 (26,96)
	0,03 ^b (0,02)	0,03 (0,02)	0,10 (0,18)	6,06 (23,93)	7,72 (28,20)
	0,03 ^c (0,03)	0,03 (0,02)	0,09 (0,16)	5,02 (18,64)	6,37 (22,42)

^a Estimativas de MV

^{b,c} Médias e medianas das amostras das densidades *a posteriori*, respectivamente.

Tabela 3.5 Cobertura dos intervalos Bayesianos e de confiança aproximados, correspondentes ao Estudo de Simulação 3.1

	λ_1	λ_2	λ_{12}	s_1	s_2
Intervalos 90%	89,8 ^a	88,6	93,6	87,4	86,6
	71,0 ^b	71,6	63,8	69,6	68,4
Intervalos 95%	94,4 ^a	94,2	97,8	90,4	88,0
	81,0 ^b	81,0	74,2	76,6	77,2

^{a,b} Intervalos de confiança e Bayesianos, respectivamente.

Na Figura 3.5 apresentamos os gráficos dos intervalos de confiança e Bayesianos dos parâmetros, para as 50 primeiras amostras estudadas e na Figura 3.6 os percentis 2,5 e 97,5 das densidades marginais *a priori* e *posteriori* dos parâmetros.

Com os gráficos das Figuras 3.5, 3.6 observamos que:

- as densidades *a posteriori* parecem ser sensíveis a escolha das densidades *a priori* para o tamanho de amostra considerado ($n=50$).
- os intervalos Bayesianos são sempre menores do que os intervalos de confiança como era esperado, uma vez que os intervalos Bayesianos foram obtidos considerando uma informação anterior.

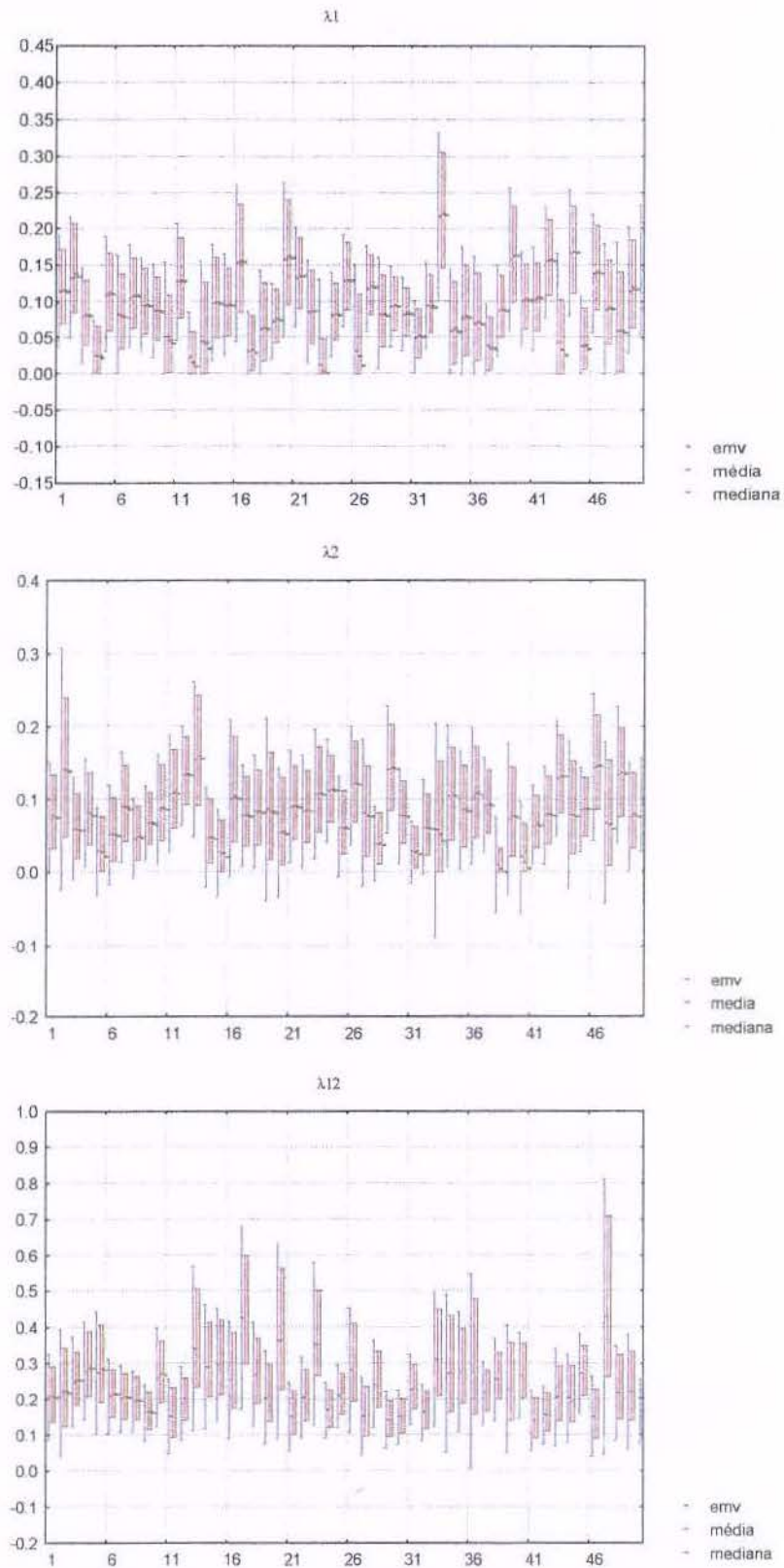


Figura 3.5 - Gráficos das estimativas de MV, médias e medianas das densidades marginais *a posteriori* dos parâmetros, juntamente com os intervalos de confiança e Bayesianos

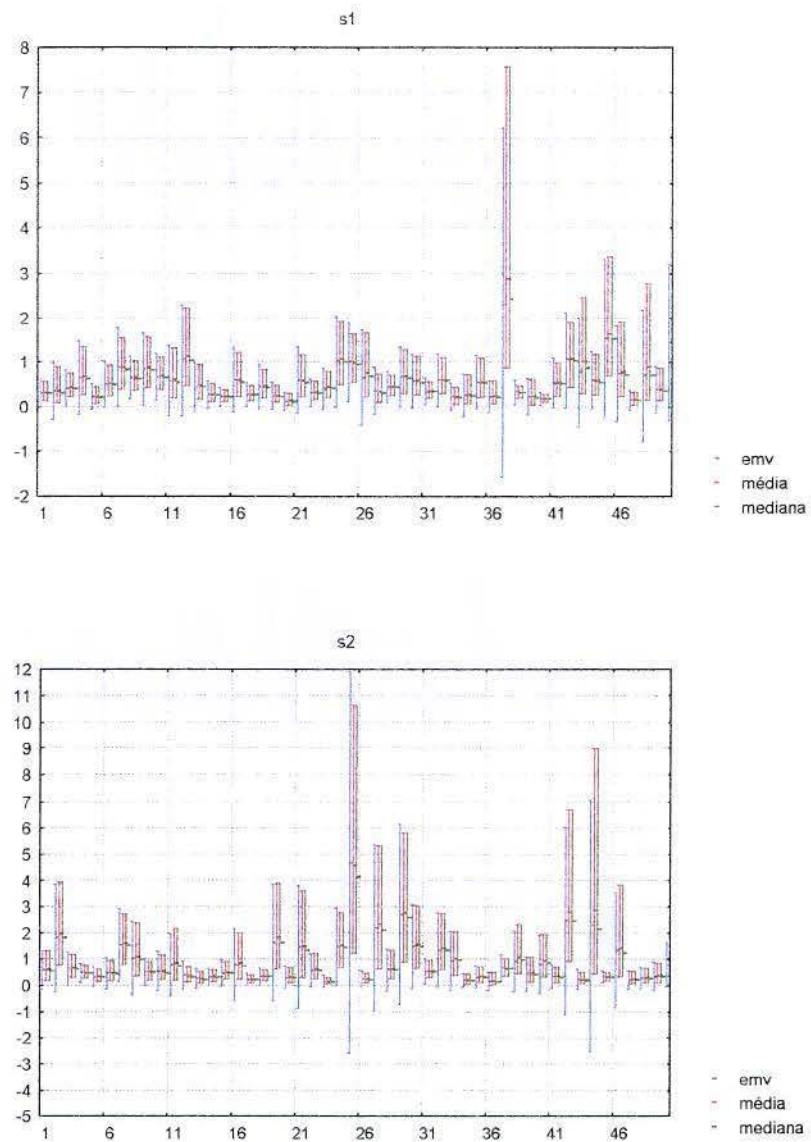


Figura 3.5 (cont)

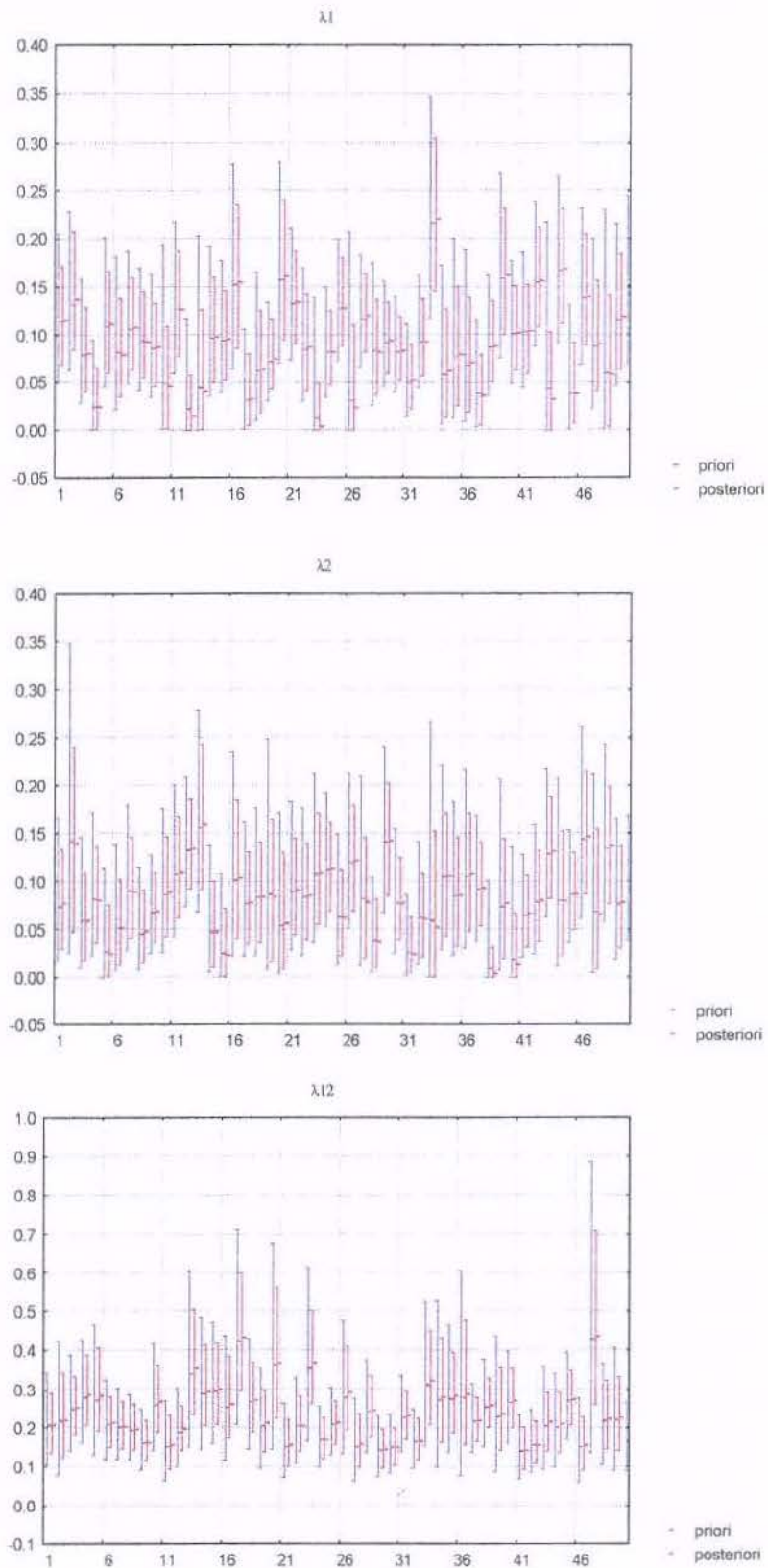


Figura 3.6 - Gráficos das médias e dos percentis 2,5 e 97,5 das densidades marginais *a priori* e *posteriori* dos parâmetros

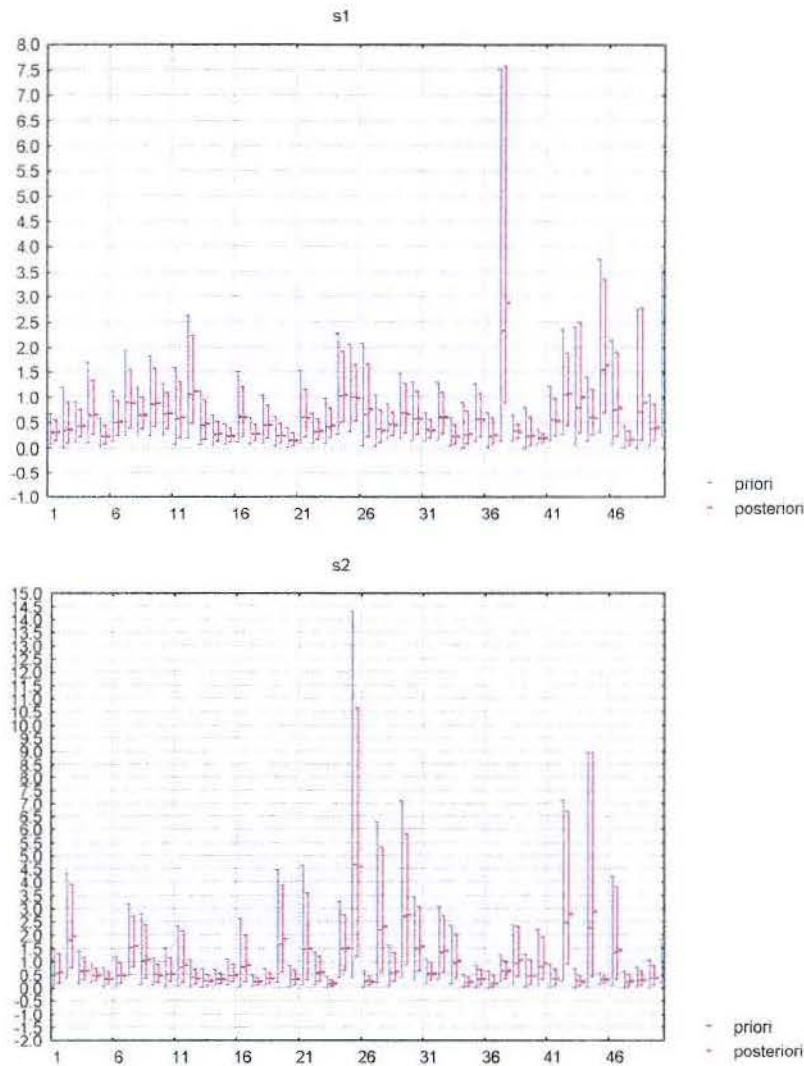


Figura 3.6 (cont)

No segundo estudo de simulação (Estudo de Simulação 3.2), os parâmetros das distribuições gama foram obtidos igualando as médias e variâncias das densidades *a priori* às médias das médias e dos desvios padrões das 500 estimativas de MV, para amostras de tamanho 500 (Tabela 2.6). Os parâmetros das densidades *a priori* são apresentados na Tabela 3.6 e os resultados das simulações nas Tabelas 3.7 e 3.8. O tempo gasto nas simulações foi 353 horas, utilizando um processador *Pentium II* – 350 MHz com 128 MB de RAM.

Tabela 3.6 Parâmetros das densidades *a priori* usados no Estudo de Simulação 3.2

i	a_{ij}	b_{ij}
1	5	50
2	5	50
3	4	20
4	1	2
5	1	2

Tabela 3.7 Resumo das estimativas dos parâmetros da EBVE obtidas no Estudo de Simulação 3.2

média×10 (dp×10)	λ_1	λ_2	λ_{12}	s_1	s_2
estimativas	0,95 ^a (0,39)	0,93 (0,39)	2,19 (0,77)	7,12 (5,46)	7,64 (6,64)
	1,01 ^b (0,19)	1,00 (0,18)	2,09 (0,34)	6,08 (1,89)	6,25 (2,08)
	0,99 ^c (0,19)	0,98 (0,18)	2,05 (0,33)	5,41 (1,81)	5,57 (1,99)
dp	0,43 ^a (0,13)	0,43 (0,16)	0,74 (0,61)	4,96 (5,61)	5,57 (7,05)
	0,30 ^{b,c} (0,04)	0,30 (0,04)	0,50 (0,10)	3,25 (1,00)	3,35 (1,04)
eqm	0,04 ^a (0,03)	0,04 (0,04)	0,15 (0,61)	9,03 (33,11)	13,15(50,69)
	0,01 ^b (0,01)	0,01 (0,01)	0,04 (0,02)	1,63 (1,39)	1,82 (1,73)
	0,01 ^c (0,01)	0,01 (0,01)	0,04 (0,02)	1,50 (1,13)	1,65 (1,42)

^a Estimativas de MV.

^{b,c} Médias e medianas das amostras das distribuições *a posteriori*.

Tabela 3.8 Cobertura dos intervalos Bayesianos e de confiança aproximados, correspondentes ao Estudo de Simulação 3.2

	λ_1	λ_2	λ_{12}	s_1	s_2
intervalos 90%	90,6 ^a	89,4	91,0	90,8	90,8
	98,4 ^b	99,2	98,2	99,0	98,4
intervalos 95%	94,4 ^a	95,6	94,4	92,2	92,2
	99,4 ^b	100	99,6	99,8	99,4

^{a,b} Intervalos de confiança e Bayesianos, respectivamente.

A Tabela 3.7 mostra que os erros quadráticos médios das médias e medianas das densidades *a posteriori* parecem ser bem menores do que os das estimativas de MV, especialmente para os parâmetros s_1 e s_2 . Observamos também que a cobertura dos intervalos Bayesianos, Tabela 3.8 está próxima de 100%, mesmo utilizando os percentis 5 e 95 das

densidades *a posteriori*, possivelmente devido a informação contida nas densidades *a priori* assumidas.

Portanto, os resultados obtidos indicam uma possível sensibilidade com relação as densidades *a priori*.

CAPÍTULO IV

A DISTRIBUIÇÃO EBVE PARA TEMPOS ACELERADOS

Neste capítulo formulamos um modelo para tempos acelerados, ou seja, incluímos na função de sobrevivência bivariada uma covariável de aceleração, assumindo uma relação de potência inversa e fazemos algumas simulações considerando amostras de vários tamanhos para estudar os estimadores de MV.

4.1 Formulação do Modelo EBVE para Tempos Acelerados

Representemos por (T_1, T_2) os tempos até as falhas de dois componentes pareados, com $T_i = \min(X_i, Z_i)$, onde X_i e Z_i são os tempos até as falhas das partes específica e comum respectivamente, do componente i , $i=1,2$.

Assumimos uma relação de potência inversa entre a voltagem e o tempo, de modo que o tempo até a falha acelerada da parte específica do componente i , $i=1,2$ é dado por:

$$X_i | v_j = \exp\left(\varphi\left(v_j, \beta\right)\right) X_i = \exp(\ln\beta_{0i} + \beta_{1i}(-\ln v_j)) X_i = \beta_{0i} v_j^{-\beta_{1i}} X_i \quad (4.1)$$

e das partes que apresentam o mesmo risco de falha por:

$$Z_i | v_j = \exp\left(\varphi\left(v_j, \beta\right)\right) Z_i = \exp(\ln\beta_{03} + \beta_{13}(-\ln v_j)) Z_i = \beta_{03} v_j^{-\beta_{13}} Z_i, \quad (4.2)$$

onde β_{0i} e β_{1i} , $i=1, 2, 3$, correspondem respectivamente aos parâmetros A e P em (1.21) e v_j é a voltagem no nível $j=1,2,\dots,k$.

Dessa forma, os tempos até as falhas aceleradas dos dois componentes é dado por:

$$(T_1|v_j, T_2|v_j) = (\min(X_1|v_j, Z_1|v_j), \min(X_2|v_j, Z_2|v_j)). \quad (4.3)$$

A função de confiabilidade do tempo até a falha acelerada da parte específica do componente i é obtida a seguir:

$$\begin{aligned} S_{X_i|v_j}(t) &= S_{X_i} \left(\exp \left(-\varphi \left(v_j, \beta \right) \right) t \right) \\ &= S_{X_i} \left(\exp \left(-\ln \beta_{0i} - \beta_{1i} (-\ln v_j) \right) t \right) \\ &= S_{X_i} \left(e^{-\ln \beta_{0i}} v_j^{\beta_{1i}} t \right) \\ &= S_{X_i} \left(\beta_{0i}^{-1} v_j^{\beta_{1i}} t \right) \\ &= \exp \left(\lambda_i \beta_{0i}^{-1} v_j^{\beta_{1i}} t \right), \end{aligned}$$

Assim sendo,

$$S_{X_i|v_j}(t) = \exp \left(\beta_{0i}^* v_j^{\beta_{1i}} t \right) \quad i = 1, 2 \quad j = 1, 2, \dots, k, \quad (4.4)$$

onde $\beta_{0i}^* = \lambda_i \beta_{0i}^{-1}$, ou seja, $X_i | v_j \sim \text{Exp} \left(\beta_{0i}^* v_j^{\beta_{1i}} \right)$ e portanto a função de risco de $X_i | v_j$ é constante, dada por:

$$h_{X_i|v_j}(t) = \beta_{0i}^* v_j^{\beta_{1i}}. \quad i = 1, 2 \quad j = 1, 2, \dots, k. \quad (4.5)$$

A função de confiabilidade de $Z_i | v_j$ é obtida da seguinte forma:

$$\begin{aligned} S_{Z_i|v_j}(t) &= S_{Z_i} \left(\exp \left(-\varphi \left(v_j, \beta \right) \right) t \right) \\ &= S_{Z_i} \left(\exp \left(-\ln \beta_{03} - \beta_{13} (-\ln v_j) \right) t \right) \\ &= S_{Z_i} \left(e^{-\ln \beta_{03}} v_j^{\beta_{13}} t \right) \\ &= S_{Z_i} \left(\beta_{03}^{-1} v_j^{\beta_{13}} t \right) \\ &\stackrel{(1.7)}{=} \exp \left(-\lambda_{12} \beta_{03}^{-1} v_j^{\beta_{13}} t + \frac{\lambda_{12} \beta_{03}^{-1}}{s_i \beta_{03}^{-1}} \left(1 - e^{-s_i \beta_{03}^{-1} v_j^{\beta_{13}} t} \right) \right), \end{aligned}$$

portanto,

$$S_{Z_i|v_j}(t) = \exp\left(-\beta_{03}^* v_j^{\beta_{13}} t + \frac{\beta_{03}^*}{\beta_{0(3+i)}^*} \left(1 - e^{-\beta_{0(3+i)}^* v_j^{\beta_{13}} t}\right)\right) \quad i = 1, 2 \quad j = 1, 2, \dots, k \quad (4.6)$$

onde $\beta_{03}^* = \lambda_{12} \beta_{03}^{-1}$ e $\beta_{0(3+i)}^* = s_i \beta_{03}^{-1}$, $i=1, 2$. Logo, a função de risco de $Z_i|v_j$ é dada por:

$$h_{Z_i|v_j}(t) = \beta_{03}^* v_j^{\beta_{13}} \left(1 - e^{-\beta_{0(3+i)}^* v_j^{\beta_{13}} t}\right) \quad i = 1, 2 \quad j = 1, 2, \dots, k. \quad (4.7)$$

Na derivação da EBVE para (T_1, T_2) , Ryu assumiu que X_i e Z_i são independentes, ou seja,

$$P(X_i < x_i, Z_i < z_i) = P(X_i < x_i)P(Z_i < z_i) \quad (4.8)$$

e $X_i|v_j$ e $Z_i|v_j$ são independentes desde que,

$$P(X_i | v_j < x_i, Z_i | v_j < z_i) = P(X_i | v_j < x_i)P(Z_i | v_j < z_i),$$

ou seja,

$$\begin{aligned} & P\left(\exp\left(\varphi\left(v_j, \beta\right)\right) X_i < x_i, \exp\left(\varphi\left(v_j, \beta\right)\right) Z_i < z_i\right) = \\ \Leftrightarrow & P\left(\exp\left(\varphi\left(v_j, \beta\right)\right) X_i < x_i\right) P\left(\exp\left(\varphi\left(v_j, \beta\right)\right) Z_i < z_i\right) \\ \Leftrightarrow & P\left(X_i < x_i \exp\left(-\varphi\left(v_j, \beta\right)\right), Z_i < z_i \exp\left(-\varphi\left(v_j, \beta\right)\right)\right) = \\ \Leftrightarrow & P\left(X_i < x_i \exp\left(-\varphi\left(v_j, \beta\right)\right)\right) P\left(Z_i < z_i \exp\left(-\varphi\left(v_j, \beta\right)\right)\right) \end{aligned}$$

mas de (4.8) temos que esta última igualdade é válida e portanto, $X_i|v_j$ e $Z_i|v_j$ são independentes. Dessa forma, a função de risco de $T_i|v_j$ é dada pela soma das funções de riscos $X_i|v_j$ e $Z_i|v_j$, ou seja,

$$h_{T_i|v_j}(t) = \beta_{0i}^* v_j^{\beta_{1i}} + \beta_{03}^* v_j^{\beta_{13}} \left(1 - e^{-\beta_{0(3+i)}^* v_j^{\beta_{13}} t}\right) \quad i = 1, 2 \quad j = 1, 2, \dots, k. \quad (4.9)$$

Da última equação obtemos a função de confiabilidade marginal de $T_i|v_j$, dada por:

$$S_{T_i|v_j}(t) = \exp\left(-\beta_{0i}^* v_j^{\beta_{1i}} t - \beta_{03}^* v_j^{\beta_{13}} t + \frac{\beta_{03}^*}{\beta_{0(3+i)}^*} \left(1 - e^{-\beta_{0(3+i)}^* v_j^{\beta_{13}} t}\right)\right) \quad i = 1, 2 \quad j = 1, 2, \dots, k. \quad (4.10)$$

Portanto, a função densidade marginal de $T_i|v_j$ é dada por:

$$f_{T_i|v_j}(t) = S_{T_i|v_j}(t) \left(\beta_{0i}^* v_j^{\beta_{1i}} + \beta_{0(3+i)}^* v_j^{\beta_{13}} \left(1 - e^{-\beta_{0(3+i)}^* v_j^{\beta_{13}} t} \right) \right) \quad i = 1, 2 \quad j = 1, 2, \dots, k. \quad (4.11)$$

A função de confiabilidade conjunta de $(T_1|v_j, T_2|v_j)$ pode ser escrita como:

$$\begin{aligned} S_{T_1|v_j, T_2|v_j}(t_1, t_2) &\stackrel{(a_2^3)}{=} E(S_{(T_1|v_j, T_2|v_j)N_{12}}(t_1, t_2)) \\ &= E(P(T_1 | v_j > t_1, T_2 | v_j > t_2) | N_{12}) \\ &= E(P(\min(X_1 | v_j, Z_1 | v_j) > t_1, \min(X_2 | v_j, Z_2 | v_j) > t_2) | N_{12}) \\ &= E(P(X_1 | v_j > t_1, Z_1 | v_j > t_1, X_2 | v_j > t_2, Z_2 | v_j > t_2) | N_{12}) \\ &= E(P(\beta_{01} v_j^{-\beta_{11}} X_1 > t_1, \beta_{03} v_j^{-\beta_{13}} Z_1 > t_1, \beta_{02} v_j^{-\beta_{12}} X_2 > t_2, \beta_{03} v_j^{-\beta_{13}} Z_2 > t_2) | N_{12}) \\ &\stackrel{ind}{=} E(P(X_1 > \beta_{01}^{-1} v_j^{\beta_{11}} t_1) P(X_2 > \beta_{02}^{-1} v_j^{\beta_{12}} t_2) P(Z_1 > \beta_{03}^{-1} v_j^{\beta_{13}} t_1, Z_2 > \beta_{03}^{-1} v_j^{\beta_{13}} t_2) | N_{12}) \end{aligned}$$

e de forma análoga ao apêndice A₂ obtemos,

$$S_{T_1|v_j, T_2|v_j}(t_1, t_2) = \begin{cases} \exp\left(-\beta_{01}^* v_j^{\beta_{11}} t_1 - \beta_{02}^* v_j^{\beta_{12}} t_2 - \beta_{03}^* v_j^{\beta_{13}} t_1 + \frac{\beta_{03}^*}{\beta_{04}^*} \left(1 - e^{-\beta_{04}^* v_j^{\beta_{13}} (t_1 - t_2)}\right)\right) \\ \times \exp\left(\frac{\beta_{03}^*}{\beta_{04}^* + \beta_{05}^*} \left(e^{-\beta_{04}^* v_j^{\beta_{13}} (t_1 - t_2)} - e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2}\right)\right) & \text{se } t_1 > t_2 \\ \exp\left(-\beta_{01}^* v_j^{\beta_{11}} t_1 - \beta_{02}^* v_j^{\beta_{12}} t_2 - \beta_{03}^* v_j^{\beta_{13}} t_2 - \frac{\beta_{03}^*}{\beta_{04}^*} \left(1 - e^{-\beta_{05}^* v_j^{\beta_{13}} (t_2 - t_1)}\right)\right) \\ \times \exp\left(\frac{\beta_{03}^*}{\beta_{04}^* + \beta_{05}^*} \left(e^{-\beta_{05}^* v_j^{\beta_{13}} (t_2 - t_1)} - e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2}\right)\right) & \text{se } t_1 \leq t_2, \end{cases} \quad (4.12)$$

onde $\beta_{01}^* = \lambda_1 \beta_{01}^{-1}$, $\beta_{02}^* = \lambda_2 \beta_{02}^{-1}$, $\beta_{03}^* = \lambda_{12} \beta_{03}^{-1}$, $\beta_{04}^* = s_1 \beta_{03}^{-1}$ e $\beta_{05}^* = s_2 \beta_{03}^{-1}$.

A função densidade bivariada de $(T_1|v_j, T_2|v_j)$ é dada por:

$$f_{T_1|v_j, T_2|v_j}(t_1, t_2) = \begin{cases} S_{T_1|v_j, T_2|v_j}(t_1, t_2) \times \\ \left(\beta_{02}^* v_j^{\beta_{12}} + \beta_{03}^* v_j^{\beta_{13}} e^{-\beta_{04}^*(t_1-t_2)} - \frac{\beta_{03}^*}{\beta_{04}^* + \beta_{05}^*} \left(\beta_{04}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}}(t_1-t_2)} + \beta_{05}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) \times \\ \left(\beta_{01}^* v_j^{\beta_{11}} + \beta_{03}^* v_j^{\beta_{13}} - \beta_{03}^* v_j^{\beta_{13}} e^{-\beta_{04}^*(t_1-t_2)} + \frac{\beta_{03}^* \beta_{04}^* v_j^{\beta_{13}}}{\beta_{04}^* + \beta_{05}^*} \left(e^{-\beta_{04}^* v_j^{\beta_{13}}(t_1-t_2)} - e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) + \\ S_{T_1|v_j, T_2|v_j}(t_1, t_2) \times \\ \left(\beta_{04}^* \beta_{03}^* v_j^{2\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}}(t_1-t_2)} - \frac{\beta_{03}^* \beta_{04}^* v_j^{\beta_{13}}}{\beta_{04}^* + \beta_{05}^*} \left(\beta_{04}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}}(t_1-t_2)} + \beta_{05}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) \\ \text{set } t_1 > t_2 \\ S_{T_1|v_j, T_2|v_j}(t_1, t_2) \times \\ \left(\beta_{01}^* v_j^{\beta_{11}} + \beta_{03}^* v_j^{\beta_{13}} e^{-\beta_{05}^*(t_2-t_1)} - \frac{\beta_{03}^*}{\beta_{04}^* + \beta_{05}^*} \left(\beta_{05}^* v_j^{\beta_{13}} e^{-\beta_{05}^* v_j^{\beta_{13}}(t_2-t_1)} - \beta_{04}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) \times \\ \left(\beta_{02}^* v_j^{\beta_{12}} + \beta_{03}^* v_j^{\beta_{13}} - \beta_{03}^* v_j^{\beta_{13}} e^{-\beta_{05}^*(t_2-t_1)} + \frac{\beta_{05}^* \beta_{03}^* v_j^{\beta_{13}}}{\beta_{04}^* + \beta_{05}^*} \left(e^{-\beta_{05}^* v_j^{\beta_{13}}(t_2-t_1)} - e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) + \\ S_{T_1|v_j, T_2|v_j}(t_1, t_2) \times \\ \left(\beta_{03}^* \beta_{05}^* v_j^{2\beta_{13}} e^{-\beta_{05}^* v_j^{\beta_{13}}(t_2-t_1)} - \frac{\beta_{03}^* \beta_{05}^* v_j^{\beta_{13}}}{\beta_{04}^* + \beta_{05}^*} \left(\beta_{05}^* v_j^{\beta_{13}} e^{-\beta_{05}^* v_j^{\beta_{13}}(t_2-t_1)} + \beta_{04}^* v_j^{\beta_{13}} e^{-\beta_{04}^* v_j^{\beta_{13}} t_1 - \beta_{05}^* v_j^{\beta_{13}} t_2} \right) \right) \\ \text{se } t_1 \leq t_2. \end{cases} \quad (4.13)$$

onde $\beta_{01}^* = \lambda_1 \beta_{01}^{-1}$, $\beta_{02}^* = \lambda_2 \beta_{02}^{-1}$, $\beta_{03}^* = \lambda_{12} \beta_{03}^{-1}$, $\beta_{04}^* = s_1 \beta_{03}^{-1}$ e $\beta_{05}^* = s_2 \beta_{03}^{-1}$.

Note que a função em (4.13) é a função de densidade da EBVE com parâmetros $\beta_{01}^* v_j^{\beta_{11}}$, $\beta_{02}^* v_j^{\beta_{12}}$, $\beta_{03}^* v_j^{\beta_{13}}$, $\beta_{04}^* v_j^{\beta_{13}}$ e $\beta_{05}^* v_j^{\beta_{13}}$. Portanto, da forma como a distribuição para tempos acelerados foi derivada é possível obter estimativas dos parâmetros nas condições usuais de operação. Neste trabalho, para ilustração, consideramos a voltagem usual, $v_0=1,5$.

Apresentamos a seguir, Figura 4.1, os gráficos das funções marginais, de sobrevivência, de densidade e de risco, da EBVE para tempos acelerados com parâmetros $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$ nas voltagens, $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$.

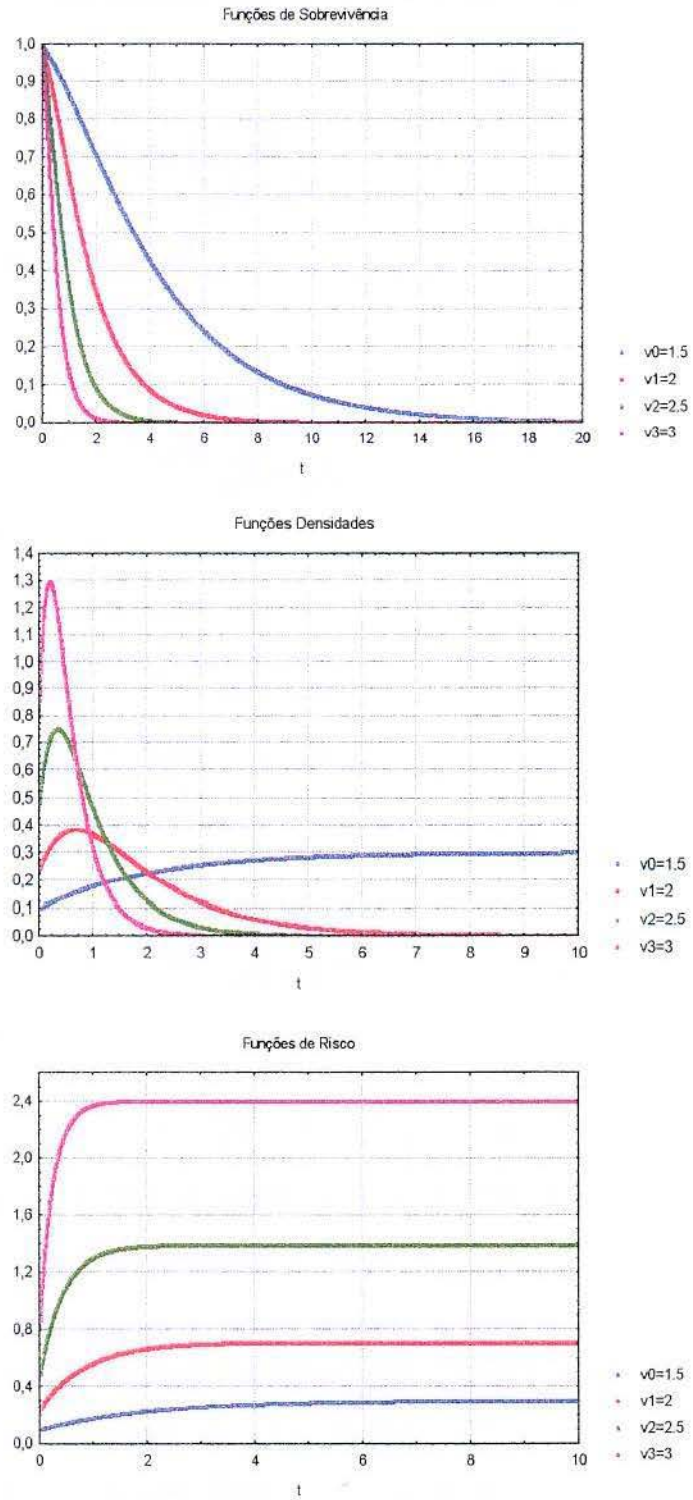


Figura 4.1 - Gráficos das funções marginais de sobrevivência, de densidade e de risco, nos níveis de voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$.

4.2 Estimação dos Parâmetros usando o Método de MV

Consideremos uma amostra de n pares de componentes submetidos, desde o início até o fim do experimento, a uma carga de voltagem constante, mais elevada do que a usual. Suponha que 3 níveis de voltagens v_j , $j=1, 2, 3$, tenham sido utilizados e em cada nível n_j pares de componentes sejam colocados em teste.

Assumindo independência entre os componentes testados em cada nível de voltagem, a função de verossimilhança dos parâmetros é obtida usando (1.30) onde $f_{T_1|v_1, T_2|v_2}(t_{1i}, t_{2i}; \theta)$ é dada em (4.13).

Apresentamos na seção 4.2.1 um exemplo de aplicação e na seção 4.2.2 fazemos algumas simulações para estudar as propriedades assintóticas dos estimadores de MV dos parâmetros da distribuição EBVE para tempos acelerados e dos parâmetros na voltagem usual.

4.2.1 Exemplo de Aplicação

Nesta seção analisamos uma amostra de tamanho $n=n_1+n_2+n_3$, com $n_i=170$, $i=1, 2, 3$, da distribuição EBVE para tempos acelerados, com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$. Esses valores foram escolhidos, de modo que os parâmetros nas condições usuais de operação, $v_0=1,5$, fossem $\lambda_1=\lambda_2=0,1$, $\lambda_{12}=0,2$, $s_1=s_2=0,5$.

A amostra foi gerada pelo método da rejeição, apresentado na seção 2.1.1. Os parâmetros das distribuições uniformes, necessárias para a geração dos dados, foram obtidos a partir dos gráficos das funções de densidades bivariadas, dadas em 4.13, para os níveis de voltagens, $v_1=2$, $v_2=2,5$ e $v_3=3$. Dessa forma, utilizamos distribuições uniformes com os seguintes parâmetros: $(0, 10)$, $(0, 10)$, $(0, 0,26)$ para $v_1=2$; $(0, 5)$, $(0, 5)$, $(0, 0,88)$ para $v_2=2,5$ e $(0, 3)$, $(0, 3)$ e $(0, 2,9)$ para $v_3=3$, sendo que a 1ª e 2ª distribuições uniformes geram os candidatos a T_1 e T_2 respectivamente e a 3ª avalia se os candidatos não serão rejeitados.

Na Tabela 4.1 apresentamos as estimativas de MV e os intervalos de confiança assintóticos dos parâmetros da EBVE para tempos acelerados.

Tabela 4.1 Estimativas de MV e intervalos de confiança assintóticos dos parâmetros da distribuição EBVE para tempos acelerados, obtidas no exemplo de aplicação

	EMV	IC 95%
β_{01}^*	0,021	(0; 0,050)
β_{02}^*	0,018	(0; 0,042)
β_{03}^*	0,071	(0,031; 0,112)
β_{04}^*	0,171	(0,041; 0,300)
β_{05}^*	0,170	(0,060; 0,279)
β_{11}	3,295	(1,842; 4,748)
β_{12}	3,544	(2,177; 4,912)
β_{13}	2,803	(2,190; 3,416)

Obtivemos as estimativas de MV dos parâmetros nas condições usuais de operação, $v_0=1,5$, substituindo nas relações abaixo, as estimativas de MV apresentadas na Tabela 4.1. Essas estimativas e os intervalos de confiança assintóticos (1.35) são apresentados na Tabela 4.2.

$$\lambda_1 | v_0 = \beta_{01}^* v_0^{\beta_{11}}$$

$$\lambda_2 | v_0 = \beta_{02}^* v_0^{\beta_{12}}$$

$$\lambda_{12} | v_0 = \beta_{03}^* v_0^{\beta_{13}}$$

$$s_1 | v_0 = \beta_{04}^* v_0^{\beta_{13}}$$

$$s_2 | v_0 = \beta_{05}^* v_0^{\beta_{13}}$$

Tabela 4.2 Estimativas de MV e intervalos de confiança dos parâmetros na voltagem $v_0=1,5$.

	EMV	IC 95%
$\lambda_1 v_0$	0,080	(0,016; 0,143)
$\lambda_2 v_0$	0,077	(0,020; 0,134)
$\lambda_{12} v_0$	0,221	(0,144; 0,300)
$s_1 v_0$	0,532	(0,221; 0,843)
$s_2 v_0$	0,528	(0,280; 0,777)

Nas tabelas 4.3, 4.4 e 4.5 apresentamos as estimativas de MV das funções marginais de confiabilidade e de risco, avaliadas no tempo $t=1$, correspondentes as voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$, para os tempos dos componentes e também de suas partes, específica e comum, respectivamente, junto com os intervalos de confiança assintóticos.

Tabela 4.3 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$, $v_3=3$, correspondentes ao exemplo de aplicação

Componente i		$S_{T_i v_j}(t)$	IC 95%	$h_{T_i v_j}(t)$	IC 95%
1	v_0	0,879	(0,832; 0,926)	0,171	(0,114; 0,228)
	v_1	0,663	(0,614; 0,712)	0,551	(0,467; 0,634)
	v_2	0,374	(0,340; 0,408)	1,255	(1,144; 1,365)
	v_3	0,147	(0,110; 0,184)	2,288	(1,960; 2,617)
2	v_0	0,881	(0,837; 0,925)	0,168	(0,113; 0,222)
	v_1	0,659	(0,609; 0,706)	0,558	(0,475; 0,640)
	v_2	0,359	(0,325; 0,393)	1,296	(1,184; 1,409)
	v_3	0,131	(0,096; 0,166)	2,405	(2,078; 2,731)

Tabela 4.4 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, da parte específica de cada componente, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$.

Componente i		$S_{x_i v_j}(t)$	IC 95%	$h_{x_i v_j}(t)$	IC 95%
1	v_0	0,923	(0,865; 0,982)	0,080	(0,016; 0,143)
	v_1	0,814	(0,742; 0,887)	0,205	(0,117; 0,294)
	v_2	0,652	(0,575; 0,728)	0,428	(0,311; 0,545)
	v_3	0,458	(0,324; 0,591)	0,781	(0,489; 1,073)
2	v_0	0,926	(0,873; 0,978)	0,077	(0,020; 0,134)
	v_1	0,808	(0,739; 0,876)	0,214	(0,129; 0,298)
	v_2	0,624	(0,546; 0,703)	0,471	(0,345; 0,597)
	v_3	0,407	(0,270; 0,544)	0,899	(0,563; 1,235)

Tabela 4.5 - Estimativas de MV e intervalos de confiança das funções marginais de confiabilidade e de risco, da parte comum de cada componente, avaliadas no tempo $t=1$, nas voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$.

Componente i		$S_{z_i v_j}(t)$	IC 95%	$h_{z_i v_j}(t)$	IC 95%
1	v_0	0,952	(0,920; 0,983)	0,091	(0,035; 0,148)
	v_1	0,814	(0,756; 0,871)	0,345	(0,246; 0,444)
	v_2	0,574	(0,509; 0,639)	0,827	(0,686; 0,967)
	v_3	0,320	(0,233; 0,408)	1,507	(1,150; 1,864)
2	v_0	0,952	(0,924; 0,979)	0,091	(0,040; 0,141)
	v_1	0,815	(0,765; 0,864)	0,344	(0,255; 0,433)
	v_2	0,575	(0,508; 0,641)	0,825	(0,672; 0,978)
	v_3	0,321	(0,224; 0,418)	1,506	(1,134; 1,878)

Calculamos também os percentis 25, 50 e 75 das distribuições EBVE para tempos acelerados, considerando as voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$. Para isto, igualamos a p cada uma das funções de distribuição marginais e encontramos numericamente as soluções em t . Os valores são apresentados na Tabela 4.6, abaixo.

Tabela 4.6 Estimativas pontuais dos quartis da distribuição EBVE para tempos acelerados nos níveis de voltagens $v_0=1,5$, $v_1=2$, $v_2=2,5$ e $v_3=3$

componente i	v	$t_{0,25}$	$t_{0,50}$	$t_{0,75}$
1	v_0	1,808	3,463	5,924
	v_1	0,764	1,478	2,539
	v_2	0,390	0,761	1,312
	v_3	0,225	0,442	0,763
2	v_0	1,846	3,523	6,016
	v_1	0,759	1,471	2,528
	v_2	0,378	0,742	1,281
	v_3	0,212	0,421	0,732

4.2.2 Estudo de Simulação

Nesta seção fazemos algumas simulações para estudar as propriedades assintóticas dos estimadores de MV, dos parâmetros da distribuição EBVE para tempos acelerados, dada em 4.13. Foram utilizadas amostras com mesmo número de observações em cada nível $j=1, 2$ e 3 de voltagem, v_j e amostras com aproximadamente 50%, 30% e 20% dos dados para os níveis $j=1, 2$ e 3 , respectivamente, visando verificar se existe uma melhora nas estimativas de MV quando as amostras apresentam mais elementos nas voltagens mais baixas.

Os tempos até as falhas aceleradas $(T_1|v_j, T_2|v_j)$, em cada um dos níveis de voltagem foram gerados pelo método da rejeição, considerando os mesmos valores das voltagens e dos parâmetros utilizados no exemplo de aplicação na seção 4.2.1. O estudo foi feito com 500 amostras de tamanhos $n=60, 105, 300, 510$ e foram desprezadas as amostras que não convergiram ou cujas estimativas não pertenceram aos intervalos: $\hat{\beta}_{01}^* < 0,6$, $\hat{\beta}_{02}^* < 0,6$, $\hat{\beta}_{03}^* < 1,2$, $\hat{\beta}_{04}^* < 3$, $\hat{\beta}_{05}^* < 3$, $\hat{\beta}_{11}^* > 0$, $\hat{\beta}_{12}^* > 0$ e $\hat{\beta}_{13}^* > 0$. Uma descrição das amostras nessas condições é feita na Tabela 4.7. No Apêndice F apresentamos o programa computacional implementado em Ox (Doornik, 1999), para o estudo de simulação.

Para verificarmos a validade das amostras geradas comparamos a média das médias, das variâncias e da correlação entre os tempos (t_1, t_2) amostrais (Tabela 4.8) com os valores das distribuições, obtidos numericamente considerando os parâmetros utilizados na geração (Tabela 4.9). Com os resultados, observamos que os valores amostrais e os das distribuições parecem estar próximos e apresentam pequenos desvios padrões.

Para cada amostra gerada encontramos as estimativas de MV, utilizando o método de *quasi-Newton* BFGS. Nas Tabelas 4.10 e 4.12 apresentamos as médias das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da EBVE para tempos acelerados e na voltagem usual, $v_0=1,5$, respectivamente. As Figuras 4.2 e 4.3 são correspondentes a estas tabelas. Com os resultados obtidos verificamos que os erros quadráticos médios dos estimadores dos parâmetros na voltagem usual e de β_{01}^* , β_{02}^* , β_{03}^* , β_{04}^* e β_{05}^* estão convergindo rapidamente para zero parecendo ser menores do que os de β_{11} , β_{12} e β_{13} . Observamos também que os estimadores obtidos nas amostras com maior número de elementos nas voltagens mais baixas parecem ser mais eficientes dos que os obtidos considerando amostras com tamanhos iguais.

Na Tabela 4.11 e Figura 4.4 estão as coberturas dos intervalos de 90% e 95% de confiança dos parâmetros da EBVE para tempos acelerados e na Tabela 4.13 e Figura 4.5 as dos parâmetros na voltagem usual. Vemos que a cobertura dos intervalos de confiança parece se aproximar da cobertura nominal para $n > 500$.

Tabela 4.7 – Descrição das amostras simuladas com tempos acelerados

condições	tamanho das amostras							
	20+20 +20	30+20 +10	35+35 +35	55+30+ 20	100+100 +100	150+90 +60	170+170 +170	260+150 +100
$\beta_{01}^* > 0.6$	21*	67	15	24	1	1	0	0
$\beta_{02}^* > 0.6$	29	64	25	22	0	0	0	0
$\beta_{03}^* > 1.2$	1	6	0	1	0	0	0	0
$\beta_{04}^* > 3$	12	19	7	2	0	0	0	0
$\beta_{05}^* > 3$	14	19	1	3	0	0	0	0
$\beta_{11} < 0$	84	91	36	45	3	4	0	0
$\beta_{12} < 0$	92	95	39	35	0	5	1	0
$\beta_{13} < 0$	6	10	0	0	0	0	0	0
$\beta_{01}^* > 0.6, \beta_{02}^* > 0.6, \beta_{03}^* > 1.2,$ $\beta_{04}^* > 3, \beta_{05}^* > 3, \beta_{11} < 0, \beta_{12} < 0, \beta_{13} < 0$	179	205	77	78	3	9	1	0
não convergiram	464	455	135	130	7	3	2	1
covariância negativa	6	4	3	0	0	1	0	0

* Número de amostras

Tabela 4.8 - Média das médias, variâncias e correlações dos tempos (t_1, t_2) das 500 amostras geradas da EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$

n	v	médias t_1 média (dp)	médias t_2 média (dp)	var t_1 média (dp)	var t_2 média (dp)	corr(t_1, t_2) média (dp)
20+20+20	v_1	1,83 (0,31)	1,82 (0,31)	2,15 (1,05)	2,16 (1,02)	0,32 (0,25)
	v_2	0,89 (0,16)	0,88 (0,17)	0,52 (0,27)	0,52 (0,27)	0,30 (0,25)
	v_3	0,55 (0,09)	0,55 (0,09)	0,19 (0,09)	0,19 (0,09)	0,30 (0,25)
30+20+10	v_1	1,82 (0,26)	1,82 (0,25)	2,19 (0,85)	2,15 (0,81)	0,33 (0,21)
	v_2	0,90 (0,17)	0,91 (0,17)	0,54 (0,28)	0,55 (0,28)	0,29 (0,26)
	v_3	0,57 (0,14)	0,54 (0,13)	0,20 (0,14)	0,19 (0,13)	0,30 (0,36)
35+35+35	v_1	1,82 (0,26)	1,81 (0,25)	2,16 (0,84)	2,12 (0,79)	0,30 (0,20)
	v_2	0,91 (0,12)	0,91 (0,12)	0,54 (0,21)	0,56 (0,20)	0,32 (0,20)
	v_3	0,54 (0,07)	0,54 (0,07)	0,19 (0,07)	0,19 (0,07)	0,32 (0,20)
55+30+20	v_1	1,81 (0,19)	1,82 (0,20)	2,16 (0,67)	2,20 (0,66)	0,34 (0,15)
	v_2	0,91 (0,14)	0,91 (0,13)	0,53 (0,22)	0,55 (0,21)	0,30 (0,21)
	v_3	0,54 (0,10)	0,54 (0,10)	0,19 (0,10)	0,18 (0,09)	0,30 (0,26)
100+100+100	v_1	1,81 (0,15)	1,81 (0,15)	2,15 (0,51)	2,14 (0,47)	0,32 (0,12)
	v_2	0,92 (0,07)	0,92 (0,07)	0,56 (0,12)	0,55 (0,12)	0,32 (0,12)
	v_3	0,53 (0,04)	0,54 (0,04)	0,19 (0,04)	0,19 (0,04)	0,32 (0,12)
150+90+60	v_1	1,80 (0,12)	1,81 (0,12)	2,16 (0,40)	2,18 (0,38)	0,32 (0,10)
	v_2	0,92 (0,08)	0,92 (0,08)	0,56 (0,13)	0,56 (0,12)	0,31 (0,13)
	v_3	0,54 (0,06)	0,54 (0,06)	0,19 (0,05)	0,19 (0,05)	0,32 (0,16)
170+170+170	v_1	1,80 (0,11)	1,81 (0,11)	2,15 (0,36)	2,17 (0,36)	0,32 (0,10)
	v_2	0,92 (0,06)	0,92 (0,05)	0,56 (0,10)	0,56 (0,09)	0,31 (0,10)
	v_3	0,53 (0,03)	0,53 (0,03)	0,19 (0,03)	0,19 (0,03)	0,32 (0,10)
260+150+100	v_1	1,80 (0,09)	1,81 (0,09)	2,15 (0,31)	2,17 (0,30)	0,32 (0,08)
	v_2	0,92 (0,06)	0,92 (0,06)	0,56 (0,10)	0,55 (0,10)	0,31 (0,10)
	v_3	0,54 (0,04)	0,54 (0,05)	0,19 (0,04)	0,19 (0,04)	0,31 (0,12)

Tabela 4.9 – Resumo dos valores teóricos da distribuição EBVE para tempos acelerados

com $\beta_{01}^* = \beta_{02}^* = 0,029$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$.

	$E(T_1)$	$E(T_2)$	$Var(T_1)$	$Var(T_2)$	$Corr(T_1, T_2)$
v_1	1,81	1,81	2,29	2,29	0,32
v_2	0,93	0,93	0,59	0,59	0,32
v_3	0,54	0,54	0,20	0,20	0,32

Tabela 4.10 - Média das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros da distribuição EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$ e $\beta_{11} = \beta_{12} = \beta_{13} = 3$

.média (dp)	Tamanho das amostras							
	20+20+20	30+20+10	35+35+35	55+30+20	100+100+100	150+90+60	170+170+170	260+150+100
β_{01}^*	0,06 (0,08)	0,06 (0,07)	0,05 (0,06)	0,05 (0,06)	0,04 (0,04)	0,04 (0,03)	0,03 (0,02)	0,03 (0,02)
β_{02}^*	0,06 (0,08)	0,05 (0,06)	0,05 (0,06)	0,05 (0,06)	0,04 (0,04)	0,04 (0,04)	0,03 (0,02)	0,03 (0,02)
β_{03}^*	0,09 (0,08)	0,10 (0,10)	0,08 (0,05)	0,08 (0,05)	0,07 (0,03)	0,07 (0,02)	0,06 (0,02)	0,07 (0,02)
β_{04}^*	0,36 (0,72)	0,31 (0,38)	0,20 (0,23)	0,22 (0,24)	0,16 (0,08)	0,16 (0,07)	0,15 (0,05)	0,16 (0,06)
β_{05}^*	0,41 (0,79)	0,31 (0,37)	0,21 (0,26)	0,21 (0,26)	0,16 (0,07)	0,16 (0,08)	0,15 (0,05)	0,15 (0,05)
β_{11}	3,12 (1,64)	3,11 (1,64)	3,15 (1,36)	3,12 (1,47)	3,02 (0,94)	2,94 (0,94)	2,99 (0,65)	2,96 (0,67)
β_{12}	3,19 (1,59)	3,27 (1,67)	3,05 (1,37)	3,10 (1,41)	2,99 (0,93)	2,95 (0,94)	3,03 (0,65)	3,02 (0,70)
β_{13}	2,95 (0,96)	2,87 (0,96)	3,00 (0,69)	2,96 (0,71)	3,00 (0,40)	3,01 (0,03)	3,01 (0,30)	2,97 (0,31)
$dp\beta_{01}^*$	0,10 (0,14)	0,10 (0,13)	0,06 (0,08)	0,07 (0,10)	0,03 (0,03)	0,03 (0,03)	0,02 (0,01)	0,02 (0,01)
$dp\beta_{02}^*$	0,10 (0,13)	0,10 (0,13)	0,07 (0,10)	0,07 (0,09)	0,03 (0,03)	0,03 (0,01)	0,02 (0,01)	0,02 (0,01)
$dp\beta_{03}^*$	0,08 (0,10)	0,10 (0,15)	0,05 (0,04)	0,05 (0,04)	0,03 (0,01)	0,02 (0,04)	0,02 (0,004)	0,02 (0,01)
$dp\beta_{04}^*$	0,47 (1,30)	0,35 (0,53)	0,17 (0,25)	0,19 (0,29)	0,07 (0,04)	0,07 (0,04)	0,05 (0,02)	0,05 (0,02)
$dp\beta_{05}^*$	0,52 (1,30)	0,36 (0,56)	0,18 (0,31)	0,18 (0,32)	0,07 (0,03)	0,07 (0,23)	0,05 (0,02)	0,05 (0,02)
$dp\beta_{11}$	2,16 (1,10)	2,27 (0,93)	1,63 (0,62)	1,74 (0,71)	0,91 (0,21)	0,94 (0,22)	0,67 (0,12)	0,69 (0,11)
$dp\beta_{12}$	2,12 (0,93)	2,45 (1,46)	1,67 (0,83)	1,73 (0,76)	0,90 (0,20)	0,93 (0,05)	0,66 (0,09)	0,68 (0,10)
$dp\beta_{13}$	0,95 (0,33)	1,00 (0,33)	0,68 (0,18)	0,71 (0,20)	0,38 (0,05)	0,39 (0,01)	0,29 (0,03)	0,30 (0,03)
$eqm\beta_{01}^*$	0,04 (0,09)	0,03 (0,09)	0,02 (0,05)	0,02 (0,06)	0,004 (0,01)	0,003 (0,01)	0,001 (0,003)	0,001 (0,002)
$eqm\beta_{02}^*$	0,03 (0,09)	0,03 (0,09)	0,02 (0,07)	0,02 (0,05)	0,003 (0,01)	0,003 (0,002)	0,001 (0,003)	0,001 (0,002)
$eqm\beta_{03}^*$	0,02 (0,08)	0,04 (0,26)	0,01 (0,02)	0,01 (0,02)	0,002 (0,002)	0,001 (0,02)	0,001 (0,001)	0,001 (0,001)
$eqm\beta_{04}^*$	2,47 (22,2)	0,57 (2,09)	0,14 (1,01)	0,18 (0,89)	0,01 (0,03)	0,01 (0,02)	0,01 (0,008)	0,01 (0,01)
$eqm\beta_{05}^*$	2,65 (18,5)	0,60 (2,31)	0,20 (1,27)	0,21 (1,29)	0,01 (0,01)	0,01 (0,02)	0,01 (0,008)	0,01 (0,01)
$eqm\beta_{11}$	8,55 (10,0)	8,71 (7,12)	4,93 (4,12)	5,68 (5,04)	1,75 (1,54)	1,81 (1,63)	0,88 (0,77)	0,94 (0,67)
$eqm\beta_{12}$	7,91 (6,92)	11,0 (15,9)	5,37 (8,37)	5,57 (5,57)	1,72 (1,53)	1,79 (1,61)	0,88 (0,66)	0,96 (0,79)
$eqm\beta_{13}$	1,92 (2,24)	2,05 (2,08)	0,97 (0,85)	1,05 (1,13)	0,31 (0,25)	0,32 (0,22)	0,17 (0,14)	0,19 (0,14)

Tabela 4.11 - Cobertura dos intervalos de confiança assintóticos dos parâmetros da distribuição EBVE para tempos acelerados

n	β_{01}^*	β_{02}^*	β_{03}^*	β_{04}^*	β_{05}^*	β_{11}	β_{12}	β_{13}
20+20+20	79,8 ^a	78,6	87,2	86,8	87,4	93,6	95,0	92,8
	81,8 ^b	79,8	89,6	88,2	89,2	97,0	97,8	96,8
30+20+10	80,2	77,0	90,2	86,8	89,6	94,4	95,2	93,8
	82,8	80,2	92,0	90,0	90,6	97,0	97,0	97,2
35+35+35	78,6	82,4	88,6	85,6	84,2	93,4	95,4	91,0
	82,4	84,6	91,0	88,2	88,0	96,8	97,0	95,8
55+30+20	77,4	81,2	91,4	86,8	85,6	93,8	93,6	92,4
	80,0	83,6	94,4	88,2	89,4	96,8	96,6	97,2
100+100+100	84,0	86,0	89,6	87,8	85,4	91,4	90,6	88,6
	86,6	88,2	93,6	90,0	88,0	96,0	95,6	93,4
150+90+60	84,8	85,0	91,6	87,2	85,4	93,2	90,4	89,4
	87,2	88,2	95,4	90,4	89,4	96,4	94,6	95,0
170+170+170	89,2	86,6	90,4	88,2	87,0	91,0	90,4	89,6
	92,0	90,0	95,0	91,6	89,4	96,0	96,2	94,8
260+150+100	89,2	87,8	92,8	87,2	87,6	91,4	89,4	89,2
	91,8	89,4	95,8	91,2	91,4	97,2	94,6	95,0

^{a,b} Coeficientes de confiança 90% e 95%, respectivamente

Tabela 4.12 - Média das 500 estimativas de MV, dos desvios padrões e dos erros quadráticos médios dos estimadores dos parâmetros na voltagem usual, $v_0=1,5$, dados por $\lambda_1|v_0=\lambda_2|v_0=0,1$, $\lambda_{12}|v_0=0,2$ e $s_1|v_0=s_2|v_0=0,5$.

Média (dp)	20+20+20	30+20+10	35+35+35	55+30+20	100+100+100	150+90+60	170+170+170	260+150+100
$\lambda_1 v_0$	0,12 (0,10)	0,12 (0,09)	0,11 (0,08)	0,11(0,07)	0,11 (0,05)	0,11 (0,04)	0,10 (0,04)	0,10 (0,03)
$\lambda_2 v_0$	0,12 (0,10)	0,11 (0,08)	0,12 (0,08)	0,11 (0,07)	0,11 (0,05)	0,11 (0,05)	0,10 (0,04)	0,10 (0,03)
$\lambda_{12} v_0$	0,25 (0,14)	0,26 (0,17)	0,23 (0,10)	0,24 (0,10)	0,22 (0,05)	0,21 (0,05)	0,21 (0,04)	0,21 (0,03)
$s_1 v_0$	0,97 (1,47)	0,84 (0,90)	0,63 (0,68)	0,67 (0,85)	0,51 (0,19)	0,50 (0,18)	0,50 (0,14)	0,50 (0,14)
$s_2 v_0$	1,17 (2,23)	0,89 (1,24)	0,64 (0,72)	0,63 (0,69)	0,51 (0,18)	0,51 (0,19)	0,50 (0,14)	0,50 (0,13)
$dp\lambda_1 v_0$	0,12 (0,08)	0,11 (0,07)	0,09 (0,05)	0,08 (0,05)	0,05 (0,02)	0,05 (0,02)	0,04 (0,01)	0,03 (0,01)
$dp\lambda_2 v_0$	0,11 (0,08)	0,11 (0,07)	0,09 (0,06)	0,08 (0,05)	0,05 (0,02)	0,04 (0,02)	0,04 (0,01)	0,03 (0,01)
$dp\lambda_{12} v_0$	0,14 (0,15)	0,15 (0,28)	0,09 (0,06)	0,09 (0,05)	0,05 (0,01)	0,05 (0,01)	0,04 (0,01)	0,04 (0,01)
$dps_1 v_0$	0,91 (2,00)	0,71 (1,12)	0,40 (0,69)	0,46 (1,35)	0,17 (0,08)	0,17 (0,07)	0,13 (0,04)	0,13 (0,04)
$dps_2 v_0$	1,21 (3,45)	0,79 (1,65)	0,42 (0,85)	0,40 (0,97)	0,17 (0,07)	0,17 (0,07)	0,13 (0,04)	0,13 (0,04)
$eqm\lambda_1 v_0$	0,03 (0,05)	0,03 (0,04)	0,09 (0,03)	0,01 (0,02)	0,01 (0,01)	0,004 (0,01)	0,003 (0,003)	0,002 (0,002)
$eqm\lambda_2 v_0$	0,03 (0,04)	0,02 (0,03)	0,09 (0,03)	0,01 (0,02)	0,01 (0,01)	0,004 (0,01)	0,003 (0,003)	0,002 (0,002)
$eqm\lambda_{12} v_0$	0,06 (0,42)	0,14 (1,54)	0,09 (0,07)	0,02 (0,05)	0,01 (0,01)	0,01 (0,01)	0,003 (0,003)	0,003 (0,002)
$eqms_1 v_0$	7,22 (45,6)	2,70 (13,0)	0,40 (11,3)	2,78 (39,4)	0,07 (0,12)	0,06 (0,11)	0,04 (0,04)	0,04 (0,06)
$eqms_2 v_0$	18,8 (141)	5,02 (35,5)	0,42 (14,3)	1,59 (21,6)	0,07 (0,08)	0,07 (0,09)	0,04 (0,04)	0,04 (0,04)
tempo	17:17'	18:56'	15:56'	18:40'	24:24'	23:04'	37:01'	38:10'

Tabela 4.13 - Cobertura dos intervalos com 90% e 95% de confiança dos parâmetros da EBVE, na voltagem $v_0=1,5$

n	$\lambda_1 v_0$	$\lambda_2 v_0$	$\lambda_{12} v_0$	$s_1 v_0$	$s_2 v_0$
20+20+20	86,0 ^a	84,2	90,2	89,0	89,2
	88,4 ^b	88,2	92,4	90,4	91,4
30+20+10	86,8	84,0	92,8	88,4	89,4
	89,2	87,2	94,2	91,6	91,2
35+35+35	85,4	87,8	90,8	87,8	85,6
	89,2	90,4	94,6	91,0	89,0
55+30+20	84,4	86,2	91,4	87,0	87,2
	89,2	89,4	96,4	90,6	90,4
100+100+100	87,6	87,8	89,2	86,0	86,2
	92,4	93,2	95,4	90,2	89,8
150+90+60	88,0	87,4	90,8	86,8	85,2
	92,6	92,4	96,0	91,8	90,8
170+170+170	89,8	88,4	89,6	88,4	86,8
	94,2	93,0	95,0	92,4	90,6
260+150+100	91,6	89,0	89,6	86,8	87,2
	94,8	92,8	95,8	92,0	92,2

^{a,b} Cobertura dos intervalos de confiança 90% e 95% respectivamente

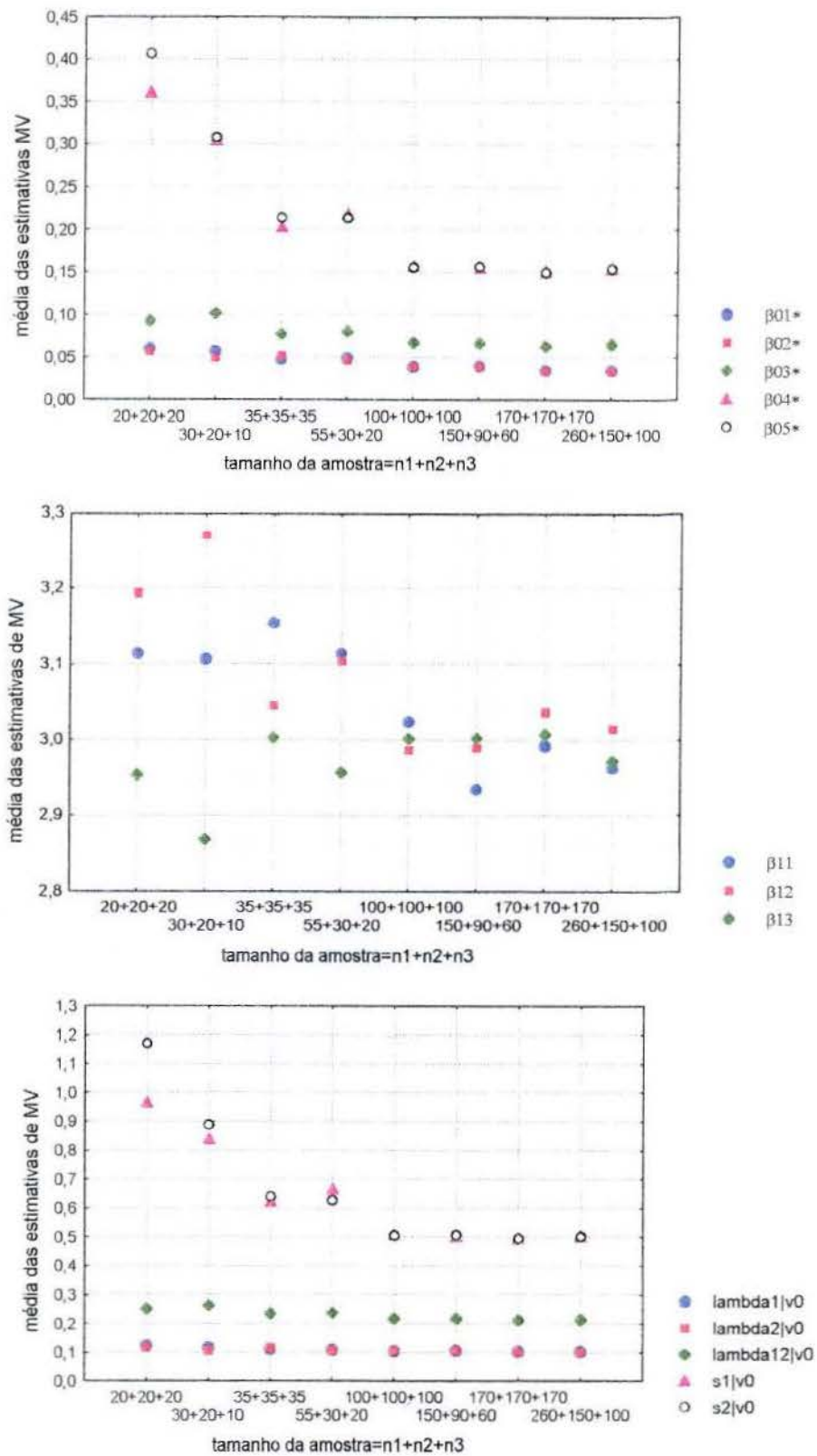


Figura 4.2-Gráficos das médias das estimativas de MV dos parâmetros do modelo para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$ e na voltagem usual, $v_0 = 1,5$, onde $\lambda_{1|v_0} = \lambda_{2|v_0} = 0,1$, $\lambda_{12|v_0} = 0,2$ e $s_{1|v_0} = s_{2|v_0} = 0,5$

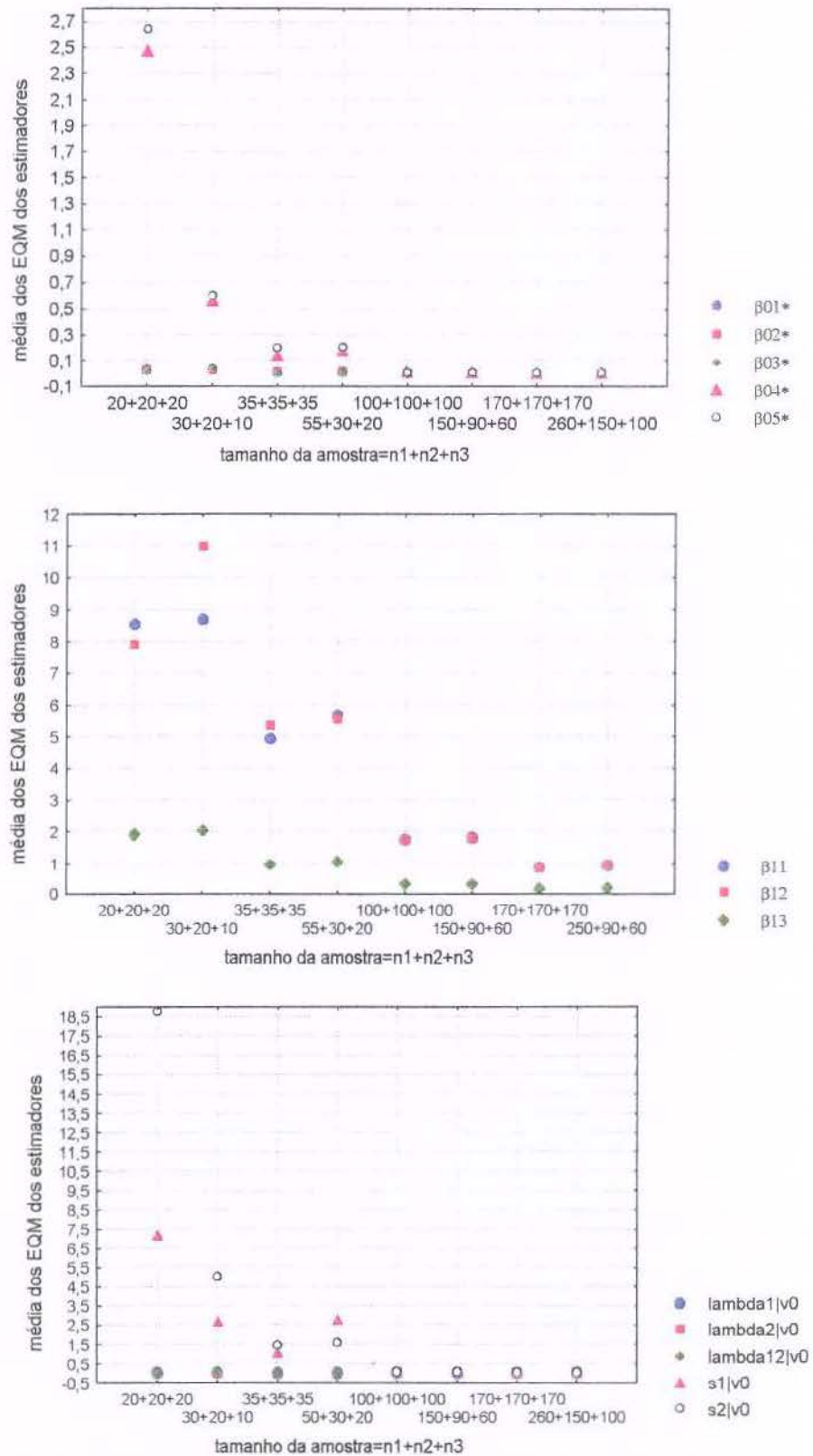


Figura 4.3 - Gráficos das médias dos erros quadráticos médios dos estimadores dos parâmetros do modelo para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$ e na voltagem usual, $v_0 = 1,5$, onde $\lambda_1|v_0 = 0,1$, $\lambda_2|v_0 = \lambda_{12}|v_0 = 0,2$ e $s_1|v_0 = s_2|v_0 = 0,5$

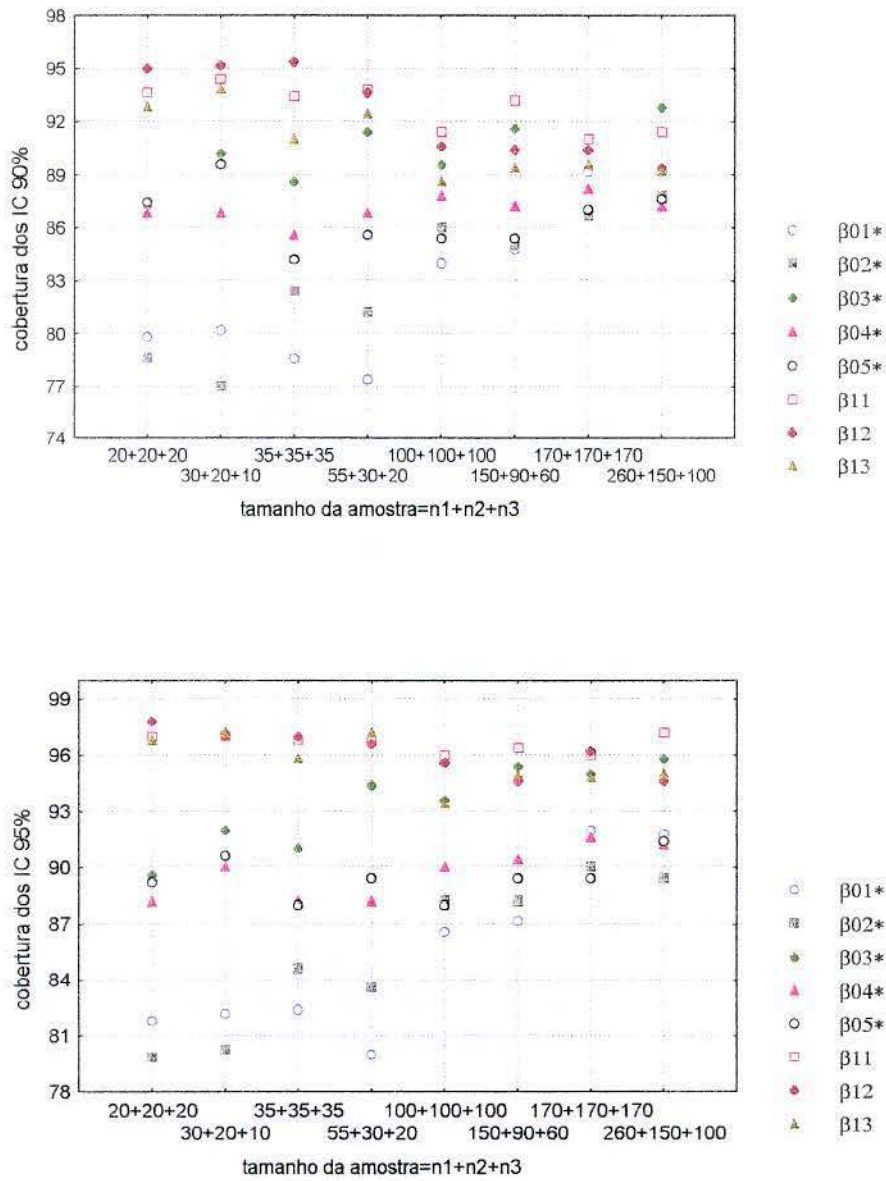


Figura 4.4 – Cobertura dos intervalos aproximados, com 90% e 95% de confiança, dos parâmetros da distribuição EBVE para tempos acelerados com $\beta_{01}^* = \beta_{02}^* = 0,0296$, $\beta_{03}^* = 0,0593$, $\beta_{04}^* = \beta_{05}^* = 0,148$, $\beta_{11} = \beta_{12} = \beta_{13} = 3$.

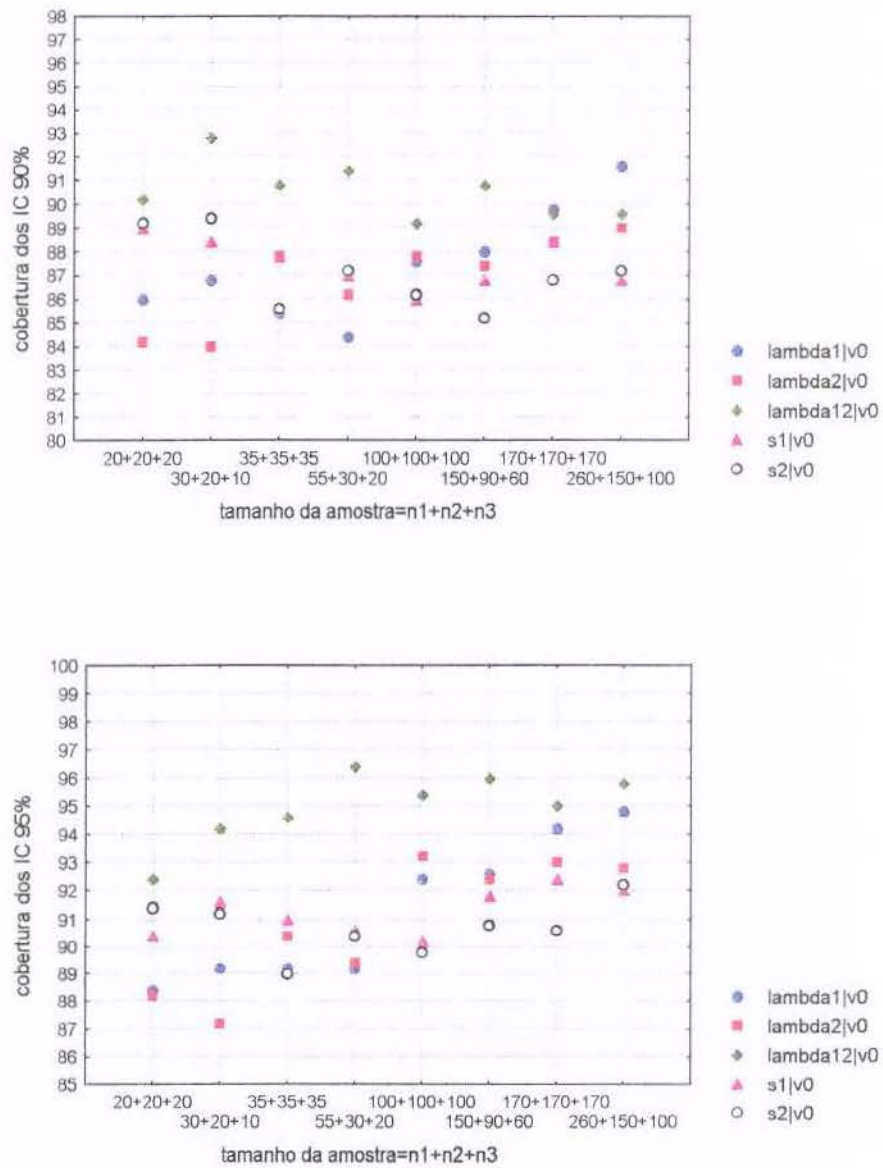


Figura 4.5 – Cobertura dos intervalos aproximados, com 90% e 95% de confiança, dos parâmetros na voltagem usual, $v_0=1,5$.

CONCLUSÕES

De acordo com os resultados obtidos com as simulações chegamos às seguintes conclusões:

i) As amostras da distribuição EBVE, geradas pelo método da rejeição, pareceram confiáveis quando $n > 100$.

ii) Considerando dados sem censura, as estimativas de MV de λ_1 e λ_2 apresentaram vícios e variabilidade pequenos, mesmo para amostras de tamanho $n=30$, enquanto que as estimativas de λ_{12} foram eficientes para amostras maiores do que 100 e as de s_1 e s_2 para amostras com $n > 300$. A distribuição dos estimadores de MV parece não se aproximar da normal multivariada, mesmo para amostras de tamanho $n=500$.

iii) Nas amostras com censuras do tipo I as estimativas de MV apresentaram vícios grandes, mesmo considerando uma pequena probabilidade de censura. As coberturas dos intervalos de confiança assintóticos não foram próximas da cobertura nominal, possivelmente devido ao fato da variância dos estimadores estarem convergindo para zero mais rapidamente do que os seus vícios.

iv) Na análise Bayesiana, as densidades *a posteriori* foram influenciadas consideravelmente pelas densidades *a priori*. As médias e medianas das densidades *a posteriori* foram mais próximas dos valores dos parâmetros quando assumimos densidades *a priori* com melhor informação, como era esperado.

v) No modelo para tempos acelerados, os estimadores de MV apresentaram vícios e variabilidade pequenos para amostras com $n > 300$.

Em relação a trabalhos futuros, na análise clássica seria interessante utilizar algumas reparametrizações para tentar melhorar a normalidade dos estimadores de MV e usar o método *Bootstrap* para estimar as variâncias dos estimadores. Na análise Bayesiana

poderíamos estudar a sensibilidade das densidades *a priori* no modelo, também assumindo densidades *a priori* não informativas.

APÊNDICE A

FUNÇÕES DE SOBREVIVÊNCIA DA EBVE

A₁ - Obtenção da Função de Sobrevivência de Z_i , $i=1, 2$

Neste apêndice obteremos a função de sobrevivência de Z_i . Com esse objetivo precisaremos dos seguintes resultados:

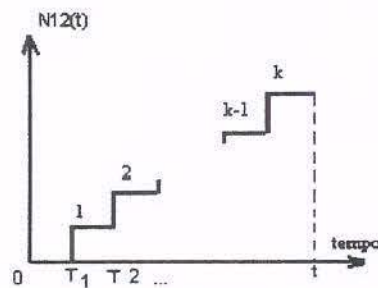


Figura 1. Processo de Poisson $\{N_{12}(t), t \geq 0\}$.

Lema: $\tau_1, \tau_2, \dots, \tau_k$ quando não ordenados, são distribuídos independentemente e uniformemente em $(0, t]$, demonstração em Barlow e Proschan, 1981.

Corolário: $t - \tau_1, t - \tau_2, \dots, t - \tau_k$, quando não ordenados, são independentes e identicamente distribuídos $U(0, t]$.

De (1.4) obtemos a função de risco de Z_i no tempo t , condicional a realização do processo $\{N_{12}(t), t \geq 0; \lambda_{12}\}$, dada por:

$$h_{Z_i|N_{12}}(t) = s_i N_{12}(t) \tag{a_1 1}$$

e utilizando a função de risco anterior obtemos a função de sobrevivência condicional à realização do processo $\{N_{12}(t), t \geq 0; \lambda_{12}\}$, dada por:

$$S_{Z_i|N_{12}}(t) = P(Z_i > t | N_{12}) = \exp\left(-s_i \int_0^t N_{12}(u) du\right). \quad (a_12)$$

Podemos escrever a função de sobrevivência de Z_i na forma

$$S_{Z_i}(t) = P(Z_i > t) = \sum_{k=0}^{\infty} P(Z_i > t | N_{12} = k) P(N_{12} = k), \quad (a_13)$$

sendo que em (a₁₃) usamos o teorema da probabilidade total. Notemos que para t fixo, $P(Z_i > t | N_{12} = k)$ é uma função de k , digamos $g(k)$ e portanto,

$$S_{Z_i}(t) = \sum_{k=0}^{\infty} g(k) P(N_{12} = k) = E(g(N_{12})) = E(P(Z_i > t | N_{12})) = E(S_{Z_i|N_{12}}(t)). \quad (a_14)$$

Agora, usando (a₁₂) em (a₁₄) temos que,

$$E(S_{Z_i|N_{12}}(t)) = E\left(\exp\left(-s_i \int_0^t N_{12}(u) du\right)\right) \quad (a_15)$$

e pelas propriedades de esperança, veja James, 1981, temos que,

$$E\left(\exp\left(-s_i \int_0^t N_{12}(u) du\right)\right) = \sum_{k=0}^{\infty} P(N_{12} = k) E\left(\exp\left(-s_i \int_0^t N_{12}(u) du\right) | N_{12} = k\right). \quad (a_16)$$

Como $\{N_{12}(t), t \geq 0\}$ é uma função escada contínua a direita e com os limites a esquerda, veja Figura 1, no início deste apêndice, então

$$\int_0^t N_{12}(u) du = (t - \tau_1) + (t - \tau_2) + \dots + (t - \tau_k) \quad (a_17)$$

e das equações anteriores temos que,

$$S_{Z_i}(t) = \sum_{k=0}^{\infty} P(N_{12} = k) E(\exp(-s_i((t - \tau_1) + (t - \tau_2) + \dots + (t - \tau_k)))) \quad (a_18)$$

$$= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t} (\lambda_{12}t)^k}{k!} E(\exp(-s_i(t - \tau_1)k)) \quad (a_19)$$

$$\begin{aligned}
 &= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t} (\lambda_{12}t)^k}{k!} E(\exp(-s_i(t - \tau_1)))^k \\
 &= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t} (\lambda_{12}t)^k}{k!} \frac{(1 - e^{-s_i t})^k}{(s_i t)^k},
 \end{aligned} \tag{a_110}$$

sendo que em (a₁₉) e em (a₁₁₀) utilizamos a conclusão do corolário apresentado no início deste apêndice. Finalmente, temos que

$$S_{Z_1}(t) = e^{-\lambda_{12}t} \sum_{k=0}^{\infty} \frac{\left(\frac{\lambda_{12}}{s_i} (1 - e^{-s_i t}) \right)^k}{k!} = \exp\left(-\lambda_{12}t + \frac{\lambda_{12}}{s_i} (1 - e^{-s_i t}) \right). \tag{a_111}$$

A₂ - Obtenção da Função de Sobrevivência de (T₁, T₂)

Neste apêndice obteremos a função de sobrevivência conjunta de (T₁, T₂). Iniciamos a partir da função de sobrevivência condicional de (T₁, T₂) dada a realização de {N₁₂(t), t ≥ 0; λ₁₂}.

$$\begin{aligned}
 S_{T_1, T_2 | N_{12}}(t_1, t_2) &= P(T_1 > t_1, T_2 > t_2 | N_{12}) \\
 &= P(\min(X_1, Z_1) > t_1, \min(X_2, Z_2) > t_2 | N_{12}) \\
 &= P(X_1 > t_1, Z_1 > t_1, X_2 > t_2, Z_2 > t_2 | N_{12}) \\
 &= P(X_1 > t_1) P(X_2 > t_2) P(Z_1 > t_1, Z_2 > t_2 | N_{12}) \\
 &= P(X_1 > t_1) P(X_2 > t_2) \exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du \right)
 \end{aligned} \tag{a_21}$$

Podemos escrever a função de sobrevivência conjunta de (T₁, T₂) na forma,

$$S_{T_1, T_2}(t_1, t_2) = P(T_1 > t_1, T_2 > t_2) = \sum_{k=0}^{\infty} P(T_1 > t_1, T_2 > t_2 | N_{12} = k) P(N_{12} = k) \tag{a_22}$$

pelo teorema da probabilidade total. Agora, utilizando a mesma propriedade em (a₁₄) temos que,

$$S_{T_1, T_2}(t_1, t_2) = E(S_{T_1, T_2 | N_{12}}(t_1, t_2)). \tag{a_23}$$

Fazendo uso de (a₂1) em (a₂3) temos que,

$$\begin{aligned} E(S_{T_1, T_2 | N_{12}}(t_1, t_2)) &= E\left(P(X_1 > t_1) P(X_2 > t_2) \exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du \right) \right) \\ &= \exp(-\lambda_1 t_1) \exp(-\lambda_2 t_2) E\left(\exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du \right) \right), \quad (a_2 4) \end{aligned}$$

pois $X_i, i=1,2$, têm distribuição exponencial e são independentes.

Estudamos agora o termo direito na última igualdade no caso em que $t_1 \leq t_2$. Temos então que

$$\begin{aligned} &E\left(\exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du \right) \right) \\ &= E\left(\exp\left(-\left(s_1 \int_0^{t_1} N_{12}(u) du + s_2 \int_0^{t_1} N_{12}(u) du + s_2 \int_{t_1}^{t_2} N_{12}(u) du \right) \right) \right) \\ &= E\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du + s_2 \int_{t_1}^{t_2} N_{12}(u) du \right) \right) \right) \\ &= E\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du + s_2 \int_{t_1}^{t_2} N_{12}(u) du + s_2 \int_{t_1}^{t_2} N_{12}(t_1) du - s_2 \int_{t_1}^{t_2} N_{12}(t_1) du \right) \right) \right) \\ &= E\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du + s_2 \int_{t_1}^{t_2} (N_{12}(u) - N_{12}(t_1)) du + s_2 (t_2 - t_1) N_{12}(t_1) \right) \right) \right) \end{aligned} \quad (a_2 5)$$

$$= E\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du + s_2 (t_2 - t_1) N_{12}(t_1) \right) \right) E\left(\exp\left(-s_2 \int_{t_1}^{t_2} [N_{12}(u) - N_{12}(t_1)] du \right) \right) \right) \quad (a_2 6)$$

sendo que em (a₂6) utilizamos a propriedade de incrementos independentes do processo de Poisson. Em relação a primeira esperança no lado esquerdo da última equação temos que,

$$\begin{aligned} &E\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du + s_2 (t_2 - t_1) N_{12}(t_1) \right) \right) \right) \\ &= \sum_{k=0}^{\infty} P(N_{12} = k) \exp(-s_2 (t_2 - t_1) k) E\left(\left(\exp\left(-\left((s_1 + s_2) \int_0^{t_1} N_{12}(u) du \right) \right) \middle| N_{12} = k \right) \right) \end{aligned} \quad (a_2 7)$$

onde utilizamos a propriedade de esperança, veja James, 1981 e o princípio da substituição de esperança condicional. Logo, usando (a₁₇) e o corolário no início do Apêndice A₁ em (a₂₇) temos que

$$\begin{aligned}
 & E\left(\exp\left(-\left(s_1 + s_2\right)\int_0^{t_1} N_{12}(u)du + s_2(t_2 - t_1)N_{12}(t_1)\right)\right) \\
 &= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t_1}(\lambda_{12}t_1)^k}{k!} e^{-s_2(t_2-t_1)k} E(\exp(-(s_1 + s_2)((t_1 - \tau_1) + \dots + (t_k - \tau_k)))) \\
 &= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t_1}(\lambda_{12}t_1)^k}{k!} e^{-s_2(t_2-t_1)k} E(\exp(-(s_1 + s_2)(t_1 - \tau_1))^k) \\
 &= \sum_{k=0}^{\infty} \frac{e^{-\lambda_{12}t_1}(\lambda_{12}t_1)^k}{k!} e^{-s_2(t_2-t_1)k} (E \exp(-(s_1 + s_2)(t_1 - \tau_1)))^k \\
 &= e^{-\lambda_{12}t_1} \sum_{k=0}^{\infty} \frac{(\lambda_{12}t_1)^k}{k!} e^{-s_2(t_2-t_1)k} \left(\frac{1 - e^{-(s_1+s_2)t_1}}{(s_1 + s_2)t_1}\right)^k \\
 &= e^{-\lambda_{12}t_1} \sum_{k=0}^{\infty} \frac{\left(\frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1t_1 - s_2t_2})\right)^k}{k!} \\
 &= e^{-\lambda_{12}t_1} \exp\left(\frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1t_1 - s_2t_2})\right) \\
 &= \exp\left(-\lambda_{12}t_1 + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1t_1 - s_2t_2})\right). \tag{a_28}
 \end{aligned}$$

Ainda no caso $t_1 \leq t_2$ temos que a segunda esperança no lado direito de (a₂₆) é obtida em forma similar a esperança em (a₁₅), assim temos,

$$\begin{aligned}
 & E\left(\exp\left(-s_2\int_{t_1}^{t_2} (N_{12}(u) - N_{12}(t_1))du\right)\right) \\
 &= \exp\left(-\lambda_{12}(t_2 - t_1) + \frac{\lambda_{12}}{s_2}(1 - e^{-s_2(t_2-t_1)})\right). \tag{a_29}
 \end{aligned}$$

Finalmente usando (a₂₈) e (a₂₉) em (a₂₄) obtemos que

$$\begin{aligned}
 S_{T_1, T_2}(t_1, t_2) &= e^{-\lambda_1 t_1} e^{-\lambda_2 t_2} \exp\left(-\lambda_{12} t_2 + \frac{\lambda_{12}}{s_2} (1 - e^{-s_2(t_2-t_1)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2})\right) \\
 &= \exp\left(-\lambda_1 t_1 - (\lambda_2 + \lambda_{12}) t_2 + \frac{\lambda_{12}}{s_2} (1 - e^{-s_2(t_2-t_1)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_2(t_2-t_1)} - e^{-s_1 t_1 - s_2 t_2})\right).
 \end{aligned}
 \tag{a_2 10}$$

O caso $t_1 > t_2$ é análogo e por isso omitiremos maiores detalhes. Temos para $t_1 > t_2$

$$S_{T_1, T_2}(t_1, t_2) = \exp\left(-(\lambda_1 + \lambda_{12}) t_1 - \lambda_2 t_2 + \frac{\lambda_{12}}{s_1} (1 - e^{-s_1(t_1-t_2)}) + \frac{\lambda_{12}}{s_1 + s_2} (e^{-s_1(t_1-t_2)} - e^{-s_1 t_1 - s_2 t_2})\right).
 \tag{a_2 11}$$

A₃ - Forma Geral da EBVE e Três Casos Especiais

Neste apêndice mostramos a forma mais geral da distribuição EBVE, derivada a partir da função de risco $h_i(t) = d_i N_i(t) + s_i N_{12}(t)$ juntamente com três casos especiais.

A função de sobrevivência conjunta de (T_1, T_2) condicional aos processos de Poisson é dada por:

$$\begin{aligned}
 S_{T_1, T_2 | N_1, N_2, N_{12}}(t_1, t_2) &= P(T_1 > t_1, T_2 > t_2 | N_1, N_2, N_{12}) \\
 &= \exp\left(-d_1 \int_0^{t_1} N_1(u) du\right) \exp\left(-d_2 \int_0^{t_2} N_{12}(u) du\right) \exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du\right),
 \end{aligned}
 \tag{a_3 1}$$

considerando a independência entre N_1, N_2 e N_{12} .

A função de sobrevivência conjunta é obtida de forma similar ao caso em que $d_1 = d_2 = \infty$, veja Apêndice A₂, pois temos,

$$\begin{aligned}
 S_{T_1, T_2 | N_1, N_2, N_{12}}(t_1, t_2) &= E(S_{T_1, T_2 | N_1, N_2, N_{12}}(t_1, t_2)) \\
 &= E \exp\left(-d_1 \int_0^{t_1} N_1(u) du\right) E \exp\left(-d_2 \int_0^{t_2} N_{12}(u) du\right) E \exp\left(-s_1 \int_0^{t_1} N_{12}(u) du - s_2 \int_0^{t_2} N_{12}(u) du\right),
 \end{aligned}
 \tag{a_3 2}$$

portanto,

$$S_{T_1, T_2}(t_1, t_2) = \begin{cases} \exp\left(-(\lambda_1 + \lambda_{12})t_1 - \lambda_2 t_2 + \frac{\lambda_1}{d_1}(1 - e^{-d_1 t_1}) + \frac{\lambda_2}{d_2}(1 - e^{-d_2 t_2})\right) \\ \times \exp\left(\frac{\lambda_{12}}{s_1}(1 - e^{-s_1(t_1 - t_2)}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_1(t_1 - t_2)} - e^{-s_1 t_1 - s_2 t_2})\right) & \text{se } t_1 > t_2 \\ \exp\left(-\lambda_1 t_1 - (\lambda_2 + \lambda_{12})t_2 + \frac{\lambda_1}{d_1}(1 - e^{-d_1 t_1}) + \frac{\lambda_2}{d_2}(1 - e^{-d_2 t_2})\right) \\ \times \exp\left(\frac{\lambda_{12}}{s_2}(1 - e^{-s_2(t_2 - t_1)}) + \frac{\lambda_{12}}{s_1 + s_2}(e^{-s_2(t_2 - t_1)} - e^{-s_1 t_1 - s_2 t_2})\right) & \text{se } t_1 \leq t_2. \end{cases} \quad (a_3 3)$$

Os três casos especiais da EBVE são citados a seguir:

1º caso: $d_1 = d_2 = \infty$ e corresponde a distribuição absolutamente contínua derivada no Capítulo I.

2º caso: $s_1 = s_2 = \infty$ e portanto somente os choques que ocorrem nas partes específicas dos componentes são acumulativos. A distribuição resultante não é absolutamente contínua e é dada por:

$$S_{T_1, T_2}(t_1, t_2) = \begin{cases} \exp\left(-(\lambda_1 + \lambda_{12})t_1 - \lambda_2 t_2 + \frac{\lambda_1}{d_1}(1 - e^{-d_1 t_1}) + \frac{\lambda_2}{d_2}(1 - e^{-d_2 t_2})\right) & \text{se } t_1 > t_2 \\ \exp\left(-\lambda_1 t_1 - (\lambda_2 + \lambda_{12})t_2 + \frac{\lambda_1}{d_1}(1 - e^{-d_1 t_1}) + \frac{\lambda_2}{d_2}(1 - e^{-d_2 t_2})\right) & \text{se } t_1 \leq t_2. \end{cases} \quad (a_3 4)$$

3º caso: $d_1 = d_2 = s_1 = s_2 = \infty$ e a distribuição reduz-se a BVE de Marshall e Olkin.

Apêndice B

Programa Computacional Implementado para a Geração de Dados
da Distribuição EBVE pelo Método da Rejeição

```

/* Com este programa obtivemos os resultados da Tabela 2.2*/

#include <oxstd.h>
#import <oxprob>
#import <maximize>

decl n=100; // tamanho das amostras

decl i,j,k,t1q,t1,t2,tmin,b,h,tempo1,tempo2;

/*-----densidade bivariada: t1<=t2-----*/
dbt1met2 (t1,t2,vP)
{
  decl d;
  d=(exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[0][0]+vP[2][0])*exp(-
vP[4][0]*(t2-t1))-vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[4][0]*exp(-
vP[4][0]*(t2-t1))+vP[3][0])*exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[1][0]+vP[2][0]-vP[2][0])*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
t1))-exp(-vP[3][0]*t1-vP[4][0]*t2)))+(exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[4][0]*(t2-t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[4][0]*vP[2][0])*exp(-
vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0])*exp(-
vP[4][0]*(t2-t1))+vP[3][0])*exp(-vP[3][0]*t1-vP[4][0]*t2));
  return d;
}
/*-----fim densidade bivariada: t1<=t2-----*/

```

```

/*-----densidade bivariada: t1>t2-----*/
dbt1mat2(t1,t2,vP)
{
  decl d;
  d=(exp(-vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-t2))-
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[1][0]+vP[2][0])*exp(-
vP[3][0]*(t1-t2))-vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0])*exp(-
vP[3][0]*(t1-t2))+vP[4][0])*exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]-vP[2][0])*exp(-vP[3][0]*(t1-
t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-
t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)))+(exp(-
(vP[0][0]+vP[2][0])*t1-vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-exp(-
vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[3][0]*vP[2][0])*exp(-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0])*exp(-
vP[3][0]*(t1-t2))+vP[4][0])*exp(-vP[3][0]*t1-vP[4][0]*t2));
  return d;
}
/*-----fim densidade bivariada: t1>t2-----*/

/*-----geração de (t1,t2)-----*/
gera()
{
  decl vP=<0.1;0.1;0.2;0.5;0.5>; //parametros utilizados
  decl u1,u2,u3,ber,d1,t1aux1,t2aux1;
  decl tempo1=zeros(n,1);
  decl tempo2=zeros(n,1);
  decl tempominimo=zeros(n,1);
  j=0;
  while(j<n)
  {
    u1=ranu(1,1)*35; // uniforme (0, 35)
    u2=ranu(1,1)*35;
    u3=ranu(1,1)*0.046;
    ber=ranbinomial(1,1,1,0.5); // bernoulli (0,5)

    if(ber==0)
    {
      t1aux1=min(u1,u2);
      t2aux1=max(u1,u2);
      d1=dbt1met2(t1aux1,t2aux1,vP);
      if (u3<d1)
      {
        tempo1[j][0]=t1aux1;

```

```

tempo2[j][0]=t2aux1;
tempominimo[j][0]=min(t1aux1,t2aux1);
j=j+1;
}
else
{
t1aux1=max(u1,u2);
t2aux1=min(u1,u2);
d1=dbt1mat2(t1aux1,t2aux1,vP);
if (u3<d1)
{
tempo1[j][0]=t1aux1;
tempo2[j][0]=t2aux1;
tempominimo[j][0]=min(t1aux1,t2aux1);
j=j+1;
}
}
return tempo1~tempo2~tempominimo;
}
/*-----fim geração de (t1,t2)-----*/

main()
{
h=500; //número de amostras geradas

/*-----declaração das variáveis utilizadas-----*/
decl time;
time=timer();
decl vetorminimo=zeros(h,1);
decl vetormediast1=zeros(h,1);
decl vetormediast2=zeros(h,1);
decl vetormediasminimo=zeros(h,1);
decl vetorvart1=zeros(h,1);
decl vetorvart2=zeros(h,1);
decl vetorcorrelacao=zeros(h,1);
decl mediat1,mediat2,mediaminimo,vart1,vart2;
decl correlacao=zeros(2,2);
decl mediageralt1,mediageralt2,mediageralminimo,
mediavart1,mediavart2,mediacorr;
decl desvpmediast1,desvpmediast2,desvpmediasminimo,
desvpvart1,desvpvart2,desvpcorr;

// obtenção das amostras
for (b=0;b<h;++b)
{
t1g=gera();
t1=t1g*<1;0;0>;
t2=t1g*<0;1;0>;
tmin=t1g*<0;0;1>;

//médias, variâncias e correlação dos tempos em cada amostra
mediat1=meanc(t1);
mediat2=meanc(t2);
mediaminimo=meanc(tmin);
vart1=varc(t1);
vart2=varc(t2);
correlacao=correlation(t1g);

vetormediast1[b][0]=mediat1;
vetormediast2[b][0]=mediat2;
vetormediasminimo[b][0]=mediaminimo;
vetorvart1[b][0]=vart1;
vetorvart2[b][0]=vart2;
vetorcorrelacao[b][0]=correlacao{1}[0];

//médias das 500 amostras
mediageralt1=meanc(vetormediast1);
mediageralt2=meanc(vetormediast2);
mediageralminimo=meanc(vetormediasminimo);
mediavart1=meanc(vetorvart1);
mediavart2=meanc(vetorvart2);
mediacorr=meanc(vetorcorrelacao);

//desvios padrões das 500 amostras
desvpmediast1=sqrt(varc(vetormediast1));
desvpmediast2=sqrt(varc(vetormediast2));
desvpmediasminimo=sqrt(varc(vetormediasminimo));
desvpvart1=sqrt(varc(vetorvart1));
desvpvart2=sqrt(varc(vetorvart2));
desvpcorr=sqrt(varc(vetorcorrelacao));
}

// SAÍDA
print(vetormediast1~vetormediast2~vetormediasminimo~
vetorvart1~vetorvart2~ vetorcorrelacao);

print(mediageralt1~mediageralt2~mediageralminimo~
mediavart1~mediavart2~ mediacorr );

print (desvpmediast1~ desvpmediast2~desvpmediasminimo~
desvpvart1~ desvpvart2~ desvpcorr);

print ("\n tempo de execucao: ", timespan(time));
}

```

Apêndice C

Programa Computacional Implementado para Estimação dos Parâmetros da Distribuição EBVE considerando Dados sem Censura

```

/* Com este programa obtivemos os resultados das Tabelas 2.6,
2.7, 2.8, 2,10 */

#include <oxstd.h>
#import <maximize>
#import <oxprob>

decl n=200; // tamanho das amostras
decl b,h,i,j,k,t1,t2;

/*-----densidade bivariada: t1<=t2 (veja Apêndice B) -----*/
/*-----densidade bivariada: t1>t2 (veja Apêndice B)-----*/

/*-----log densidade bivariada: t1<t2-----*/
logdbt1met2(t1,t2,vP)
{
  decl d;
  decl logd;
  d=dbt1met2(t1,t2,vP);
  logd=log(d);
  return logd;
}
/*-----fim log densidade bivariada: t1<t2-----*/

/*-----log densidade bivariada: t1>t2-----*/
logdbt1mat2(t1,t2,vP)
{
  decl d;
  decl logd;
  d=dbt1mat2(t1,t2,vP);
  logd=log(d);
  return logd;
}

```

```
/*-----fim log densidade bivariada: t1>t2-----*/
```

```

/*-----derivada em relação a lambda1 da densidade
bivariada (t1<=t2)-----*/
derivf1t1met2(t1,t2,vP)
{
  decl s1,s2,s3,s4,s5,t0;
  s2=-t1*exp(-vP[0][0]*t1-(vP[1][0]+vP[2][0])
  *t2+vP[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
  t1))+vP[2][0]/(vP[3][0]
  +vP[4][0])*exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
  vP[4][0]*t2))
  *(vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
  vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
  vP[3][0]*t1-
  vP[4][0]*t2)))+(vP[1][0]+vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
  t1))+
  vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
  t1))-exp(
  -vP[3][0]*t1-vP[4][0]*t2)));
  s4=exp(-vP[0][0]*t1-
  (vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
  exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
  vP[4][0]
  *(t2-t1))-exp(-vP[3][0]*t1-
  vP[4][0]*t2)))+(vP[1][0]+vP[2][0]-vP[2][0]*
  exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
  s5=-t1*exp(-vP[0][0]*t1-
  (vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]
  *(1-exp(-vP[4][0]*(t2-
  t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
  vP[4][0]*t2)))+(vP[4][0]*vP[2][0]*
  exp(-vP[4][0]*(t2-t1))-
  vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
  vP[4][0]*t2)
  ));
  s3=s4+s5;
  s1=s2+s3;
  s4=exp(-vP[0][0]*t1-
  (vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
  exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
  vP[4][0]

```



```

]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[4][0]*exp(-vP[
4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[1][0]+
vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][
0]+vP[4][0]))*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)
));
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[4][0]
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[4][0]*vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[4][0]
*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*-----fim derivada em relação a lambda da densidade
bivariada (t1<=t2)-----*/

/*-----derivada em relação a lambda da densidade
bivariada (t1>t2)-----*/
derivf11t1mat2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,t0;
s2=-t1*exp(-(vP[0][0]+vP[2][0])*t1-vP[1][0]
*t2+vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2))+vP[2][0]/(vP[3][0]
]+vP[4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)
)*{vP[1][0]+vP[2][0]*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[
4][0])*{vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]-vP[2][0]*exp(-vP[3][0]*(t1-
t2))+
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[3][0]*(t1-
t2))-exp(

```

```

-vP[3][0]*t1-vP[4][0]*t2)););
s4=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
]*{t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[1][0]+vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP[
3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-vP[4][0]*t2)););
s5=-t1*exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]
*(1-exp(-vP[3][0]*(t1-
t2))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[
3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[3][0]*vP[2][0]*
exp(-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3]
][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)
)););
s3=s4+s5;
s1=s2+s3;
s4=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
]*{t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[1][0]+vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP[
3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)
)*{vP[0][0]+
vP[2][0]-vP[2][0]*exp(-vP[3][0]*(t1-
t2))+vP[3][0]*vP[2][0]/(vP[3][
0]+vP[4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)
)););
s5=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
]*{t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[3][0]*vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]
*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)););

```

```

s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*-----fim derivada em relação a lambda1 da densidade
bivariada (t1>t2)-----*/

/*-----derivada em relação a lambda2 da densidade
bivariada (t1<=t2)-----*/
derivf12t1met2(t1,t2,vP)
{
  decl s1,s2,s3,s4,s5,t0;
  s2=-t2*exp(-vP[0][0]*t1-(vP[1][0]+vP[2][0])
*t2+vP[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1))+vP[2][0]/(vP[3][0]
)+vP[4][0]*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))
)* (vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0]* (vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)))+(vP[1][0]+vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
t1))-exp(
-vP[3][0]*t1-vP[4][0]*t2)));
s4=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0]
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)));
s5=-t2*exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]
*(1-exp(-vP[4][0]*(t2-
t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))+(vP[1][0]+vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)
));
}

```

```

s3=s4+s5;
s1=s2+s3;
s4=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0]
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)))+(vP[1][0]+
vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][0]
)+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)
));
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0]
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[4][0]*vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]
*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*-----fim derivada em relação a lambda2 da densidade
bivariada (t1<=t2)-----*/

/*-----derivada em relação a lambda2 da densidade
bivariada (t1>t2)-----*/
derivf12t1mat2(t1,t2,vP)
{
  decl s1,s2,s3,s4,s5,t0;
  s2=-t2*exp(-(vP[0][0]+vP[2][0])*t1-vP[1][0]
*t2+vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2))+vP[2][0]/vP[3][0]
)+vP[4][0]*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))
)* (vP[1][0]+vP[2][0]*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0])
));
}

```

```

4] [0]} * (vP[3] [0] * exp(-vP[3] [0] * (t1-t2)) + vP[4] [0] * exp(-
vP[3] [0] * t1-
vP[4] [0] * t2)) * (vP[0] [0] + vP[2] [0] - vP[2] [0] * exp(-vP[3] [0] * (t1-
t2)) +
vP[3] [0] * vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-vP[3] [0] * (t1-
t2)) - exp(
-vP[3] [0] * t1 - vP[4] [0] * t2)));
s4=exp(-(vP[0] [0] + vP[2] [0]) * t1 -
vP[1] [0] * t2 + vP[2] [0] / vP[3] [0] * (1-
exp(-vP[3] [0] * (t1-t2))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-
vP[3] [0]
] * (t1-t2)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)) * (vP[0] [0] + vP[2] [0] - vP[2]
] [0] * exp(-vP[3] [0] * (t1-
t2)) + vP[3] [0] * vP[2] [0] / (vP[3] [0] + vP[4] [0]) *
(exp(-vP[3] [0] * (t1-t2)) - exp(-vP[3] [0] * t1 - vP[4] [0] * t2)));
s5=-t2 * exp(-(vP[0] [0] + vP[2] [0]) * t1 -
vP[1] [0] * t2 + vP[2] [0] / vP[3] [0]
* (1 - exp(-vP[3] [0] * (t1-
t2))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-vP[
3] [0] * (t1-t2)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2))) * (vP[3] [0] * vP[2] [0] *
exp(-vP[3] [0] * (t1-t2)) -
vP[3] [0] * vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (vP[3]
] [0] * exp(-vP[3] [0] * (t1-t2)) + vP[4] [0] * exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)
));
s3=s4+s5;
s1=s2+s3;
s4=exp(-(vP[0] [0] + vP[2] [0]) * t1 -
vP[1] [0] * t2 + vP[2] [0] / vP[3] [0] * (1-
exp(-vP[3] [0] * (t1-t2))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-
vP[3] [0]
] * (t1-t2)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)) * (vP[1] [0] + vP[2] [0] * exp(
-vP[3] [0] * (t1-t2)) -
vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (vP[3] [0] * exp(-vP[
3] [0] * (t1-t2)) + vP[4] [0] * exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)) * (vP[0] [0] +
vP[2] [0] - vP[2] [0] * exp(-vP[3] [0] * (t1-
t2)) + vP[3] [0] * vP[2] [0] / (vP[3] [
0] + vP[4] [0]) * (exp(-vP[3] [0] * (t1-t2)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)
));
s5=exp(-(vP[0] [0] + vP[2] [0]) * t1 -
vP[1] [0] * t2 + vP[2] [0] / vP[3] [0] * (1-
exp(-vP[3] [0] * (t1-t2))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-
vP[3] [0]

```

```

] * (t1-t2)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)) * (vP[3] [0] * vP[2] [0] * exp(
-vP[3] [0] * (t1-t2)) -
vP[3] [0] * vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (vP[3] [0]
* exp(-vP[3] [0] * (t1-t2)) + vP[4] [0] * exp(-vP[3] [0] * t1 -
vP[4] [0] * t2)));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*-----fim derivada em relação a lambda2 da densidade
bivariada (t1>t2)-----*/

/*-----derivada em relação a lambda12 da densidade
bivariada (t1<=t2)-----*/
derivf112t1met2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,s6,s7,t0;
s4=(-t2+1/vP[4] [0] * (1-exp(-vP[4] [0] * (t2-
t1))) + 1 / (vP[3] [0] + vP[4] [
0]) * (exp(-vP[4] [0] * (t2-t1)) - exp(-vP[3] [0] * t1 -
vP[4] [0] * t2))) * exp(-vP
[0] [0] * t1 - (vP[1] [0] + vP[2] [0]) * t2 + vP[2] [0] / vP[4] [0] * (1-exp(-
vP[4] [
0] * (t2-t1))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-vP[4] [0] * (t2-
t1)) -
exp(-vP[3] [0] * t1 - vP[4] [0] * t2)));
s5=(vP[0] [0] + vP[2] [0] * exp(-vP[4] [0] * (t2-t1)) -
vP[2] [0] / (vP[3] [0] +
vP[4] [0]) * (vP[4] [0] * exp(-vP[4] [0] * (t2-t1)) + vP[3] [0] * exp(-
vP[3] [0] *
t1 - vP[4] [0] * t2)) * (vP[1] [0] + vP[2] [0] - vP[2] [0] * exp(-
vP[4] [0] * (t2-t1
)) + vP[4] [0] * vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-vP[4] [0] * (t2-
t1)) - exp(-vP[3] [0] * t1 - vP[4] [0] * t2)));
s3=s4*s5;
s4=exp(-vP[0] [0] * t1 -
(vP[1] [0] + vP[2] [0]) * t2 + vP[2] [0] / vP[4] [0] * (1-
exp(-vP[4] [0] * (t2-t1))) + vP[2] [0] / (vP[3] [0] + vP[4] [0]) * (exp(-
vP[4] [0]
] * (t2-t1)) - exp(-vP[3] [0] * t1 - vP[4] [0] * t2)) * (exp(-
vP[4] [0] * (t2-t1))
- 1 / (vP[3] [0] + vP[4] [0]) * (vP[4] [0] * exp(-vP[4] [0] * (t2-
t1)) + vP[3] [0] *
exp(-vP[3] [0] * t1 - vP[4] [0] * t2)) * (vP[1] [0] + vP[2] [0] -
vP[2] [0] * exp(-vP

```

```
[4][0]*(t2-t1))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][
0]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s2=s3+s4;
s3=s2+exp(-vP[0][0]*t1-(vP[1][0]+vP[2][0])*
t2+vP[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]
+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))
*(vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4
][0])*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0]*t1-vP[4][0]*t2)))*(1-exp(-vP[4][0]*(t2-
t1))+vP[4][0]/(vP[3][0]+vP[4][0
]))*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s4=s3;
s6=(-t2+1/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+1/(vP[3][0]+vP[4][
0]))*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*exp(-vP
[0][0]*t1-(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-exp(-
vP[4][
0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[4][0]*vP[2][0]*exp(-
vP[4][0]*
t2-t1))-vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-
vP[4][
0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-vP[4][0]*t2));
s7=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[4][0]*exp(-
vP[4][0]
*(t2-t1))-vP[4][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-
vP[4][0]*t2
-t1))+vP[3][0]*exp(-vP[3][0]*t1-vP[4][0]*t2));
s5=s6+s7;
s1=s4+s5;
s4=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+vP[2][0]*exp(-
vP[4][0]*
(t2-t1))-
vP[4][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[
```

```
4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[1][0]+
vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][
0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2
));
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[4][0]*vP[2][0]*exp(-
vP[4][0]*
(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]
*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0;
}
/*-----fim derivada em relação a lambda12 da densidade
bivariada (t1<=t2)-----*/

/*-----derivada em relação a lambda12 da densidade
bivariada (t1>t2)-----*/
derivf112t1mat2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,s6,s7,t0;
s4=(-t1+1/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2)))+1/(vP[3][0]+vP[4][
0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*exp(-
vP[0][0]+vP[2][0])*t1-vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-exp(-
vP[3][0]
*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-
t2))-
exp(-vP[3][0]*t1-vP[4][0]*t2));
s5=(vP[1][0]+vP[2][0]*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/vP[3][0]+
vP[4][0])*(vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2)))*(vP[0][0]+vP[2][0]-vP[2][0]*exp(-
vP[3][0]*(t1-t2
))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-
t2))-
```



```

exp(-vP[3][0]*t1-vP[4][0]*t2));
s3=s4*s5;
s4=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
]* (t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)))*(exp(-
vP[3][0]*(t1-t2))
-1/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP[3][0]*(t1-
t2))+vP[4][0]*
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[0][0]+vP[2][0]-
vP[2][0]*exp(-vP
[3][0]*(t1-t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
0)*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s2=s3+s4;
s3=s2+exp(-(vP[0][0]+vP[2][0])*t1-vP[1][0]*
t2+vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2))+vP[2][0]/(vP[3][0]
+vP[4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))
*(vP[1][0]+vP[2][0]*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4]
[0]))*(vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]*t1-vP[4][0]*t2)))*(1-exp(-vP[3][0]*(t1-
t2))+vP[3][0]/(vP[3][0]+vP[4][0]
))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s4=s3;
s6=(-t1+1/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2)))+1/(vP[3][0]+vP[4][0]
0))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*exp(-
vP[0][0]+vP[2][0])*t1-vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-exp(-
vP[3][0]
0)*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[3][0]*(t1-
t2))-
exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[3][0]*vP[2][0]*exp(-
vP[3][0]*{
t1-t2}-vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-
vP[3][0]
0)*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-vP[4][0]*t2));
s7=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
]* (t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)))*(vP[3][0]*exp(-
vP[3][0]
*(t1-t2))-vP[3][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-
vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]/(vP[3][0]+vP[4][0])*t1*exp(-

```

```

/*-----fim derivada em relação a lambda12 da densidade
bivariada (t1>t2)-----*/

```

```

/*---derivada em relação a s1 da densidade bivariada (t1<=t2)*
derivfs1t1met2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,s6,s7,t0;
s4=(-vP[2][0]/(vP[3][0]+vP[4][0]))^2*(exp(-vP[4][0]*(t2-t1))-
exp
(-vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]/(vP[3][0]+vP[4][0])*t1*exp(-

```

```

vP[3][0]*t1-vP[4][0]*t2))*exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP
[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]+vP[
4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)));
s5=(vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/vP[3][0]+
vP[4][0])*vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2))*vP[1][0]+vP[2][0]-vP[2][0]*exp(-
vP[4][0]*(t2-t1
))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2)));
s3=s4*s5;
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
])*t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
s6=(vP[2][0]/(vP[3][0]+vP[4][0])^2*(vP[4][0]*exp(-
vP[4][0]*(t2-
t1))+vP[3][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))-
vP[2][0]/vP[3][0]+vP
[4][0])*exp(-vP[3][0]*t1-vP[4][0]*t2)-vP[3][0]*t1*exp(-
vP[3][0]*t1-vP[4][0]*t2))*vP[1][0]+vP[2][0]-vP[2][0]*exp(-
vP[4][0]*(t2-t1
))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[4][0]*(t2-
t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2)));
s4=s5*s6;
s2=s3+s4;
s4=s2;
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
])*t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[0][0]+vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*vP[4][0]*exp(-vP[
4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))*(-
vP[4][0]
*vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[4][0]*(t2-t1))-exp(-
vP[3
][0]*t1-
vP[4][0]*t2))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*t1*exp

```

```

(-vP[3][0]*t1-vP[4][0]*t2));
s3=s4+s5;
s4=s3;
s6=(-vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[4][0]*(t2-t1))-
exp
(-vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]/(vP[3][0]+vP[4][0])*t1*exp(-
vP[3][0]*t1-vP[4][0]*t2))*exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP
[2][0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]+vP[
4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP
[4][0]*vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]
+vP[4][0])*vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0]
*t1-vP[4][0]*t2)));
s7=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
])*t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[4][0]*vP[2][0]/(vP[
3][0]+vP[4][0])^2*(vP[4][0]*exp(-vP[4][0]*(t2-
t1))+vP[3][0]*exp(-
vP[3][0]*t1-vP[4][0]*t2))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*t1-
vP[4][0]*t2)-vP[3][0]*t1*exp(-vP[3][0]*t1-vP[4][0]
*t2)));
s5=s6+s7;
s1=s4+s5;
s4=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[4][0
])*t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[0][0]+vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*vP[4][0]*exp(-vP[
4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[1][0]+
vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][
0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2
)));

```

```

s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0]
)*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[4][0]*vP[2][0]*exp(
-vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*vP[4][0]
*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*-----fim derivada em relação a s1 da densidade bivariada
(t1<=t2)-----*/

/*---derivada em relação a s1 da densidade bivariada (t1>t2)---*/
derivs1t1mat2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,s6,s7,s8,s9,t0;
s4=(-vP[2][0]/vP[3][0]^2*(1-exp(-vP[3][0]*(t1-t2)))-
vP[2][0]/vP
[3][0]*(t2-t1)*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0])
^2*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]
]/(vP[3][0]+vP[4][0])*(t2-t1)*exp(-vP[3][0]*(t1-
t2))+t1*exp(-vP[3]
)[0]*t1-vP[4][0]*t2))*exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[
2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
t2)))+vP[2][0]/(vP[3][0]+vP[4]
[0])*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
s5=(vP[1][0]+vP[2][0])*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+
vP[4][0])*vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2))*vP[0][0]+vP[2][0]-vP[2][0]*exp(-
vP[3][0]*(t1-t2
))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-
t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
s3=s4*s5;
s5=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-

```

```

exp(-vP[3][0]*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]
)*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s6=(vP[2][0]*(t2-t1)*exp(-vP[3][0]*(t1-
t2))+vP[2][0]/(vP[3][0]+
vP[4][0])^2*(vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]
)*t1-vP[4][0]*t2))-vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]*t1
-t2))+vP[3][0]*(t2-t1)*exp(-vP[3][0]*(t1-t2))-
vP[4][0]*t1*exp(-vP[
3][0]*t1-vP[4][0]*t2))*vP[0][0]+vP[2][0]-vP[2][0]*exp(-
vP[3][0]*
(t1-t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]*(t1-
t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
s4=s5*s6;
s2=s3+s4;
s4=s2;
s6=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]
)*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2)));
s7=(vP[1][0]+vP[2][0])*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+
vP[4][0])*vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2))*vP[0][0]+vP[2][0]-vP[2][0]*exp(-
vP[3][0]*(t1-
t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-
t2))-exp(-
vP[3][0]*t1-vP[4][0]*t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-
vP[3][0]
)*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*(t1-t2))-exp(-
vP[3][0]*t1-
vP[4][0]*t2)));
s5=s6*s7;
s3=s4+s5;
s4=s3;
s7=-vP[2][0]/vP[3][0]^2*(1-exp(-vP[3][0]*(t1-t2)))-
vP[2][0]/vP
[3][0]*(t2-t1)*exp(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[3][0]*(t1-t2))-exp(-
vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]

```

```

/(vP[3][0]+vP[4][0])*((t2-t1)*exp(-vP[3][0]*(t1-t2))+t1*exp(-
vP[3][0]*t1-vP[4][0]*t2));
s8=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]*
(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))* (vP[3][0]*vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]
*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s6=s7*s8;
s8=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s9=vP[2][0]*exp(-vP[3][0]*(t1-t2))+vP[3][0]*vP[2][0]*(t2-
t1)*exp
(-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP
[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[3][0]^2*(vP[3][0]
*exp(-vP[3][0]*(t1-t2))+vP
[4][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+
vP[4][0]))*(exp(-vP[3][0]*(t1-t2))+vP[3][0]*(t2-t1)*exp(-
vP[3][0]*(
t1-t2))-vP[4][0]*t1*exp(-vP[3][0]*t1-vP[4][0]*t2));
s7=s8*s9;
s5=s6+s7;
s1=s4+s5;
s4=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))* (vP[1][0]+vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP[3]
[0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))* (vP[0][0]+
vP[2][0]-vP[2][0]*exp(-vP[3][0]*(t1-
t2))+vP[3][0]*vP[2][0]/(vP[3][0]
[0]+vP[4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)

```

```

));
s5=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
vP[3][0]
(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2))* (vP[3][0]*vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]
*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*fim derivada em relação a s1 da densidade bivariada (t1>t2)*/

/*---derivada em relação a s2 da densidade bivariada (t1<=t2)*/
derivfs2t1met2(t1,t2,vP)
{
decl s1,s2,s3,s4,s5,s6,s7,s8,s9,t0;
s4=(-vP[2][0]/vP[4][0]^2*(1-exp(-vP[4][0]*(t2-t1)))-
vP[2][0]/vP
[4][0]*(t1-t2)*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])
^2*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]
]/(vP[3][0]+vP[4][0]))*(t1-t2)*exp(-vP[4][0]*(t2-
t1))+t2*exp(-vP[3]
[0]*t1-vP[4][0]*t2))*exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2]
[0]/vP[4][0]*(1-exp(-vP[4][0]*(t2-
t1)))+vP[2][0]/(vP[3][0]+vP[4]
[0]))*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s5=(vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+
vP[4][0]))*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2))* (vP[1][0]+vP[2][0]-vP[2][0]*exp(-
vP[4][0]*(t2-t1)
))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[4][0]*(t2-
t1))-
exp(-vP[3][0]*t1-vP[4][0]*t2));
s3=s4*s5;
s5=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-

```



```

exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s6=(vP[2][0]*(t1-t2)*exp(-vP[4][0]*(t2-
t1))+vP[2][0]/(vP[3][0]+
vP[4][0])^2*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0
]*t1-vP[4][0]*t2))-vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0]*(t2
-t1))+vP[4][0]*(t1-t2)*exp(-vP[4][0]*(t2-t1))-
vP[3][0]*t2*exp(-vP[
3][0]*t1-vP[4][0]*t2))*(vP[1][0]+vP[2][0]-vP[2][0]*exp(-
vP[4][0]*
(t2-t1))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0]*(t2-
t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s4=s5*s6;
s2=s3+s4;
s4=s2;
s6=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s7=(vP[0][0]+vP[2][0]*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+
vP[4][0])*(vP[4][0]*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-
vP[3][0]*
t1-vP[4][0]*t2))*(vP[2][0]*(t1-t2)*exp(-vP[4][0]*(t2-
t1))+vP[2][
0]/(vP[3][0]+vP[4][0])*exp(-vP[4][0]*(t2-t1))-exp(-
vP[3][0]*t1-vP
[4][0]*t2))-vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-
vP[4][0
]*t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[4][0]*vP[2][0]/(vP[3][
0]+vP[4][0])*((t1-t2)*exp(-vP[4][0]*(t2-t1))+t2*exp(-
vP[3][0]*t1-
vP[4][0]*t2));
s5=s6*s7;
s3=s4+s5;
s4=s3;
s7=-vP[2][0]/vP[4][0]^2*(1-exp(-vP[4][0]*(t2-t1))-
vP[2][0]/vP[
4][0]*(t1-t2)*exp(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])^
2*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[2][0]

```

```

/(vP[3][0]+vP[4][0])*((t1-t2)*exp(-vP[4][0]*(t2-t1))+t2*exp(-
vP[3][
0]*t1-vP[4][0]*t2));
s8=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[4][0]*vP[2][0]*exp(-
vP[4][0]*(t2-t1))-
vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]
*exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2));
s6=s7*s8;
s8=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-vP[4][0]*t2));
s9=vP[2][0]*exp(-vP[4][0]*(t2-t1))+vP[4][0]*vP[2][0]*(t1-
t2)*exp
(-vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP
[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))+vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0])^2*(vP[4][
0]*exp(-vP[4][0]*(t2-t1))+vP
[3][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))-
vP[4][0]*vP[2][0]/(vP[3][0]+
vP[4][0])*(exp(-vP[4][0]*(t2-t1))+vP[4][0]*(t1-t2)*exp(-
vP[4][0]*(
t2-t1))-vP[3][0]*t2*exp(-vP[3][0]*t1-vP[4][0]*t2));
s7=s8*s9;
s5=s6*s7;
s1=s4+s5;
s4=exp(-vP[0][0]*t1-
(vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
exp(-vP[4][0]*(t2-t1))+vP[2][0]/(vP[3][0]+vP[4][0])*exp(-
vP[4][0
]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[0][0]+vP[2][0]*exp(-
vP[4][0]*(t2-t1))-
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[4][0]*exp(-vP[
4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2))*vP[1][0]+
vP[2][0]-vP[2][0]*exp(-vP[4][0]*(t2-
t1))+vP[4][0]*vP[2][0]/(vP[3][
0]+vP[4][0])*(exp(-vP[4][0]*(t2-t1))-exp(-vP[3][0]*t1-
vP[4][0]*t2)

```

```

    );
    s5=exp(-vP[0][0]*t1-
    (vP[1][0]+vP[2][0])*t2+vP[2][0]/vP[4][0]*(1-
    exp(-vP[4][0]*(t2-t1)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
    vP[4][0]
    )*(t2-t1))-exp(-vP[3][0]*t1-
    vP[4][0]*t2))* (vP[4][0]*vP[2][0]*exp(
    -vP[4][0]*(t2-t1))-
    vP[4][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[4][0]
    *exp(-vP[4][0]*(t2-t1))+vP[3][0]*exp(-vP[3][0]*t1-
    vP[4][0]*t2));
    s3=s4+s5;
    s2=1/s3;
    t0=s1*s2;
    return(t0);
}
/* -----fim derivada em relação a s2 da densidade
bivariada (t1<=t2)-----*/

/*---derivada em relação a s2 da densidade bivariada (t1>t2)---*/
derivfs2tmat2(t1,t2,vP)
{
    decl s1,s2,s3,s4,s5,s6,s7,t0;
    s4=(-vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[3][0]*(t1-t2))-
    exp
    (-vP[3][0]*t1-
    vP[4][0]*t2))+vP[2][0]/(vP[3][0]+vP[4][0])*t2*exp(-
    vP[3][0]*t1-vP[4][0]*t2))*exp(-vP[0][0]+vP[2][0])*t1-
    vP[1][0]*t2+
    vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
    t2)))+vP[2][0]/(vP[3][0]+vP[
    4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
    vP[4][0]*t2));
    s5=(vP[1][0]+vP[2][0]*exp(-vP[3][0]*(t1-t2)))-
    vP[2][0]/(vP[3][0]+
    vP[4][0]))*(vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
    vP[3][0]*
    t1-vP[4][0]*t2))* (vP[0][0]+vP[2][0]-vP[2][0]*exp(-
    vP[3][0]*t1-t2
    ))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[3][0]*(t1-
    t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
    s3=s4*s5;
    s5=exp(-(vP[0][0]+vP[2][0])*t1-
    vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
    exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
    vP[3][0]
    )*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
    s6=(vP[2][0]/(vP[3][0]+vP[4][0])^2*(vP[3][0]*exp(-
    vP[3][0]*(t1-
    t2))+vP[4][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))-
    vP[2][0]/(vP[3][0]+vP
    [4][0]))*(exp(-vP[3][0]*t1-vP[4][0]*t2))-vP[4][0]*t2*exp(-
    vP[3][0]*t1-vP[4][0]*t2))* (vP[0][0]+vP[2][0]-vP[2][0]*exp(-
    vP[3][0]*t1-t2)
    )+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-vP[3][0]*(t1-
    t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
    s4=s5*s6;
    s2=s3+s4;
    s4=s2;
    s5=exp(-(vP[0][0]+vP[2][0])*t1-
    vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
    exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
    vP[3][0]
    )*(t1-t2))-exp(-vP[3][0]*t1-
    vP[4][0]*t2))* (vP[1][0]+vP[2][0]*exp(
    -vP[3][0]*(t1-t2))-
    vP[2][0]/(vP[3][0]+vP[4][0]))*(vP[3][0]*exp(-vP[
    3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-vP[4][0]*t2))* (-
    vP[3][0]
    *vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[3][0]*(t1-t2))-exp(-
    vP[3]
    ) [0]*t1-
    vP[4][0]*t2))+vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*t2*exp
    (-vP[3][0]*t1-vP[4][0]*t2));
    s3=s4+s5;
    s4=s3;
    s6=(-vP[2][0]/(vP[3][0]+vP[4][0])^2*(exp(-vP[3][0]*(t1-t2))-
    exp
    (-vP[3][0]*t1-
    vP[4][0]*t2))+vP[2][0]/(vP[3][0]+vP[4][0])*t2*exp(-
    vP[3][0]*t1-vP[4][0]*t2))*exp(-vP[0][0]+vP[2][0])*t1-
    vP[1][0]*t2+
    vP[2][0]/vP[3][0]*(1-exp(-vP[3][0]*(t1-
    t2)))+vP[2][0]/(vP[3][0]+vP[
    4][0]))*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
    vP[4][0]*t2))* (vP
    [3][0]*vP[2][0]*exp(-vP[3][0]*(t1-t2))-
    vP[3][0]*vP[2][0]/(vP[3][0]
    +vP[4][0]))*(vP[3][0]*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-
    vP[3][0]
    *t1-vP[4][0]*t2));
    s7=exp(-(vP[0][0]+vP[2][0])*t1-
    vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
    exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0]))*(exp(-
    vP[3][0]
    )*(t1-t2))-exp(-vP[3][0]*t1-vP[4][0]*t2));
}

```

```

]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[3][0]*vP[2][0]/(vP[
3][0]+vP[4][0])^2*(vP[3][0]*exp(-vP[3][0]*(t1-
t2))+vP[4][0]*exp(-
vP[3][0]*t1-vP[4][0]*t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-vP[3][0]*t1-
vP[4][0]*t2)-vP[4][0]*t2*exp(-vP[3][0]*t1-vP[4][0]
*t2)));
s5=s6+s7;
s1=s4+s5;
s4=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]
]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[1][0]+vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[2][0]/(vP[3][0]+vP[4][0])*(vP[3][0]*exp(-vP[
3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[0][0]+
vP[2][0]-vP[2][0]*exp(-vP[3][0]*(t1-
t2))+vP[3][0]*vP[2][0]/(vP[3][
0]+vP[4][0])*(exp(-vP[3][0]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)
));
s5=exp(-(vP[0][0]+vP[2][0])*t1-
vP[1][0]*t2+vP[2][0]/vP[3][0]*(1-
exp(-vP[3][0]*(t1-t2)))+vP[2][0]/(vP[3][0]+vP[4][0])*(exp(-
vP[3][0]
]*(t1-t2))-exp(-vP[3][0]*t1-
vP[4][0]*t2)))*(vP[3][0]*vP[2][0]*exp(
-vP[3][0]*(t1-t2))-
vP[3][0]*vP[2][0]/(vP[3][0]+vP[4][0])*(vP[3][0]
*exp(-vP[3][0]*(t1-t2))+vP[4][0]*exp(-vP[3][0]*t1-
vP[4][0]*t2)));
s3=s4+s5;
s2=1/s3;
t0=s1*s2;
return(t0);
}
/*fim derivada em relação a s2 da densidade bivariada (t1>t2)*/

/*-----derivada em relação a lambda1 do log da função de
verossimilhança-----*/
derivlogverolambda1(const vP, const u1, const avScore, const
amHess)
{
decl derivlogveros=zeros(n,1);
decl derivlogver;
for(i=0;i<n;++i)
{
if (t1[i][0]<t2[i][0])
{
derivlogveros[i][0]=derivf11t1met2(t1[i][0],t2[i][0],vP);
}
else
{
derivlogveros[i][0]=derivf11t1mat2(t1[i][0],t2[i][0],vP);
}
}
u1[0]=sumc(derivlogveros);
return 1;
}
/*-----fim derivada em relação a lambda1 do log da função de
verossimilhança-----*/

/*-----derivada em relação a lambda2 do log da função de
verossimilhança-----*/
derivlogverolambda2(const vP, const u2, const avScore, const
amHess)
{
decl derivlogveros=zeros(n,1);
decl derivlogver;
for(i=0;i<n;++i)
{
if (t1[i][0]<t2[i][0])
{
derivlogveros[i][0]=derivf12t1met2(t1[i][0],t2[i][0],vP);
}
else
{
derivlogveros[i][0]=derivf12t1mat2(t1[i][0],t2[i][0],vP);
}
}
u2[0]=sumc(derivlogveros);
return 1;
}
/*-----fim derivada em relação a lambda2 do log da função
de verossimilhança-----*/

/*-----derivada em relação a lambda12 do log da função de
verossimilhança-----*/

```

```

derivlogverolambdal2(const vP, const u3, const avScore, const
amHess)
{
  decl derivlogveros=zeros(n,1);
  decl derivlogver;
  for(i=0;i<n;++i)
  {
    if (t1[i][0]<t2[i][0])
    {
      derivlogveros[i][3]=derivf112t1met2(t1[i][0],t2[i][0],vP);
    }
    else
    {
      derivlogveros[i][0]=derivf112t1mat2(t1[i][0],t2[i][0],vP);
    }
  }
  u3[0]=sumc(derivlogveros);
  return 1;
}
/*-----fim derivada em relação a lambdal2 do log da
função de verossimilhança-----*/

/*-----derivada em relação a s1 do log da função de
verossimilhança-----*/
derivlogveros1(const vP, const u4, const avScore, const amHess)
{
  decl derivlogveros=zeros(n,1);
  decl derivlogver;
  for(i=0;i<n;++i)
  {
    if (t1[i][0]<t2[i][0])
    {
      derivlogveros[i][0]=derivfs1t1met2(t1[i][0],t2[i][0],vP);
    }
    else
    {
      derivlogveros[i][0]=derivfs1t1mat2(t1[i][0],t2[i][0],vP);
    }
  }
  u4[0]=sumc(derivlogveros);
  return 1;
}
/*-----fim derivada em relação a s1 do log da função de
verossimilhança-----*/

```

```

/*-----derivada em relação a s2 do log da função de
verossimilhança-----*/
derivlogveros2(const vP, const u5, const avScore, const amHess)
{
  decl derivlogveros=zeros(n,1);
  decl derivlogver;
  for(i=0;i<n;++i)
  {
    if (t1[i][0]<t2[i][0])
    {
      derivlogveros[i][0]=derivfs2t1met2(t1[i][0],t2[i][0],vP);
    }
    else
    {
      derivlogveros[i][0]=derivfs2t1mat2(t1[i][0],t2[i][0],vP);
    }
  }
  u5[0]=sumc(derivlogveros);
  return 1;
}
/*-----fim derivada em relação a s2 do log da função de
verossimilhança-----*/

/*-----geração de (t1,t2)-----*/
gera()
{
  decl vP=<0.1;0.1;0.2;0.5;0.5>; //parâmetros utilizados
  decl u1,u2,u3,ber,d,t1aux1,t2aux1;
  decl tempol=zeros(n,1);
  decl tempo2=zeros(n,1);
  j=0;
  while(j<n)
  {
    u1=ranu(1,1)*35;
    u2=ranu(1,1)*35;
    u3=ranu(1,1)*0.046;
    ber=ranbinomial(1,1,1,0.5);

    if(ber==0)
    {
      t1aux1=mjn(u1,u2);
      t2aux1=max(u1,u2);
      d=dbt1met2(t1aux1,t2aux1,vP);
      if (u3<d)
      {
        tempol[j][0]=t1aux1;
        tempo2[j][0]=t2aux1;
      }
    }
  }
}

```



```

        j=j+1;
    }
}
else
{
    t1aux1=max(u1,u2);
    t2aux1=min(u1,u2);
    d=dbt1mat2(t1aux1,t2aux1,vP);
    if (u3<d)
    {
        tempo1[j][0]=t1aux1;
        tempo2[j][0]=t2aux1;
        j=j+1;
    }
}
}
return tempo1-tempo2;
}
/*-----fim geração de (t1,t2)-----*/

/*-----log da função de verossimilhança-----*/
logfveros(const vP, const l, const avScore, const anHess)
{
    decl logf=zeros(n,1);
    for(i=0; i<n; ++i)
    {
        if(t1[i][0]<t2[i][0])
        {
            logf[i][0]= logdbt1met2(t1[i][0],t2[i][0],vP);
        }
        else
        {
            logf[i][0]= logdbt1mat2(t1[i][0],t2[i][0],vP);
        }
    }
    l[0]=sumc(logf); //log da função de verossimilhança
    if ( vP[0][0]<0 || vP[1][0]<0 || vP[2][0]<0 || vP[3][0]<0 ||
        vP[4][0]<0) //restrição para os estimadores
    {
        return 0;
    }
    return l;
}
/* -----fim log função de verossimilhança -----*/

main()
{
    h=500;//total de amostras
    decl e, e1, e2, e3, e4, e5, v, z;
    e=0; e1=0; e2=0; e3=0; e4=0;
    e5=0; v=0; z=0;

    // declaração das variáveis
    decl lambda1=zeros(h,1);
    decl lambda2=zeros(h,1);
    decl lambda12=zeros(h,1);
    decl s1=zeros(h,1);
    decl s2=zeros(h,1);

    decl varlambda1=zeros(h,1);
    decl varlambda2=zeros(h,1);
    decl varlambda12=zeros(h,1);
    decl vars1=zeros(h,1);
    decl vars2=zeros(h,1);

    decl desvplambda1=zeros(h,1);
    decl desvplambda2=zeros(h,1);
    decl desvplambda12=zeros(h,1);
    decl desvps1=zeros(h,1);
    decl desvps2=zeros(h,1);

    decl eqmlambda1=zeros(h,1);
    decl eqmlambda2=zeros(h,1);
    decl eqmlambda12=zeros(h,1);
    decl eqms1=zeros(h,1);
    decl eqms2=zeros(h,1);

    decl liminflambda190=zeros(h,1);
    decl limsuplambda190=zeros(h,1);
    decl liminflambda290=zeros(h,1);
    decl limsuplambda290=zeros(h,1);
    decl liminflambda1290=zeros(h,1);
    decl limsuplambda1290=zeros(h,1);
    decl liminfls190=zeros(h,1);
    decl limsupsl190=zeros(h,1);
    decl liminfls290=zeros(h,1);
    decl limsupsl290=zeros(h,1);

    decl liminflambda195=zeros(h,1);
    decl limsuplambda195=zeros(h,1);
    decl liminflambda295=zeros(h,1);
    decl limsuplambda295=zeros(h,1);
    decl liminflambda1295=zeros(h,1);
    decl limsuplambda1295=zeros(h,1);
    decl liminfls195=zeros(h,1);
}

```

```

decl l1msups195=zeros(h,1);
decl l1minfs295=zeros(h,1);
decl l1msups295=zeros(h,1);

decl MatInfObs=zeros(5,5);
decl invMatInfObs=zeros(5,5);
decl diag=zeros(5,1);
decl desvpad=zeros(5,1);
decl l,ir,mhess;
decl tlg=zeros(n,2);
decl time;
time=timer();

decl bb=zeros(5,5);
decl d1=zeros(5,5);
decl d2=zeros(5,5);
decl d3=zeros(5,5);
decl d4=zeros(5,5);
decl d5=zeros(5,5);

decl produto=zeros(25,1);
decl STD=zeros(h,1);
decl mhess1, mhess2, mhess3, mhess4, mhess5;
decl p,q,r,s,t,u;

decl quantlambda190=0;
decl quantlambda290=0;
decl quantlambda1290=0;
decl quants190=0;
decl quants290=0;
decl quantlambda195=0;
decl quantlambda295=0;
decl quantlambda1295=0;
decl quants195=0;
decl quants295=0;

decl cobertura90lambda1,
cobertura90lambda2, cobertura90lambda12,
cobertura90s1, cobertura90s2;
decl cobertura95lambda1, cobertura95lambda2,
cobertura95lambda12, cobertura95s1, cobertura95s2;

b=0;
while(b<h)//gera as amostras
{
    tlg=gera();
    t1=tlg*<1;0>;
    t2=tlg*<0;1>;

/*maximiza o log da verossimilhança*/
decl vP=<1;1;1;1>; // chutes iniciais
logfveros(vP,&i,0,0);
MaxControl(1000,1000);
mhess=0.00001*unit(5);
ir=MaxBFGS(logfveros,&vP,&l,&mhess,TRUE);
z=z+1;

if(ir==MAX_CONV)
    v=v+1;

if (ir==MAX_CONV && vP[0][0]<1)
    e1=e1+1;
if (ir==MAX_CONV && vP[1][0]<1)
    e2=e2+1;
if (ir==MAX_CONV && vP[2][0]<2)
    e3=e3+1;
if (ir==MAX_CONV && vP[3][0]<5)
    e4=e4+1;
if (ir==MAX_CONV && vP[4][0]<5)
    e5=e5+1;

if(ir==MAX_CONV && vP[0][0]<1 && vP[1][0]<1
&& vP[2][0]<2 && vP[3][0]<5 && vP[4][0]<5)
{

/*EMV*/
lambda1[b][0]=vP[0][0];
lambda2[b][0]=vP[1][0];
lambda12[b][0]=vP[2][0];
s1[b][0]=vP[3][0];
s2[b][0]=vP[4][0];

/*medidas de dispersão*/
Num2Derivative(logfveros,vP,&mhess);
MatInfObs=mhess*(-1);
invMatInfObs=invert(MatInfObs);
bb=invMatInfObs;
diag=diagonal(invMatInfObs);
e=e+1;
if (invMatInfObs[0][0]>0 && invMatInfObs[1][1]>0 &&
invMatInfObs[2][2]>0&& invMatInfObs[3][3]>0 &&
invMatInfObs[4][4] > 0)
{
    desvpad=sqrt(diag);

```

```

desvplambda1[b][0]=sqrt(bb [0][0]);
desvplambda2[b][0]=sqrt(bb [1][1]);
desvplambda12[b][0]=sqrt(bb [2][2]);
desvps1[b][0]=sqrt(bb [3][3]);
desvps2[b][0]=sqrt(bb [4][4]);

//Cálculo de STD
Num2Derivative(derivlogverolambda1, vP, &mhess1);
Num2Derivative(derivlogverolambda2, vP, &mhess2);
Num2Derivative(derivlogverolambda12, vP, &mhess3);
Num2Derivative(derivlogveros1, vP, &mhess4);
Num2Derivative(derivlogveros2, vP, &mhess5);

d1=mhess1;
d2=mhess2;
d3=mhess3;
d4=mhess4;
d5=mhess5;

produto[0][0]=0;
p=0; s=0;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[0][0]=produto[0][0]+bb[p][q]*bb[s][t]*
                bb[r][u]*d1[q][r]*d1[r][u];
            }
        }
    }
}
produto[1][0]=0;
p=0; s=1;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[1][0]=produto[1][0]+bb[p][q]*bb[s][t]*b
                b[r][u]*d1[q][r]*d2[t][u];
            }
        }
    }
}

```

```

}
}
}
}
}
produto[2][0]=0;
p=0; s=2;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[2][0]=produto[2][0]+bb[p][q]*bb[s][t]*b
                b[r][u]*d1[q][r]*d3[t][u];
            }
        }
    }
}
produto[3][0]=0;
p=0; s=3;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[3][0]=produto[3][0]+bb[p][q]*bb[s][t]*b
                b[r][u]*d1[q][r]*d4[t][u];
            }
        }
    }
}
produto[4][0]=0;
p=0; s=4;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[4][0]=produto[4][0]+bb[p][q]*bb[s][t]*b
                b[r][u]*d1[q][r]*d5[t][u];
            }
        }
    }
}

```

```

    }
  }
  produto[5][0]=0;
  p=1; s=0;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {
          produto[5][0]=produto[5][0]+bb[p][q]*bb[s][t]*b
          b[r][u]*d2[q][r]*d1[t][u];
        }
      }
    }
  }
  produto[6][0]=0;
  p=1; s=1;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {
          produto[6][0]=produto[6][0]+bb[p][q]*bb[s][t]*b
          b[r][u]*d2[q][r]*d2[t][u];
        }
      }
    }
  }
  produto[7][0]=0;
  p=1; s=2;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {

```

```

          produto[7][0]=produto[7][0]+bb[p][q]*bb[s][t]*b
          b[r][u]*d2[q][r]*d3[t][u];
        }
      }
    }
  }
  produto[8][0]=0;
  p=1; s=3;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {
          produto[8][0]=produto[8][0]+bb[p][q]*bb[s][t]*b
          b[r][u]*d2[q][r]*d4[t][u];
        }
      }
    }
  }
  produto[9][0]=0;
  p=1; s=4;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {
          produto[9][0]=produto[9][0]+bb[p][q]*bb[s][t]*b
          b[r][u]*d2[q][r]*d5[t][u];
        }
      }
    }
  }
  produto[10][0]=0;
  p=2; s=0;
  for(q=0;q<5;++q)
  {
    for(r=0;r<5;++r)
    {
      for(t=0;t<5;++t)
      {
        for(u=0;u<5;++u)
        {

```



```

        produto[10][0]=produto[10][0]+bb[p][q]*bb[s][t]
        *bb[r][u]*d3[q][r]*d1[t][u];
    }
}
}
produto[11][0]=0;
p=2; s=1;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[11][0]=produto[11][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d3[q][r]*d2[t][u];
            }
        }
    }
}
produto[12][0]=0;
p=2; s=2;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[12][0]=produto[12][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d3[q][r]*d3[t][u];
            }
        }
    }
}
produto[13][0]=0;
p=2; s=3;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {

```

```

        produto[13][0]=produto[13][0]+bb[p][q]*bb[s][t]
        *bb[r][u]*d3[q][r]*d4[t][u];
    }
}
}
}
produto[14][0]=0;
p=2; s=4;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[14][0]=produto[14][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d3[q][r]*d5[t][u];
            }
        }
    }
}
produto[15][0]=0;
p=3; s=0;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[15][0]=produto[15][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d4[q][r]*d1[t][u];
            }
        }
    }
}
produto[16][0]=0;
p=3; s=1;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {

```

```

        produto[16][0]=produto[16][0]+bb[p][q]*bb[s][t]
        *bb[r][u]*d4[q][r]*d2[t][u];
    }
}
}
produto[17][0]=0;
p=3; s=2;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[17][0]=produto[17][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d4[q][r]*d3[t][u];
            }
        }
    }
}
produto[18][0]=0;
p=3; s=3;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[18][0]=produto[18][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d4[q][r]*d4[t][u];
            }
        }
    }
}
}
produto[19][0]=0;
p=3; s=4;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {

```

```

        produto[19][0]=produto[19][0]+bb[p][q]*bb[s][t]
        *bb[r][u]*d4[q][r]*d5[t][u];
    }
}
}
}
produto[20][0]=0;
p=4; s=0;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[20][0]=produto[20][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d5[q][r]*d1[t][u];
            }
        }
    }
}
}
produto[21][0]=0;
p=4; s=1;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[21][0]=produto[21][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d5[q][r]*d2[t][u];
            }
        }
    }
}
}
produto[22][0]=0;
p=4; s=2;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {

```

```

        produto[22][0]=produto[22][0]+bb[p][q]*bb[s][t]
        *bb[r][u]*d5[q][r]*d3[t][u];
    }
}
produto[23][0]=0;
p=4; s=3;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[23][0]=produto[23][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d5[q][r]*d4[t][u];
            }
        }
    }
}
produto[24][0]=0;
p=4; s=4;
for(q=0;q<5;++q)
{
    for(r=0;r<5;++r)
    {
        for(t=0;t<5;++t)
        {
            for(u=0;u<5;++u)
            {
                produto[24][0]=produto[24][0]+bb[p][q]*bb[s][t]
                *bb[r][u]*d5[q][r]*d5[t][u];
            }
        }
    }
}
STD[b][0]=(sumc(produto));
print("\n STD:",(b+1),STD[b][0]);
b=b+1;
}
}

// erros quadráticos médios dos estimadores
for (j=0;j<h;++j)
{
    eqmlambda1[j][0]= (desvplambda1[j][0])^2+
    (lambda1[j][0]-0.1).^2;
    eqmlambda2[j][0]= (desvplambda2[j][0])^2+
    (lambda2[j][0]-0.1).^2;
    eqmlambda12[j][0]= (desvplambda12[j][0])^2+
    (lambda12[j][0]-0.2).^2;
    eqms1[j][0]= (desvps1[j][0])^2+(s1[j][0]-0.5).^2;
    eqms2[j][0]= (desvps2[j][0])^2+(s2[j][0]-0.5).^2;

    /*intervalos de confiança 90%*/
    liminflambda190[j][0]=lambda1[j][0]-
    1.6449*desvplambda1[j][0];
    limsuplambda190[j][0]=lambda1[j][0]+
    1.6449*desvplambda1[j][0];
    liminflambda290[j][0]=lambda2[j][0]-
    1.6449*desvplambda2[j][0];
    limsuplambda290[j][0]=lambda2[j][0]+
    1.6449*desvplambda2[j][0];
    liminflambda1290[j][0]=lambda12[j][0]-
    1.6449*desvplambda12[j][0];
    limsuplambda1290[j][0]=lambda12[j][0]+
    1.6449*desvplambda12[j][0];
    liminfs190[j][0]=s1[j][0]-1.6449*desvps1[j][0];
    limsups190[j][0]=s1[j][0]+1.6449*desvps1[j][0];
    liminfs290[j][0]=s2[j][0]-1.6449*desvps2[j][0];
    limsups290[j][0]=s2[j][0]+1.6449*desvps2[j][0];

    /*intervalos de confiança 95%*/
    liminflambda195[j][0]=lambda1[j][0]-
    1.96*desvplambda1[j][0];
    limsuplambda195[j][0]=lambda1[j][0]+
    1.96*desvplambda1[j][0];
    liminflambda295[j][0]=lambda2[j][0]-
    1.96*desvplambda2[j][0];
    limsuplambda295[j][0]=lambda2[j][0]+
    1.96*desvplambda2[j][0];
    liminflambda1295[j][0]=lambda12[j][0]-
    1.96*desvplambda12[j][0];
    limsuplambda1295[j][0]=lambda12[j][0]+
    1.96*desvplambda12[j][0];
    liminfs195[j][0]=s1[j][0]-1.96*desvps1[j][0];
    limsups195[j][0]=s1[j][0]+1.96*desvps1[j][0];
    liminfs295[j][0]=s2[j][0]-1.96*desvps2[j][0];
    limsups295[j][0]=s2[j][0]+1.96*desvps2[j][0];
}

/*Médias EMV*/
decl medialambda1,medialambda2,medialambda12,medias1,medias2;

```

```

medialambda1=meanc(lambda1);
medialambda2=meanc(lambda2);
medialambda12=meanc(lambda12);
medias1=meanc(s1);
medias2=meanc(s2);

// Médias dos desvios padrões das estimativas
decl mediadplambda1, mediadplambda2, mediadplambda12,
mediadps1, mediadps2;
mediadplambda1=meanc(desvplambda1);
mediadplambda2=meanc(desvplambda2);
mediadplambda12=meanc(desvplambda12);
mediadps1=meanc(desvps1);
mediadps2=meanc(desvps2);

// médias dos erros quadráticos médios das estimativas
decl mediaeqmlambda1, mediaeqmlambda2, mediaeqmlambda12,
mediaeqms1, mediaeqms2;
mediaeqmlambda1=meanc(eqmlambda1);
mediaeqmlambda2=meanc(eqmlambda2);
mediaeqmlambda12=meanc(eqmlambda12);
mediaeqms1=meanc(eqms1);
mediaeqms2=meanc(eqms2);

// média das medidas STD
decl mediaSTD;
mediaSTD=meanc(STD);

// desvios padrões das estimativas de MV
decl desvpalambda1, desvpalambda2, desvpalambda12,
desvps1, desvps2;
desvpalambda1=sqrt(varc(lambda1));
desvpalambda2=sqrt(varc(lambda2));
desvpalambda12=sqrt(varc(lambda12));
desvps1=sqrt(varc(s1));
desvps2=sqrt(varc(s2));

// desvio padrão dos desvios padrões das estimativas de MV
decl dpdplambda1, dpdplambda2, dpdplambda12, dpdps1, dpdps2;
dpdplambda1=sqrt(varc(desvplambda1));
dpdplambda2=sqrt(varc(desvplambda2));
dpdplambda12=sqrt(varc(desvplambda12));
dpdps1=sqrt(varc(desvps1));
dpdps2=sqrt(varc(desvps2));

// desvios padrões dos erros quadráticos médios
decl dpeqmlambda1, dpeqmlambda2, dpeqmlambda12,
dpeqms1, dpeqms2;
dpeqmlambda1=sqrt(varc(eqmlambda1));
dpeqmlambda2=sqrt(varc(eqmlambda2));
dpeqmlambda12=sqrt(varc(eqmlambda12));
dpeqms1=sqrt(varc(eqms1));
dpeqms2=sqrt(varc(eqms2));

// desvio padrão dos STD
decl dpSTD;
dpSTD=sqrt(varc(STD));

/* quantidade intervalos 90% que não contém o verdadeiro
valor dos parâmetros*/
for (k=0;k<h;++k)
{
if ((liminflambda190[k][0]>0.1||
limsuplambda190[k][0]<0.1))
quantlambda190=quantlambda190+1;
if ((liminflambda290[k][0]>0.1||
limsuplambda290[k][0]<0.1))
quantlambda290=quantlambda290+1;
if ((liminflambda1290[k][0]>0.2||
limsuplambda1290[k][0]<0.2))
quantlambda1290=quantlambda1290+1;
if ((liminf190[k][0]>0.5|| limsup190[k][0]<0.5))
quants190=quants190+1;
if ((liminf290[k][0]>0.5|| limsup290[k][0]<0.5))
quants290=quants290+1;
}

// cobertura dos intervalos com 90% de confiança
cobertura90lambda1=((h-quantlambda190)/h)*100;
cobertura90lambda2=((h-quantlambda290)/h)*100;
cobertura90lambda12=((h-quantlambda1290)/h)*100;
cobertura90s1=((h-quants190)/h)*100;
cobertura90s2=((h-quants290)/h)*100;

/*----quantidade intervalos 95% que não contiveram o
verdadeiro valor dos parâmetros*/
for (k=0;k<h;++k)
{
if ((liminflambda195[k][0]>0.1||
limsuplambda195[k][0]<0.1))
quantlambda195=quantlambda195+1;
if ((liminflambda295[k][0]>0.1||
limsuplambda295[k][0]<0.1))
quantlambda295=quantlambda295+1;
if ((liminflambda1295[k][0]>0.2||
limsuplambda1295[k][0]<0.2))

```



```

decl l,ir,mhoss;
decl rig=zeros(50,2);
u=500; // total de amostras

decl lamb1=zeros(u,1);
decl lamb2=zeros(u,1);
decl lamb12=zeros(u,1);
decl s11=zeros(u,1);
decl s12=zeros(u,1);

decl lamb1a=zeros(it,1);
decl lamb2a=zeros(it,1);
decl lamb12a=zeros(it,1);
decl s11a=zeros(it,1);
decl s12a=zeros(it,1);

decl varlambda1=zeros(u,1);
decl varlambda2=zeros(u,1);
decl varlambda12=zeros(u,1);
decl varsl1=zeros(u,1);
decl varsl2=zeros(u,1);

decl dplambda1=zeros(u,1);
decl dplambda2=zeros(u,1);
decl dplambda12=zeros(u,1);
decl dps1=zeros(u,1);
decl dps2=zeros(u,1);

decl matrizlambda1=zeros(it,m);
decl matrizlambda2=zeros(it,m);
decl matrizlambda12=zeros(it,m);
decl matrizsl1=zeros(it,m);
decl matrizsl2=zeros(it,m);

decl matrizlambda1ok=zeros(itfm,m);
decl matrizlambda2ok=zeros(itfm,m);
decl matrizlambda12ok=zeros(itfm,m);
decl matrizsl1ok=zeros(itfm,m);
decl matrizsl2ok=zeros(itfm,m);

decl lambda1aux=zeros(6000,1);
decl lambda2aux=zeros(6000,1);
decl lambda12aux=zeros(6000,1);
decl slaux=zeros(6000,1);
decl sl2aux=zeros(6000,1);

decl vetorlambda1=zeros(2500,1);
decl vetorlambda2=zeros(2500,1);

decl vetorlambda12=zeros(2500,1);
decl vetors1=zeros(2500,1);
decl vetors2=zeros(2500,1);

decl vetorlambda1ok=zeros(1200,1);
decl vetorlambda2ok=zeros(1200,1);
decl vetorlambda12ok=zeros(1200,1);
decl vetors1ok=zeros(1200,1);
decl vetors2ok=zeros(1200,1);

decl R1=zeros(u,1);
decl R2=zeros(u,1);
decl R3=zeros(u,1);
decl R4=zeros(u,1);
decl R5=zeros(u,1);

decl mediavetorlambda1ok=zeros(u,1);
decl mediavetorlambda2ok=zeros(u,1);
decl mediavetorlambda12ok=zeros(u,1);
decl mediavetors1ok=zeros(u,1);
decl mediavetors2ok=zeros(u,1);

decl varvetorlambda1ok=zeros(u,1);
decl varvetorlambda2ok=zeros(u,1);
decl varvetorlambda12ok=zeros(u,1);
decl varvetors1ok=zeros(u,1);
decl varvetors2ok=zeros(u,1);

decl dpvetorlambda1ok=zeros(u,1);
decl dpvetorlambda2ok=zeros(u,1);
decl dpvetorlambda12ok=zeros(u,1);
decl dpvetors1ok=zeros(u,1);
decl dpvetors2ok=zeros(u,1);

decl vetorlambda1okord=zeros(1200,1);
decl vetorlambda2okord=zeros(1200,1);
decl vetorlambda12okord=zeros(1200,1);
decl vetors1okord=zeros(1200,1);
decl vetors2okord=zeros(1200,1);

decl medianavetorlambda1ok=zeros(u,1);
decl medianavetorlambda2ok=zeros(u,1);
decl medianavetorlambda12ok=zeros(u,1);
decl medianavetors1ok=zeros(u,1);
decl medianavetors2ok=zeros(u,1);

decl medobscad1lamb1,medobscad2lamb1,medobscad3lamb1;
decl medobscad4lamb1,medobscad5lamb1;

```

```

decl medobscad11amb2, medobscad21amb2, medobscad31amb2;
decl medobscad41amb2, medobscad51amb2;
decl medobscad11amb12, medobscad21amb12, medobscad31amb12;
decl medobscad41amb12, medobscad51amb12;
decl medobscad1s1, medobscad2s1, medobscad3s1;
decl medobscad4s1, medobscad5s1;
decl medobscad1s2, medobscad2s2, medobscad3s2;
decl medobscad4s2, medobscad5s2;

decl quadmedobscad11amb1, quadmedobscad21amb1;
decl quadmedobscad31amb1;
decl quadmedobscad41amb1, quadmedobscad51amb1;
decl quadmedobscad11amb2, quadmedobscad21amb2;
decl quadmedobscad31amb2;
decl quadmedobscad41amb2, quadmedobscad51amb2;
decl quadmedobscad11amb12, quadmedobscad21amb12;
decl quadmedobscad31amb12;
decl quadmedobscad41amb12, quadmedobscad51amb12;
decl quadmedobscad1s1, quadmedobscad2s1, quadmedobscad3s1;
decl quadmedobscad4s1, quadmedobscad5s1;
decl quadmedobscad1s2, quadmedobscad2s2, quadmedobscad3s2;
decl quadmedobscad4s2, quadmedobscad5s2;

decl vetoruns=ones(it,1); decl vetorzeros=zeros(it,1);

decl vetor10000=zeros(2500,1);
decl vetor01000=zeros(2500,1);
decl vetor00100=zeros(2500,1);
decl vetor00010=zeros(2500,1);
decl vetor00001=zeros(2500,1);

decl lambda1c, logfi1c, logfi1, razao1, minimo1, un1, uni1;
decl lambda2c, logfi2c, logfi2, razao2, minimo2, un2, uni2;
decl lambda12c, logfi3c, logfi3, razao3, minimo3, un3, uni3;
decl s1c, logfi4c, logfi4, razao4, minimo4, un4, uni1;
decl s2c, logfi5c, logfi5, razao5, minimo5, un5, uni2;

decl gamal1=zeros(3,1);
decl gamal2=zeros(3,1);
decl gamal12=zeros(3,1);
decl gamas1=zeros(3,1);
decl gamas2=zeros(3,1);

decl quadi fobsemedcad11amb1=zeros(it,1);
decl quadi fobsemedcad21amb1=zeros(it,1);
decl quadi fobsemedcad31amb1=zeros(it,1);
decl quadi fobsemedcad41amb1=zeros(it,1);
decl quadi fobsemedcad51amb1=zeros(it,1);

decl quadi fobsemedcad11amb2=zeros(it,1);
decl quadi fobsemedcad21amb2=zeros(it,1);
decl quadi fobsemedcad31amb2=zeros(it,1);
decl quadi fobsemedcad41amb2=zeros(it,1);
decl quadi fobsemedcad51amb2=zeros(it,1);

decl quadi fobsemedcad11amb12=zeros(it,1);
decl quadi fobsemedcad21amb12=zeros(it,1);
decl quadi fobsemedcad31amb12=zeros(it,1);
decl quadi fobsemedcad41amb12=zeros(it,1);
decl quadi fobsemedcad51amb12=zeros(it,1);

decl quadi fobsemedcad1s1=zeros(it,1);
decl quadi fobsemedcad2s1=zeros(it,1);
decl quadi fobsemedcad3s1=zeros(it,1);
decl quadi fobsemedcad4s1=zeros(it,1);
decl quadi fobsemedcad5s1=zeros(it,1);

decl quadi fobsemedcad1s2=zeros(it,1);
decl quadi fobsemedcad2s2=zeros(it,1);
decl quadi fobsemedcad3s2=zeros(it,1);
decl quadi fobsemedcad4s2=zeros(it,1);
decl quadi fobsemedcad5s2=zeros(it,1);

decl medtot1amb1, medtot1amb2, medtot1amb12;
decl medtots1, medtots2;

decl siquad1=zeros(m,1);
decl siquad2=zeros(m,1);
decl siquad3=zeros(m,1);
decl siquad4=zeros(m,1);
decl siquad5=zeros(m,1);

decl varsiquad1, varsiquad2, varsiquad3;
decl varsiquad4, varsiquad5;

decl matriz1a=zeros(m,2);
decl matriz2a=zeros(m,2);
decl matriz3a=zeros(m,2);
decl matriz4a=zeros(m,2);
decl matriz5a=zeros(m,2);

decl matriz1b=zeros(m,2);
decl matriz2b=zeros(m,2);
decl matriz3b=zeros(m,2);
decl matriz4b=zeros(m,2);
decl matriz5b=zeros(m,2);

```

```

decl covar1a=zeros(2,2);
decl covar2a=zeros(2,2);
decl covar3a=zeros(2,2);
decl covar4a=zeros(2,2);
decl covar5a=zeros(2,2);

decl covar1b=zeros(2,2);
decl covar2b=zeros(2,2);
decl covar3b=zeros(2,2);
decl covar4b=zeros(2,2);
decl covar5b=zeros(2,2);

decl cov1a, cov1b, V1, sigma21, varV1, df1;
decl cov2a, cov2b, V2, sigma22, varV2, df2;
decl cov3a, cov3b, V3, sigma23, varV3, df3;
decl cov4a, cov4b, V4, sigma24, varV4, df4;
decl cov5a, cov5b, V5, sigma25, varV5, df5;

decl Dlamb1, Elamb1, Rlamb1;
decl Dlamb2, Elamb2, Rlamb2;
decl Dlamb12, Elamb12, Rlamb12;
decl Ds1, Es1, Rs1, Ds2, Es2, Rs2;

decl quadifmedlamb1=zeros(m,1);
decl quadifmedlamb2=zeros(m,1);
decl quadifmedlamb12=zeros(m,1);
decl quadifmeds1=zeros(m,1);
decl quadifmeds2=zeros(m,1);

decl medobscadlamb1=zeros(m,1);
decl medobscadlamb2=zeros(m,1);
decl medobscadlamb12=zeros(m,1);
decl medobscads1=zeros(m,1);
decl medobscads2=zeros(m,1);

decl quadmedobscadlamb1=zeros(m,1);
decl quadmedobscadlamb2=zeros(m,1);
decl quadmedobscadlamb12=zeros(m,1);
decl quadmedobscads1=zeros(m,1);
decl quadmedobscads2=zeros(m,1);

decl mediaemvlambda1, mediaemvlambda2, mediaemvlambda12,
  mediaemvs1, mediaemvs2;
decl mediavaremvlambda1, mediavaremvlambda2,
  mediavaremvlambda12, mediavaremvs1, mediavaremvs2;
decl mediadpemvlambda1, mediadpemvlambda2, mediadpemvlambda12,
  mediadpemvs1, mediadpemvs2;

decl mediamediaslamb1, mediamediaslamb2, mediamediaslamb12,
  mediamediass1, mediamediass2;
decl mediavarslamb1, mediavarslamb2, mediavarslamb12,
  mediavarss1, mediavarss2;
decl mediadplamb1, mediadplamb2, mediadplamb12, mediadps1,
  mediadps2;
decl maxmaxlamb1, maxmaxlamb2, maxmaxlamb12, maxmaxs1,
  maxmaxs2;
decl minminlamb1, minminlamb2, minminlamb12, minmins1,
  minmins2;
decl mediamediana1lamb1, mediamediana1lamb2,
  mediamediana1lamb12, mediamediana1, mediamediana2;

decl liminflambda190=zeros(u,1);
decl liminflambda195=zeros(u,1);
decl limsuplambda190=zeros(u,1);
decl limsuplambda195=zeros(u,1);
decl liminflambda290=zeros(u,1);
decl liminflambda295=zeros(u,1);
decl limsuplambda290=zeros(u,1);
decl limsuplambda295=zeros(u,1);
decl liminflambda1290=zeros(u,1);
decl liminflambda1295=zeros(u,1);
decl limsuplambda1290=zeros(u,1);
decl limsuplambda1295=zeros(u,1);
decl liminfsl90=zeros(u,1);
decl liminfsl95=zeros(u,1);
decl limsupsl90=zeros(u,1);
decl limsupsl95=zeros(u,1);
decl liminfsl290=zeros(u,1);
decl liminfsl295=zeros(u,1);
decl limsupsl290=zeros(u,1);
decl limsupsl295=zeros(u,1);

decl ic90supl1=zeros(u,1);
decl ic90supl2=zeros(u,1);
decl ic90supl12=zeros(u,1);
decl ic90supsl=zeros(u,1);
decl ic90supsl2=zeros(u,1);
decl ic90supsl12=zeros(u,1);
decl ic90infl1=zeros(u,1);
decl ic90infl2=zeros(u,1);
decl ic90infl12=zeros(u,1);
decl ic90infs1=zeros(u,1);
decl ic90infs2=zeros(u,1);

decl ic95supl1=zeros(u,1);
decl ic95supl2=zeros(u,1);
decl ic95supl12=zeros(u,1);

```

```

decl ic95supsl=zeros(u,1);
decl ic95sup2=zeros(u,1);
decl ic95infl1=zeros(u,1);
decl ic95infl2=zeros(u,1);
decl ic95infl12=zeros(u,1);
decl ic95infs1=zeros(u,1);
decl ic95infs2=zeros(u,1);

decl eqmlambda1=zeros(u,1);
decl eqmlambda2=zeros(u,1);
decl eqmlambda12=zeros(u,1);
decl eqms1=zeros(u,1);
decl eqms2=zeros(u,1);

decl eqmmedialambda1=zeros(u,1);
decl eqmmedialambda2=zeros(u,1);
decl eqmmedialambda12=zeros(u,1);
decl eqmmedias1=zeros(u,1);
decl eqmmedias2=zeros(u,1);

decl eqmmedialambda1=zeros(u,1);
decl eqmmedialambda2=zeros(u,1);
decl eqmmedialambda12=zeros(u,1);
decl eqmmedias1=zeros(u,1);
decl eqmmedias2=zeros(u,1);

decl mediaeqmlambda1, mediaeqmlambda2, mediaeqmlambda12,
  mediaeqms1, mediaeqms2;
decl mediaeqmmedialambda1, mediaeqmmedialambda2,
  mediaeqmmedialambda12, mediaeqmmedias1, mediaeqmmedias2;
decl mediaeqmmedialambda1, mediaeqmmedialambda2,
  mediaeqmmedialambda12, mediaeqmmedias1, mediaeqmmedias2;

b=0;
while(b<500)
{
  tig=gera();
  t1=tig*<1;0>;
  t2=tig*<0;1>;

  /*maximização de log da verossimilhança*/

  decl vP=<1;1;1;1>; // chutes iniciais
  logfveros(vP,&t,0,0);
  MaxControl(1000,1000);
  mhess=0.0001*unit(5);
  ir=MaxBFGS(logfveros,&vP,&t,&mhess,TRUE);
}
if(ir==MAX_CONV && vP[0][0]<1 && vP[1][0]<1
  && vP[2][0]<2 && vP[3][0]<5 && vP[4][0]<5)
{
  /*EMV*/
  lamb1[b][0]=vP[0][0];
  lamb2[b][0]=vP[1][0];
  lamb12[b][0]=vP[2][0];
  s11[b][0]=vP[3][0];
  s12[b][0]=vP[4][0];

  /*variância EMV*/
  Num2Derivative(logfveros,vP,&mhess);
  MatInfObs=mhess*(-1);
  invMatInfObs=invert(MatInfObs);
  varlambda1[b][0]=invMatInfObs [0][0];
  varlambda2[b][0]=invMatInfObs [1][1];
  varlambda12[b][0]=invMatInfObs [2][2];
  vars1[b][0]=invMatInfObs [3][3];
  vars2[b][0]=invMatInfObs [4][4];

  dplambda1[b][0]=sqrt(varlambda1[b][0]);
  dplambda2[b][0]=sqrt(varlambda2[b][0]);
  dplambda12[b][0]=sqrt(varlambda12[b][0]);
  dps1[b][0]=sqrt(vars1[b][0]);
  dps2[b][0]=sqrt(vars2[b][0]);

  //hiperparâmetros das gamas
  a1=(lamb1[b][0])^2/varlambda1[b][0];
  b1=lamb1[b][0]/varlambda1[b][0];
  a2=(lamb2[b][0])^2/varlambda2[b][0];
  b2=lamb2[b][0]/varlambda2[b][0];
  a3=(lamb12[b][0])^2/varlambda12[b][0];
  b3=lamb12[b][0]/varlambda12[b][0];
  a4=(s11[b][0])^2/vars1[b][0];
  b4=s11[b][0]/vars1[b][0];
  a5=(s12[b][0])^2/vars2[b][0];
  b5=s12[b][0]/vars2[b][0];

  h=0; //h=cadeia
  for(h=0;h<5;++h)
  {
    if(h==0)
    { //chutes iniciais 1ªcadeia
      lambda1=vP[0][0];
      lambda2=vP[1][0];
      lambda12=vP[2][0];
      s1=vP[3][0];
    }
  }
}

```



```

s2=vP[4][0];
}
if(h==1)
{ //chutes iniciais 2ª cadeia
lambda1=0.01;
lambda2=0.01;
lambda12=0.01;
s1=0.01;
s2=0.01;
}
if(h==2)
{ //chutes iniciais 3ª cadeia
lambda1=0.5;
lambda2=0.5;
lambda12=0.5;
s1=0.5;
s2=0.5;
}
if(h==3)
{ //chutes iniciais 4ª cadeia
lambda1=1;
lambda2=1;
lambda12=2;
s1=5;
s2=5;
}
if(h==4)
{ //chutes iniciais 5ª cadeia
lambda1=0;
lambda2=0;
lambda12=0;
s1=0;
s2=0;
}

j=0; //j=interação
while (j<6000)
{
gamal1=rangamma(3,1,a1,b1);
lambda1c=gamal1[2][0]; //candidato a lambda1
vPb[0][0]=lambda1c;
vPb[1][0]=lambda2;
vPb[2][0]=lambda12;
vPb[3][0]=s1;
vPb[4][0]=s2;
logfilc=logfi(t1,t2,vPb);
vPb[0][0]=lambda1;
logfil=logfi(t1,t2,vPb);

razao1=exp(logfilc-logfi1);
minimo1=min(razao1,1);
uni1=ranu(3,1);
un1=uni1[2];
if(un1<=minimo1)
{
lambda1=lambda1c;
}
else
{
lambda1=vPb[0][0];
}
lambda1aux[j][0]=lambda1;

gamal2=rangamma(3,1,a2,b2);
lambda2c=gamal2[2][0]; //candidato a lambda2
vPb[0][0]=lambda1;
vPb[1][0]=lambda2c;
logfi2c=logfi(t1,t2,vPb);
vPb[1][0]=lambda2;
logfi2=logfi(t1,t2,vPb);
razao2=exp(logfi2c-logfi2);
minimo2=min(razao2,1);
uni2=ranu(3,1);
un2=uni2[2];
if(un2<=minimo2)
{
lambda2=lambda2c;
}
else
{
lambda2=vPb[1][0];
}
lambda2aux[j][0]=lambda2;

gamal12=rangamma(3,1,a3,b3);
lambda12c=gamal12[2][0]; // candidato a lambda12
vPb[0][0]=lambda1;
vPb[1][0]=lambda2;
vPb[2][0]=lambda12c;
logfi3c=logfi(t1,t2,vPb);
vPb[2][0]=lambda12;
logfi3=logfi(t1,t2,vPb);
razao3=exp(logfi3c-logfi3);
minimo3=min(razao3,1);
uni3=ranu(3,1);
un3=uni3[2];
if(un3<=minimo3)
{

```

```

lambda12=lambda12c;
}
else
{
lambda12=vPb[2][0];
}
lambda12aux[j][0]=lambda12;

gamas1=rangamma(3,1,a4,b4);
s1c=gamas1[2][0]; // candidato a s1
vPb[0][0]=lambda1;
vPb[1][0]=lambda2;
vPb[2][0]=lambda12;
vPb[3][0]=s1c;
logfi4c=logfi(t1,t2,vPb);
vPb[3][0]=s1;
logfi4=logfi(t1,t2,vPb);
razao4=exp(logfi4c-logfi4);
minimo4=min(razao4,1);
unis1=ranu(3,1);
un4=unis1[2];
if(un4<=minimo4)
{
s1=s1c;
}
else
{
s1=vPb[3][0];
}
s1aux[j][0]=s1;

gamas2=rangamma(3,1,a5,b5);
s2c=gamas2[2][0]; //candidato a s2
vPb[0][0]=lambda1;
vPb[1][0]=lambda2;
vPb[2][0]=lambda12;
vPb[3][0]=s1;
vPb[4][0]=s2c;
logfi5c=logfi(t1,t2,vPb);
vPb[4][0]=s2;
logfi5=logfi(t1,t2,vPb);
razao5=exp(logfi5c-logfi5);
minimo5=min(razao5,1);
unis2=ranu(3,1);
un5=unis2[2];
if(un5<=minimo5)
{
s2=s2c;
}
else
{
s2=vPb[4][0];
}
s2aux[j][0]=s2;
i++;
}
}
//desprezando as 1000as observações e selecionando as
demais de 10 em 10
decl l;
g=0;
for(l=1000;l<6000;l+=10)
{
lambda[g][0]=lambda1aux[l][0];
lamb2a[g][0]=lambda2aux[l][0];
lamb12a[g][0]=lambda12aux[l][0];
s1a[g][0]=s1aux[l][0];
s12a[g][0]=s2aux[l][0];
++g;
}

//construindo matrizes com os vetores colunas
decl x;
for(x=0;x<500;++x)
{
matrizlambda1[x][h]=lambda[x][0];
matrizlambda2[x][h]=lamb2a[x][0];
matrizlambda12[x][h]=lamb12a[x][0];
matrizs1[x][h]=s1a[x][0];
matrizs2[x][h]=s12a[x][0];
}
} //fecha looping h

decl s;
for(h=0;h<n;++h)
{
decl v=0;
for(s=260;s<500;++s)
{
matrizlambda1ok[v][h]=matrizlambda1[s][h];
matrizlambda2ok[v][h]=matrizlambda2[s][h];
matrizlambda12ok[v][h]=matrizlambda12[s][h];
matrizs1ok[v][h]=matrizs1[s][h];
matrizs2ok[v][h]=matrizs2[s][h];
++v;
}
}

```

```

}
//amostras das distribuições a posteriori
vetorlambda1=vec(matrizlambda1);
vetorlambda2=vec(matrizlambda2);
vetorlambda12=vec(matrizlambda12);
vetors1=vec(matrizs1);
vetors2=vec(matrizs2);

vetorlambda1ok=vec(matrizlambda1ok);
vetorlambda2ok=vec(matrizlambda2ok);
vetorlambda12ok=vec(matrizlambda12ok);
vetors1ok=vec(matrizs1ok);
vetors2ok=vec(matrizs2ok);

//sumário das amostras das distribuições a posteriori
mediavetorlambda1ok[b][0]=meanc(vetorlambda1ok);
mediavetorlambda2ok[b][0]=meanc(vetorlambda2ok);
mediavetorlambda12ok[b][0]=meanc(vetorlambda12ok);
mediavetors1ok[b][0]=meanc(vetors1ok);
mediavetors2ok[b][0]=meanc(vetors2ok);

varvetorlambda1ok[b][0]=varc(vetorlambda1ok);
varvetorlambda2ok[b][0]=varc(vetorlambda2ok);
varvetorlambda12ok[b][0]=varc(vetorlambda12ok);
varvetors1ok[b][0]=varc(vetors1ok);
varvetors2ok[b][0]=varc(vetors2ok);

dpvetorlambda1ok[b][0]=sqrt(varvetorlambda1ok[b][0]);
dpvetorlambda2ok[b][0]=sqrt(varvetorlambda2ok[b][0]);
dpvetorlambda12ok[b][0]=sqrt(varvetorlambda12ok[b][0]);
dpvetors1ok[b][0]=sqrt(varvetors1ok[b][0]);
dpvetors2ok[b][0]=sqrt(varvetors2ok[b][0]);

vetorlambda1okord=sortc(vetorlambda1ok);
vetorlambda2okord=sortc(vetorlambda2ok);
vetorlambda12okord=sortc(vetorlambda12ok);
vetors1okord=sortc(vetors1ok);
vetors2okord=sortc(vetors2ok);

medianavetorlambda1ok[b][0]={vetorlambda1okord[599][0]+
vetorlambda1okord[600][0]}/2;
medianavetorlambda2ok[b][0]={vetorlambda2okord[599][0]+
vetorlambda2okord[600][0]}/2;
medianavetorlambda12ok[b][0]={vetorlambda12okord[599][0]+
vetorlambda12okord[600][0]}/2;
medianavetors1ok[b][0]={vetors1okord[599][0]+
vetors1okord[600][0]}/2;
medianavetors2ok[b][0]={vetors2okord[599][0]+
vetors2okord[600][0]}/2;

ic95inf11[b][0]=vetorlambda1okord[29][0];
ic95inf12[b][0]=vetorlambda2okord[29][0];
ic95inf112[b][0]=vetorlambda12okord[29][0];
ic95inf1[b][0]=vetors1okord[29][0];
ic95inf2[b][0]=vetors2okord[29][0];

ic95sup11[b][0]=vetorlambda1okord[1170][0];
ic95sup12[b][0]=vetorlambda2okord[1170][0];
ic95sup112[b][0]=vetorlambda12okord[1170][0];
ic95sup1[b][0]=vetors1okord[1170][0];
ic95sup2[b][0]=vetors2okord[1170][0];

ic90inf11[b][0]=vetorlambda1okord[59][0];
ic90inf12[b][0]=vetorlambda2okord[59][0];
ic90inf112[b][0]=vetorlambda12okord[59][0];
ic90inf1[b][0]=vetors1okord[59][0];
ic90inf2[b][0]=vetors2okord[59][0];

ic90sup11[b][0]=vetorlambda1okord[1140][0];
ic90sup12[b][0]=vetorlambda2okord[1140][0];
ic90sup112[b][0]=vetorlambda12okord[1140][0];
ic90sup1[b][0]=vetors1okord[1140][0];
ic90sup2[b][0]=vetors2okord[1140][0];

/*Verificacao de convergencia*/
vetor10000=(vetoruns|vetorzeros|vetorzeros|vetorzeros|
vetorzeros);
vetor01000=(vetorzeros|vetoruns|vetorzeros|vetorzeros|
vetorzeros);
vetor00100=(vetorzeros|vetorzeros|vetoruns|vetorzeros|
vetorzeros);
vetor00010=(vetorzeros|vetorzeros|vetorzeros|vetoruns|
vetorzeros);
vetor00001=(vetorzeros|vetorzeros|vetorzeros|vetorzeros|
vetoruns);

/*lambda1*/
medtotlamb1=meanc(vetorlambda1);
medobscad1lamb1={({vetor10000}'*vetorlambda1)/it;
medobscad2lamb1={({vetor01000}'*vetorlambda1)/it;
medobscad3lamb1={({vetor00100}'*vetorlambda1)/it;
medobscad4lamb1={({vetor00010}'*vetorlambda1)/it;
medobscad5lamb1={({vetor00001}'*vetorlambda1)/it;

quadmedobscad1lamb1=medobscad1lamb1^2;

```

```

quadmedobscad2lamb1=medobscad2lamb1^2;
quadmedobscad3lamb1=medobscad3lamb1^2;
quadmedobscad4lamb1=medobscad4lamb1^2;
quadmedobscad5lamb1=medobscad5lamb1^2;

o=0; i=0;
for(o=0; o<500; ++o)
{
quadifobsemedcad1lamb1[i][0]=(vetorlambdal[o][0]-
medobscad1lamb1)^2;
++i;
}
siquad1[0][0]=(1/(it-1))*sumc(quadifobsemedcad1lamb1);

i=0;
for(o=500; o<1000; ++o)
{
quadifobsemedcad2lamb1[i][0]=(vetorlambdal[o][0]-
medobscad2lamb1)^2;
++i;
}
siquad1[1][0]=(1/(it-1))*sumc(quadifobsemedcad2lamb1);

i=0;
for(o=1000; o<1500; ++o)
{
quadifobsemedcad3lamb1[i][0]=(vetorlambdal[o][0]-
medobscad3lamb1)^2;
++i;
}
siquad1[2][0]=(1/(it-1))*sumc(quadifobsemedcad3lamb1);

i=0;
for(o=1500; o<2000; ++o)
{
quadifobsemedcad4lamb1[i][0]=(vetorlambdal[o][0]-
medobscad4lamb1)^2;
++i;
}
siquad1[3][0]=(1/(it-1))*sumc(quadifobsemedcad4lamb1);

i=0;
for(o=2000; o<2500; ++o)
{
quadifobsemedcad5lamb1[i][0]=(vetorlambdal[o][0]-
medobscad5lamb1)^2;
++i;
}
siquad1[4][0]=(1/(it-1))*sumc(quadifobsemedcad5lamb1);

varsiquad1=varc(siquad1)*5/4;
Dlamb1=(1/m)*sumc(siquad1);
medobscad1lamb1=medobscad1lamb1|medobscad2lamb1|
medobscad3lamb1|medobscad4lamb1|medobscad5lamb1;
quadifmedlamb1=(medobscad1lamb1-medtotlamb1).^2;
Elamb1=(it/(m-1))*(sumc(quadifmedlamb1));

quadmedobscad1lamb1=quadmedobscad1lamb1|
quadmedobscad2lamb1|
quadmedobscad3lamb1|quadmedobscad4lamb1|
quadmedobscad5lamb1;
matriz1a=siquad1~medobscad1lamb1;
matriz1b=siquad1~quadmedobscad1lamb1;
covar1a=variance(matriz1a);
covar1b=variance(matriz1b);
cov1a=covar1a[1][0]*5/4;
cov1b=covar1b[1][0]*5/4;
varV1=((it-1)/it)^2*(1/m)*varsiquad1+((m+1)/(m*it))^2*
(2/(m-1))*Elamb1^2+2*(((m+1)*(it-1))/(m*it^2))*
(it/m)*(cov1b-2*medtotlamb1*cov1a);
sigma21=((it-1)/it)*Dlamb1+Elamb1/it;
V1=sigma21*Elamb1/(m*it);
df1=(2*V1^2)/varV1;
Rlamb1=sqrt((V1/Dlamb1)*(df1/(df1-2)));
R1[b][0]=Rlamb1;

/*lambda2*/
medtotlamb2=meanc(vetorlambda2);

medobscad1lamb2=((vetor10000')*vetorlambda2)/it;
medobscad2lamb2=((vetor01000')*vetorlambda2)/it;
medobscad3lamb2=((vetor00100')*vetorlambda2)/it;
medobscad4lamb2=((vetor00010')*vetorlambda2)/it;
medobscad5lamb2=((vetor00001')*vetorlambda2)/it;

quadmedobscad1lamb2=medobscad1lamb2^2;
quadmedobscad2lamb2=medobscad2lamb2^2;
quadmedobscad3lamb2=medobscad3lamb2^2;
quadmedobscad4lamb2=medobscad4lamb2^2;
quadmedobscad5lamb2=medobscad5lamb2^2;

o=0; i=0;
for(o=0; o<500; ++o)
{
quadifobsemedcad1lamb2[i][0]=(vetorlambda2[o][0]-
medobscad1lamb2)^2;
}

```



```

++i;
}
siquad2[0][0]=(1/(it-1))*sumc(quadifobsemedcad1lamb2);

i=0;
for(o=500;o<1000;++o)
{
quadifobsemedcad2lamb2[i][0]=(vetorlambda2[o][0]-
medobscad2lamb2)^2;
++i;
}
siquad2[1][0]=(1/(it-1))*sumc(quadifobsemedcad2lamb2);

i=0;
for(o=1000;o<1500;++o)
{
quadifobsemedcad3lamb2[i][0]=(vetorlambda2[o][0]-
medobscad3lamb2)^2;
++i;
}
siquad2[2][0]=(1/(it-1))*sumc(quadifobsemedcad3lamb2);

i=0;
for(o=1500;o<2000;++o)
{
quadifobsemedcad4lamb2[i][0]=(vetorlambda2[o][0]-
medobscad4lamb2)^2;
++i;
}
siquad2[3][0]=(1/(it-1))*sumc(quadifobsemedcad4lamb2);

i=0;
for(o=2000;o<2500;++o)
{
quadifobsemedcad5lamb2[i][0]=(vetorlambda2[o][0]-
medobscad5lamb2)^2;
++i;
}
siquad2[4][0]=(1/(it-1))*sumc(quadifobsemedcad5lamb2);

varsiquad2=varc(siquad2)*5/4;
Dlamb2=(1/m)*sumc(siquad2);

medobscad1lamb2=medobscad1lamb2|medobscad2lamb2|
medobscad3lamb2|medobscad4lamb2|medobscad5lamb2;
quadifmed1lamb2=(medobscad1lamb2-medtotlamb2)^2;
Elamb2=(it/(m-1))*sumc(quadifmed1lamb2);
quadmedobscad1lamb2=quadmedobscad1lamb2|quadmedobscad2lamb2|

```

```

quadmedobscad3lamb2|quadmedobscad4lamb2|
quadmedobscad5lamb2;
matriz2a=siquad2-medobscad1lamb2;
matriz2b=siquad2-quadmedobscad1lamb2;
covar2a=variance(matriz2a);
covar2b=variance(matriz2b);
cov2a=covar2a[1][0]*5/4;
cov2b=covar2b[1][0]*5/4;
varV2=((it-1)/it)^2*(1/m)*varsiquad2+((m+1)/(m*it))^2*
(2/(m-1))*Elamb2^2+2*((m+1)*(it-1)/(m*it^2))*(it/m)*
(cov2b-2*medtotlamb2*cov2a);
sigma22=((it-1)/it)*Dlamb2+Elamb2/it;
V2=sigma22+Elamb2/(m*it);
df2=(2*V2^2)/varV2;
Rlamb2=sqrt((V2/Dlamb2)*(df2/(df2-2)));
R2[b][0]=Rlamb2;

/*lambda12*/
medtotlamb12=meanc(vetorlambda12);

medobscad1lamb12={(vetor10000')*vetorlambda12}/it;
medobscad2lamb12={(vetor01000')*vetorlambda12}/it;
medobscad3lamb12={(vetor00100')*vetorlambda12}/it;
medobscad4lamb12={(vetor00010')*vetorlambda12}/it;
medobscad5lamb12={(vetor00001')*vetorlambda12}/it;

quadmedobscad1lamb12=medobscad1lamb12^2;
quadmedobscad2lamb12=medobscad2lamb12^2;
quadmedobscad3lamb12=medobscad3lamb12^2;
quadmedobscad4lamb12=medobscad4lamb12^2;
quadmedobscad5lamb12=medobscad5lamb12^2;

o=0; i=0;
for(o=0;o<500;++o)
{
quadifobsemedcad1lamb12[i][0]=(vetorlambda12[o][0]-
medobscad1lamb12)^2;
++i;
}
siquad3[0][0]=(1/(it-1))*sumc(quadifobsemedcad1lamb12);

i=0;
for(o=500;o<1000;++o)
{
quadifobsemedcad2lamb12[i][0]=(vetorlambda12[o][0]-
medobscad2lamb12)^2;
++i;
}

```

```

siquad3[1][0]=(1/(it-1))*sumc(quadifobsemedcad2lamb12);

i=0;
for(o=1000;o<1500;++o)
{
quadifobsemedcad3lamb12[i][0]=(vetorlambd12[o][0]-
medobscad3lamb12)^2;
++i;
}
siquad3[2][0]=(1/(it-1))*sumc(quadifobsemedcad3lamb12);

i=0;
for(o=1500;o<2000;++o)
{
quadifobsemedcad4lamb12[i][0]=(vetorlambd12[o][0]-
medobscad4lamb12)^2;
++i;
}
siquad3[3][0]=(1/(it-1))*sumc(quadifobsemedcad4lamb12);

i=0;
for(o=2000;o<2500;++o)
{
quadifobsemedcad5lamb12[i][0]=(vetorlambd12[o][0]-
medobscad5lamb12)^2;
++i;
}
siquad3[4][0]=(1/(it-1))*sumc(quadifobsemedcad5lamb12);

varsiquad3=varc(siquad3)*5/4;
Dlamb12=(1/m)*sumc(siquad3);

medobscad1lamb12=medobscad1lamb12|medobscad2lamb12|
medobscad3lamb12|medobscad4lamb12|medobscad5lamb12;
quadifmedlamb12=(medobscad1lamb12-medtotlamb12).^2;
Elamb12=(it/(m-1))*sumc(quadifmedlamb12);
quadmedobscad1lamb12=quadmedobscad1lamb12|quadmedobscad2lamb12
|quadmedobscad3lamb12|quadmedobscad4lamb12|
quadmedobscad5lamb12;
matriz3a=siquad3~medobscad1lamb12;
matriz3b=siquad3~quadmedobscad1lamb12;
covar3a=variance(matriz3a);
covar3b=variance(matriz3b);
cov3a=covar3a[1][0]*5/4;
cov3b=covar3b[1][0]*5/4;
varV3=((it-1)/it)^2*(1/m)*varsiquad3+((m+1)/(m*it))^2*
(2/(m-1))*Elamb12^2+2*((m+1)*(it-1))/(m*it^2))
*(it/m)*(cov3b-2*medtotlamb12*cov3a);

sigma23=((it-1)/it)*Dlamb12+Elamb12/it;
V3=sigma23+Elamb12/(m*it);
df3=(2*V3^2)/varV3;
Rlamb12=sqrt((V3/Dlamb12)*(df3/(df3-2)));
R3[b][0]=Rlamb12;

/*s1*/
medtots1=meanc(vetors1);

medobscad1s1=((vetor10000')*vetors1)/it;
medobscad2s1=((vetor01000')*vetors1)/it;
medobscad3s1=((vetor00100')*vetors1)/it;
medobscad4s1=((vetor00010')*vetors1)/it;
medobscad5s1=((vetor00001')*vetors1)/it;

quadmedobscad1s1=medobscad1s1^2;
quadmedobscad2s1=medobscad2s1^2;
quadmedobscad3s1=medobscad3s1^2;
quadmedobscad4s1=medobscad4s1^2;
quadmedobscad5s1=medobscad5s1^2;

o=0;i=0;
for(o=0;o<500;++o)
{
quadifobsemedcad1s1[i][0]=(vetors1[o][0]-
medobscad1s1)^2;
++i;
}
siquad4[0][0]=(1/(it-1))*sumc(quadifobsemedcad1s1);

i=0;
for(o=500;o<1000;++o)
{
quadifobsemedcad2s1[i][0]=(vetors1[o][0]-
medobscad2s1)^2;
++i;
}
siquad4[1][0]=(1/(it-1))*sumc(quadifobsemedcad2s1);

i=0;
for(o=1000;o<1500;++o)
{
quadifobsemedcad3s1[i][0]=(vetors1[o][0]-
medobscad3s1)^2;
++i;
}
siquad4[2][0]=(1/(it-1))*sumc(quadifobsemedcad3s1);

```

```

i=0;
for(o=1500;o<2000;++o)
{
quadifobsemedcad4s1[i][0]=(vetors1[o][0]-
medobscad4s1)^2;
++i;
}
siquad4[3][0]=(1/(it-1))*sumc(quadifobsemedcad4s1);

i=0;
for(o=2000;o<2500;++o)
{
quadifobsemedcad5s1[i][0]=(vetors1[o][0]-
medobscad5s1)^2;
++i;
}
siquad4[4][0]=(1/(it-1))*sumc(quadifobsemedcad5s1);
varsiquad4=varc(siquad4)*5/4;
Ds1=(1/m)*sumc(siquad4);

medobscads1=medobscad1s1|medobscad2s1|medobscad3s1|
medobscad4s1|medobscad5s1;
quadifmeds1=(medobscads1-medlots1).^2;
Es1=(it/(m-1))*(sumc(quadifmeds1));
quadmedobscads1=quadmedobscad1s1|quadmedobscad2s1|
quadmedobscad3s1|quadmedobscad4s1|quadmedobscad5s1;
matriz4a=siquad4-medobscads1;
matriz4b=siquad4-quadmedobscads1;
covar4a=variance(matriz4a);
covar4b=variance(matriz4b);
cov4a=covar4a[1][0]*5/4;
cov4b=covar4b[1][0]*5/4;
varV4=((it-1)/it)^2*(1/m)*varsiquad4+((m+1)/(m*it))^2*
(2/(m-1))*Es1^2+2*((m+1)*(it-1)/(m*it^2))*(it/m)*
(cov4b-2*medlots1*cov4a);
sigma24=((it-1)/it)*Ds1+Es1/it;
V4=sigma24+Es1/(m*it);
df4=(2*V4^2)/varV4;
Rs1=sqrt((V4/Ds1)*(df4/(df4-2)));
R4[b][0]=Rs1;

/*s2*/
medlots2=meanc(vetors2);

medobscad1s2=((vetor10000')*vetors2)/it;
medobscad2s2=((vetor01000')*vetors2)/it;
medobscad3s2=((vetor00100')*vetors2)/it;
medobscad4s2=((vetor00010')*vetors2)/it;
medobscad5s2=((vetor00001')*vetors2)/it;

quadmedobscad1s2=medobscad1s2^2;
quadmedobscad2s2=medobscad2s2^2;
quadmedobscad3s2=medobscad3s2^2;
quadmedobscad4s2=medobscad4s2^2;
quadmedobscad5s2=medobscad5s2^2;

o=0;i=0;
for(o=0;o<500;++o)
{
quadifobsemedcad1s2[i][0]=(vetors2[o][0]-
medobscad1s2)^2;
++i;
}
siquad5[0][0]=(1/(it-1))*sumc(quadifobsemedcad1s2);

i=0;
for(o=500;o<1000;++o)
{
quadifobsemedcad2s2[i][0]=(vetors2[o][0]-
medobscad2s2)^2;
++i;
}
siquad5[1][0]=(1/(it-1))*sumc(quadifobsemedcad2s2);

i=0;
for(o=1000;o<1500;++o)
{
quadifobsemedcad3s2[i][0]=(vetors2[o][0]-
medobscad3s2)^2;
++i;
}
siquad5[2][0]=(1/(it-1))*sumc(quadifobsemedcad3s2);

i=0;
for(o=1500;o<2000;++o)
{
quadifobsemedcad4s2[i][0]=(vetors2[o][0]-
medobscad4s2)^2;
++i;
}
siquad5[3][0]=(1/(it-1))*sumc(quadifobsemedcad4s2);

i=0;
for(o=2000;o<2500;++o)
{
quadifobsemedcad5s2[i][0]=(vetors2[o][0]-

```

```

medobscad5s2)^2;
++i;
}
}
siquad5[i]{0}=(1/(it-1))*sumc(quadifobsmedcad5s2);

varsiquad5=varc(siquad5)*b/4;
Ds2=(1/m)*sumc(siquad5);

medobscads2=medobscad1s2|medobscad2s2|medobscad3s2|
medobscad4s2|medobscad5s2;
quadifmeds2=(medobscads2-medtot.s2).^2;
Es2={it/(m-1)}*sumc(quadifmeds2);
quadmedobscads2=quadmedobscad1s2|quadmedobscad2s2|
quadmedobscad3s2|quadmedobscad4s2|quadmedobscad5s2;
matriz5a=siquad5~medobscads2;
matriz5b=siquad5~quadmedobscads2;
covar5a=variance(matriz5a);
covar5b=variance(matriz5b);
cov5a=covar5a[1][0]*5/4;
cov5b=covar5b[1][0]*5/4;
varV5={(it-1)/it}^2*(1/m)*varsiquad5+{(m+1)/(m*it)}^2*
(2/(m-1))*Es2^2+2*{(m+1)*(it-1)/(m*it^2)}*(it/m)*
(cov5b-2*medtot.s2*cov5a);
sigma25=(it-1)/it*Ds2+Es2/it;
V5=sigma25+Es2/(m*it);
df5=(2*V5^2)/varV5;
Rs2=sqrt((V5/Ds2)*(dfb/(df5-2)));
R5[b]{0}=Rs2;

if{(R1[b]{0}<=1.004) && (R1[b]{0}>=0.990) &&
(R2[b]{0}<=1.004) && (R2[b]{0}>=0.990) &&
(R3[b]{0}<=1.004) && (R3[b]{0}>=0.990) &&
(R4[b]{0}<=1.004) && (R4[b]{0}>=0.990) &&
(R5[b]{0}<=1.004) && (R5[b]{0}>=0.990) &&
(mediavetorlambda1ok[b]{0}<1) &&
(mediavetorlambda2ok[b]{0}<1) &&
(mediavetorlambda12ok[b]{0}<2) &&
(mediavetors1ok[b]{0}<5) &&
(mediavetors2ok[b]{0}<5) &&
(medianavetorlambda1ok[b]{0}<1) &&
(medianavetorlambda2ok[b]{0}<1) &&
(medianavetorlambda12ok[b]{0}<2) &&
(medianavetors1ok[b]{0}<5) &&
(medianavetors2ok[b]{0}<5)}
{
b=b+1;
}

```

```

}
}
} // fecha b

for (i=0;i<u;+i)
{
eqmlambda1[i]{0}=varlambda1[i]{0}+(lambda1[i]{0}-0.1)^2;
eqmlambda2[i]{0}=varlambda2[i]{0}+(lambda2[i]{0}-0.1)^2;
eqmlambda12[i]{0}=varlambda12[i]{0}+(lambda12[i]{0}-0.2)^2;
eqns1[i]{0}=vars1[i]{0}+(s1[i]{0}-0.5)^2;
eqns2[i]{0}=vars2[i]{0}+(s2[i]{0}-0.5)^2;
eqmmedialambda1[i]{0}=varvetorlambda1ok[i]{0}+
(mediavetorlambda1ok[i]{0}-0.1)^2;
eqmmedialambda2[i]{0}=varvetorlambda2ok[i]{0}+
(mediavetorlambda2ok[i]{0}-0.1)^2;
eqmmedialambda12[i]{0}=varvetorlambda12ok[i]{0}+
(mediavetorlambda12ok[i]{0}-0.2)^2;
eqmmedias1[i]{0}=varvetors1ok[i]{0}+
(mediavetors1ok[i]{0}-0.5)^2;
eqmmedias2[i]{0}=varvetors2ok[i]{0}+
(mediavetors2ok[i]{0}-0.5)^2;
eqmmedialambda1[i]{0}=varvetorlambda1ok[i]{0}+
(medianavetorlambda1ok[i]{0}-0.1)^2;
eqmmedialambda2[i]{0}=varvetorlambda2ok[i]{0}+
(medianavetorlambda2ok[i]{0}-0.1)^2;
eqmmedialambda12[i]{0}=varvetorlambda12ok[i]{0}+
(medianavetorlambda12ok[i]{0}-0.2)^2;
eqmmedianas1[i]{0}=varvetors1ok[i]{0}+
(medianavetors1ok[i]{0}-0.5)^2;
eqmmedianas2[i]{0}=varvetors2ok[i]{0}+
(medianavetors2ok[i]{0}-0.5)^2;
}

/*intervalos de confianca 90%*/
for (i=0;i<u;+i)
{
liminflambda190[i]{0}=lambda1[i]{0}-1.6449*dplambda1[i]{0};
limsuplambda190[i]{0}=lambda1[i]{0}+1.6449*dplambda1[i]{0};

liminflambda290[i]{0}=lambda2[i]{0}-1.6449*dplambda2[i]{0};
limsuplambda290[i]{0}=lambda2[i]{0}+1.6449*dplambda2[i]{0};

liminflambda1290[i]{0}=lambda12[i]{0}-
1.6449*dplambda12[i]{0};
limsuplambda1290[i]{0}=lambda12[i]{0}+
1.6449*dplambda12[i]{0};
}

```

```

liminfs190[i][0]=s11[i][0]-1.6449*dps1[i][0];
limsup190[i][0]=s11[i][0]+1.6449*dps1[i][0];

liminfs290[i][0]=s12[i][0]-1.6449*dps2[i][0];
limsup290[i][0]=s12[i][0]+1.6449*dps2[i][0];
}

/* intervalos de confiança 95%*/
for (i=0;i<u; ++i)
{
liminflambda195[i][0]=lamb1[i][0]-1.96*dplambda1[i][0];
limsuplambda195[i][0]=lamb1[i][0]+1.96*dplambda1[i][0];

liminflambda295[i][0]=lamb2[i][0]-1.96*dplambda2[i][0];
limsuplambda295[i][0]=lamb2[i][0]+1.96*dplambda2[i][0];

liminflambda1295[i][0]=lamb12[i][0]-
1.96*dplambda12[i][0];
limsuplambda1295[i][0]=lamb12[i][0]+1.96*dplambda12[i][0];

liminfs195[i][0]=s11[i][0]-1.96*dps1[i][0];
limsup195[i][0]=s11[i][0]+1.96*dps1[i][0];

liminfs295[i][0]=s12[i][0]-1.96*dps2[i][0];
limsup295[i][0]=s12[i][0]+1.96*dps2[i][0];
}

/* cobertura 90%*/
decl quantlambda190=0;
decl quantlambda290=0;
decl quantlambda1290=0;
decl quants190=0;
decl quants290=0;
decl cobertura90lambda1, cobertura90lambda2,
cobertura90lambda12, cobertura90s1;
decl cobertura90s2;
decl k;
for (k=0;k<u; ++k)
{
if ((liminflambda190[k][0]>0.1 || limsuplambda190[k][0]<0.1))
quantlambda190=quantlambda190+1;

if ((liminflambda290[k][0]>0.1 || limsuplambda290[k][0]<0.1))
quantlambda290=quantlambda290+1;

if ((liminflambda1290[k][0]>0.2 || limsuplambda1290[k][0]<0.2))
quantlambda1290=quantlambda1290+1;

if ((liminfs190[k][0]>0.5 || limsup190[k][0]<0.5))
quants190=quants190+1;

if ((liminfs290[k][0]>0.5 || limsup290[k][0]<0.5))
quants290=quants290+1;
}

cobertura90lambda1=(500-quantlambda190)/500;
cobertura90lambda2=(500-quantlambda290)/500;
cobertura90lambda12=(500-quantlambda1290)/500;
cobertura90s1=(500-quants190)/500;
cobertura90s2=(500-quants290)/500;

/* cobertura 95%*/
decl quantlambda195=0;
decl quantlambda295=0;
decl quantlambda1295=0;
decl quants195=0;
decl quants295=0;
decl cobertura95lambda1, cobertura95lambda2,
cobertura95lambda12, cobertura95s1;
decl cobertura95s2;

for (k=0;k<u; ++k)
{
if ((liminflambda195[k][0]>0.1 ||
limsuplambda195[k][0]<0.1))
quantlambda195=quantlambda195+1;

if ((liminflambda295[k][0]>0.1 ||
limsuplambda295[k][0]<0.1))
quantlambda295=quantlambda295+1;

if ((liminflambda1295[k][0]>0.2 ||
limsuplambda1295[k][0]<0.2))
quantlambda1295=quantlambda1295+1;

if ((liminfs195[k][0]>0.5 || limsup195[k][0]<0.5))
quants195=quants195+1;

if ((liminfs295[k][0]>0.5 || limsup295[k][0]<0.5))
quants295=quants295+1;
}

cobertura95lambda1=(500-quantlambda195)/500;
cobertura95lambda2=(500-quantlambda295)/500;
cobertura95lambda12=(500-quantlambda1295)/500;
cobertura95s1=(500-quants195)/500;
cobertura95s2=(500-quants295)/500;

```



```

/* cobertura 95%*/
decl quantl195b=0;
decl quantl295b=0;
decl quantl1295b=0;
decl quants195b=0;
decl quants295b=0;
decl cobertura95blambda1, cobertura95blambda2,
cobertura95blambda12, cobertura95bs1;
decl cobertura95bs2;

for (k=0;k<u;++k)
{
  if ((ic95inf11[k][0]>0.1|| ic95sup11[k][0]<0.1))
    quantl195b=quantl195b+1;

  if ((ic95inf12[k][0]>0.1|| ic95sup12[k][0]<0.1))
    quantl295b=quantl295b+1;

  if ((ic95inf112[k][0]>0.2|| ic95sup112[k][0]<0.2))
    quantl1295b=quantl1295b+1;

  if ((ic95infs1[k][0]>0.5|| ic95sups1[k][0]<0.5))
    quants195b=quants195b+1;

  if ((ic95infs2[k][0]>0.5|| ic95sups2[k][0]<0.5))
    quants295b=quants295b+1;
}

cobertura95blambda1=(500-quantl195b)/500;
cobertura95blambda2=(500-quantl295b)/500;
cobertura95blambda12=(500-quantl1295b)/500;
cobertura95bs1=(500-quants195b)/500;
cobertura95bs2=(500-quants295b)/500;

/* cobertura 90%*/
decl quantl190b=0;
decl quantl290b=0;
decl quantl1290b=0;
decl quants190b=0;
decl quants290b=0;
decl cobertura90blambda1, cobertura90blambda2,
cobertura90blambda12, cobertura90bs1;
decl cobertura90bs2;

for (k=0;k<u;++k)
{
  if ((ic90inf11[k][0]>0.1|| ic90sup11[k][0]<0.1))
    quantl190b=quantl190b+1;

  if ((ic90inf12[k][0]>0.1|| ic90sup12[k][0]<0.1))
    quantl290b=quantl290b+1;

  if ((ic90inf112[k][0]>0.2|| ic90sup112[k][0]<0.2))
    quantl1290b=quantl1290b+1;

  if ((ic90infs1[k][0]>0.5|| ic90sups1[k][0]<0.5))
    quants190b=quants190b+1;

  if ((ic90infs2[k][0]>0.5|| ic90sups2[k][0]<0.5))
    quants290b=quants290b+1;
}

cobertura90blambda1=(500-quantl190b)/500;
cobertura90blambda2=(500-quantl290b)/500;
cobertura90blambda12=(500-quantl1290b)/500;
cobertura90bs1=(500-quants190b)/500;
cobertura90bs2=(500-quants290b)/500;

mediaemvlambda1=meanc(lamb1);
mediaemvlambda2=meanc(lamb2);
mediaemvlambda12=meanc(lamb12);
mediaemvs1=meanc(s11);
mediaemvs2=meanc(s12);

mediavaremvlambda1=meanc(varlambda1);
mediavaremvlambda2=meanc(varlambda2);
mediavaremvlambda12=meanc(varlambda12);
mediavaremvvs1=meanc(vars1);
mediavaremvvs2=meanc(vars2);

mediadpemvlambda1=meanc(dplambda1);
mediadpemvlambda2=meanc(dplambda2);
mediadpemvlambda12=meanc(dplambda12);
mediadpemvs1=meanc(dps1);
mediadpemvs2=meanc(dps2);

mediamediaslamb1=meanc(mediavetorlambda1ok);
mediamediaslamb2=meanc(mediavetorlambda2ok);
mediamediaslamb12=meanc(mediavetorlambda12ok);
mediamediasvs1=meanc(mediavetors1ok);
mediamediasvs2=meanc(mediavetors2ok);

mediavarslamb1=meanc(varvetorlambda1ok);
mediavarslamb2=meanc(varvetorlambda2ok);
mediavarslamb12=meanc(varvetorlambda12ok);

```

```

mediavarss1=meanc(varvetors1ok);
mediavarss2=meanc(varvetors2ok);

mediadplamb1=meanc(dpvetorlambda1ok);
mediadplamb2=meanc(dpvetorlambda2ok);
mediadplamb12=meanc(dpvetorlambda12ok);
mediadps1=meanc(dpvetors1ok);
mediadps2=meanc(dpvetors2ok);

mediamedianalamb1=meanc(medianavetorlambda1ok);
mediamedianalamb2=meanc(medianavetorlambda2ok);
mediamedianalamb12=meanc(medianavetorlambda12ok);
mediamedianas1=meanc(medianavetors1ok);
mediamedianas2=meanc(medianavetors2ok);

mediaeqmlambda1=meanc(eqmlambda1);
mediaeqmlambda2=meanc(eqmlambda2);
mediaeqmlambda12=meanc(eqmlambda12);
mediaeqms1=meanc(eqms1);
mediaeqms2=meanc(eqms2);

mediaeqmmedialambda1=meanc(eqmmedialambda1);
mediaeqmmedialambda2=meanc(eqmmedialambda2);
mediaeqmmedialambda12=meanc(eqmmedialambda12);
mediaeqmmedias1=meanc(eqmmedias1);
mediaeqmmedias2=meanc(eqmmedias2);

mediaeqmmedianalambda1=meanc(eqmmedianalambda1);
mediaeqmmedianalambda2=meanc(eqmmedianalambda2);
mediaeqmmedianalambda12=meanc(eqmmedianalambda12);
mediaeqmmedianas1=meanc(eqmmedianas1);
mediaeqmmedianas2=meanc(eqmmedianas2);

//SAIDA
print ("\n\n cobertura 90% lambda1: " ,cobertura90lambda1);
print ("\n\n cobertura 90% lambda2: " ,cobertura90lambda2);
print ("\n\n cobertura 90% lambda12: " ,cobertura90lambda12);
print ("\n\n cobertura 90% s1: " ,cobertura90s1);
print ("\n\n cobertura 90% s2: " ,cobertura90s2);

print ("\n\n cobertura 95% lambda1: " ,cobertura95lambda1);
print ("\n\n cobertura 95% lambda2: " ,cobertura95lambda2);
print ("\n\n cobertura 95% lambda12: " ,cobertura95lambda12);
print ("\n\n cobertura 95% s1: " ,cobertura95s1);
print ("\n\n cobertura 95% s2: " ,cobertura95s2);

print ("\n\n cobertura 90%b lambda1: " ,cobertura90blambda1);
print ("\n\n cobertura 90%b lambda2: " ,cobertura90blambda2);

print ("\n\n cobertura 90%b lambda1: "
,cobertura90blambda1);
print ("\n\n cobertura 90%b s1: " ,cobertura90bs1);
print ("\n\n cobertura 90%b s2: " ,cobertura90bs2);

print ("\n\n cobertura 95%b lambda1: "
,cobertura95blambda1);
print ("\n\n cobertura 95%b lambda2: "
,cobertura95blambda2);
print ("\n\n cobertura 95%b lambda12: "
,cobertura95blambda12);
print ("\n\n cobertura 95%b s1: " ,cobertura95bs1);
print ("\n\n cobertura 95%b s2: " ,cobertura95bs2);

print("\n\n estimativas de máxima verossimilhança
lambda1:", lamb1);
print("\n estimativas de máxima verossimilhança lambda2:",
lamb2);
print("\n estimativas de máxima verossimilhança lambda12:",
lamb12);
print("\n estimativas de máxima verossimilhança s1:", s1);
print("\n estimativas de máxima verossimilhança s2:", s2);

print("\n\n var estimativas de máxima verossimilhança
lambda1:", varlambda1);
print("\n var estimativas de máxima verossimilhança
lambda2:", varlambda2);
print("\n var estimativas de máxima verossimilhança
lambda12:", varlambda12);
print("\n var estimativas de máxima verossimilhança s1:",
vars1);
print("\n var estimativas de máxima verossimilhança s2:",
vars2);

print("\n\n dp estimativas de máxima verossimilhança
lambda1:", dplambda1);
print("\n dp estimativas de máxima verossimilhança
lambda2:", dplambda2);
print("\n dp estimativas de máxima verossimilhança
lambda12:", dplambda12);
print("\n dp estimativas de máxima verossimilhança s1:",
dps1);
print("\n dp estimativas de máxima verossimilhança s2:",
dps2);

print ("\n\n IC 95% para
lambda1:",liminflambda195,limsuplambda195);

```

```

print ("\n IC 95% para
lambda2:", liminflambda295, limsuplambda295);
print ("\n IC 95% para
lambda12:", liminflambda1295, limsuplambda1295);
print ("\n IC 95% para s1:", liminfs195, limsups195);
print ("\n IC 95% para s2:", liminfs295, limsups295);

print ("\n\n IC 90% para
lambda1:", liminflambda190, limsuplambda190);
print ("\n IC 90% para
lambda2:", liminflambda290, limsuplambda290);
print ("\n IC 90% para
lambda12:", liminflambda1290, limsuplambda1290);
print ("\n IC 90% para s1:", liminfs190, limsups190);
print ("\n IC 90% para s2:", liminfs290, limsups290);

print ("\n\n indices de convergencia lambda1:", R1);
print ("\n indices de convergencia lambda2:", R2);
print ("\n indices de convergencia lambda12:", R3);
print ("\n indices de convergencia s1:", R4);
print ("\n indices de convergencia s2:", R5);

print ("\n\n media lambda1:", mediavetorlambda1ok);
print ("\n media lambda2:", mediavetorlambda2ok);
print ("\n media lambda12:", mediavetorlambda12ok);
print ("\n media s1:", mediavetors1ok);
print ("\n media s2:", mediavetors2ok);

print ("\n\n var lambda1:", varvetorlambda1ok);
print ("\n var lambda2:", varvetorlambda2ok);
print ("\n var lambda12:", varvetorlambda12ok);
print ("\n var s1:", varvetors1ok);
print ("\n var s2:", varvetors2ok);

print ("\n\n dp lambda1:", dpvetorlambda1ok);
print ("\n dp lambda2:", dpvetorlambda2ok);
print ("\n dp lambda12:", dpvetorlambda12ok);
print ("\n dp s1:", dpvetors1ok);
print ("\n dp s2:", dpvetors2ok);

print ("\n\n mediana lambda1:", medianavetorlambda1ok);
print ("\n mediana lambda2:", medianavetorlambda2ok);
print ("\n mediana lambda12:", medianavetorlambda12ok);
print ("\n mediana s1:", medianavetors1ok);
print ("\n mediana s2:", medianavetors2ok);

print ("\n\n media das cmv lambda1:", mediaemvlambda1);
print ("\n\n media das emv lambda2:", mediaemvlambda2);
print ("\n\n media das emv lambda12:", mediaemvlambda12);
print ("\n\n media das emv s1:", mediaemvs1);
print ("\n\n media das emv s2:", mediaemvs2);

print ("\n\n media das vars emv lambda1:",
mediavaremvlambda1);
print ("\n\n media das vars emv lambda2:",
mediavaremvlambda2);
print ("\n\n media das vars emv lambda12:",
mediavaremvlambda12);
print ("\n\n media das vars emv s1:", mediavaremvvs1);
print ("\n\n media das vars emv s2:", mediavaremvvs2);

print ("\n\n media dp emv lambda1:", mediadpemvlambda1);
print ("\n\n media dp emv lambda2:", mediadpemvlambda2);
print ("\n\n media dp emv lambda12:", mediadpeavlambda12);
print ("\n\n media dp emv s1:", mediadpeavvs1);
print ("\n\n media dp emv s2:", mediadpemvs2);

print ("\n\n media das medias lambda1:", mediamediaslamb1);
print ("\n media das medias lambda2:", mediamediaslamb2);
print ("\n media das medias lambda12:", mediamediaslamb12);
print ("\n media das medias s1:", mediamediasvs1);
print ("\n media das medias s2:", mediamediasvs2);

print ("\n\n media das vars lambda1:", mediavarslamb1);
print ("\n media das vars lambda2:", mediavarslamb2);
print ("\n\n media das vars lambda12:", mediavarslamb12);
print ("\n media das vars s1:", mediavarvs1);
print ("\n media das vars s2:", mediavarvs2);

print ("\n\n media dp lambda1:", mediadplamb1);
print ("\n media dp lambda2:", mediadplamb2);
print ("\n media dp lambda12:", mediadplamb12);
print ("\n media dp s1:", mediadps1);
print ("\n media dp s2:", mediadps2);

print ("\n\n media das medianas lambda1:",
mediamedianalamb1);
print ("\n media das medianas lambda2:", mediamedianalamb2);
print ("\n\n media das medianas lambda12:",
mediamedianalamb12);
print ("\n media das medianas s1:", mediamedianasvs1);
print ("\n media das medianas s2:", mediamedianasvs2);

```

```

print ("\n\n intervalo de credibilidade 90% lambda1 :",
      ic90infi1, ic90sup1);
print ("\n\n intervalo de credibilidade 90% lambda2:",
      ic90infi2, ic90sup2);
print ("\n\n intervalo de credibilidade 90% lambda12:",
      ic90infi12, ic90sup12);
print ("\n\n intervalo de credibilidade 90% s1:", ic90infs1,
      ic90sups1);
print ("\n\n intervalo de credibilidade 90% s2:", ic90infs2,
      ic90sups2);

print ("\n\n intervalo de credibilidade 95% lambda1 :",
      ic95infi1, ic95sup1);
print ("\n\n intervalo de credibilidade 95% lambda2:",
      ic95infi2, ic95sup2);
print ("\n\n intervalo de credibilidade 95% lambda12:",
      ic95infi12, ic95sup12);
print ("\n\n intervalo de credibilidade 95% s1:", ic95infs1,
      ic95sups1);
print ("\n\n intervalo de credibilidade 95% s2:", ic95infs2,
      ic95sups2);

print ("\n\n media eqm das emv lambda1:", mediaeqmlambda1);
print ("\n\n media eqm das emv lambda2:", mediaeqmlambda2);
print ("\n\n media eqm das emv lambda12:", mediaeqmlambda12);
print ("\n\n media eqm das emv s1:", mediaeqms1);
print ("\n\n media eqm das emv s2:", mediaeqms2);

print ("\n\n media eqm das medias lambda1:",
      mediaeqmmedialambda1);
print ("\n\n media eqm das medias lambda2:", mediaeqmmedialambda2);
print ("\n\n media eqm das medias lambda12:",
      mediaeqmmedialambda12);
print ("\n\n media eqm das medias s1:", mediaeqmmedias1);
print ("\n\n media eqm das medias s2:", mediaeqmmedias2);
print ("\n\n media eqm das medianas lambda1:",
      mediaeqmmedianalambda1);
print ("\n\n media eqm das medianas lambda2:",
      mediaeqmmedianalambda2);
print ("\n\n media eqm das medianas lambda12:",
      mediaeqmmedianalambda12);
print ("\n\n media eqm das medianas s1:", mediaeqmmedianas1);
print ("\n\n media eqm das medianas s2:", mediaeqmmedianas2);

print ("\n\n tempo de execucao:", timespan{time});
} // fecha main

```

Apêndice F

Programa Computacional Implementado para Estimação dos Parâmetros da da Distribuição EBVE considerando Tempos Acelerados

```

/* Com este programa obtivemos os resultados das tabelas 4.1,
4.8, 4.10, 4.11, 4.12, 4.13*/

#include <cxstd.h>
#import <maximize>
#import <oxprob>

decl n1=260; // no. de elementos na voltagem v1
decl n2=150; // no. de elementos na voltagem v2
decl n3=100; // no. de elementos na voltagem v3
decl n=510; // tamanho das amostras

decl i,j,k,t1g,t1,t2,b,b,tempo1,tempo2,V;

/*-----parâmetros na voltagem usual-----*/
lambda1usual(const vP, const f1, const avScore, const amHess)
{
  V=1.5;
  f1[0]=vP[0][0]*(V^vP[5][0]);
  return 1;
}

lambda2usual(const vP, const f2, const avScore, const amHess)
{
  V=1.5;
  f2[0]=vP[1][0]*(V^vP[6][0]);
  return 1;
}

lambda12usual(const vP, const f3, const avScore, const amHess)
{
  V=1.5;
  f3[0]=vP[2][0]*(V^vP[7][0]);
  return 1;
}

slusual(const vP, const f4, const avScore, const amHess)
{
  V=1.5;
  f4[0]=vP[3][0]*(V^vP[7][0]);
  return 1;
}

s2usual(const vP, const f5, const avScore, const amHess)
{
  V=1.5;
  f5[0]=vP[4][0]*(V^vP[7][0]);
  return 1;
}
/*-----fim parâmetros na voltagem usual-----*/

/*-----geração de (t1,t2) pelo método da rejeição -----*/
gera()
{
  decl vP=<0.0296296296296;
    0.0296296296296;
    0.0592592592593;
    0.148148148148;
    0.148148148148;
    3;
    3;
    3>;
  decl u1v1,u2v1,u3v1,berv1,d1v1,t1aux1v1,t2aux1v1;
  decl u1v2,u2v2,u3v2,berv2,d1v2,t1aux1v2,t2aux1v2;
  decl u1v3,u2v3,u3v3,berv3,d1v3,t1aux1v3,t2aux1v3;
  decl tempo1v1=zeros(n1,1);
  decl tempo2v1=zeros(n1,1);
  decl tempo1v2=zeros(n2,1);
  decl tempo2v2=zeros(n2,1);
  decl tempo1v3=zeros(n3,1);
  decl tempo2v3=zeros(n3,1);
  j=0;
  while(j<n1) // gera tempos na voltagem v1
  {
    V=2;
    u1v1=ranu(1,1)*10;
    u2v1=ranu(1,1)*10;
    u3v1=ranu(1,1)*0.26;
    berv1=ranbinomial(1,1,1,0.5);
    if(berv1==0)
    {
      t1aux1v1=min(u1v1,u2v1);

```



```

t2aux1v1=max(u1v1,u2v1);
d1v1=dbt1met2(t1aux1v1,t2aux1v1,vP);
if(u3v1<d1v1)
{
tempo1v1[j][0]=t1aux1v1;
tempo2v1[j][0]=t2aux1v1;
j=j+1;
}
}
else
{
t1aux1v1=max(u1v1,u2v1);
t2aux1v1=min(u1v1,u2v1);
d1v1=dbt1mat2(t1aux1v1,t2aux1v1,vP);
if(u3v1<d1v1)
{
tempo1v1[j][0]=t1aux1v1;
tempo2v1[j][0]=t2aux1v1;
j=j+1;
}
}
}
j=0;
while(j<n2) // gera tempos na voltagem v2
{
V=2.5;
u1v2=ranu(1,1)*5;
u2v2=ranu(1,1)*5;
u3v2=ranu(1,1)*0.88;
berv2=ranbinomial(1,1,1,0.5);
if(berv2==0)
{
t1aux1v2=min(u1v2,u2v2);
t2aux1v2=max(u1v2,u2v2);
d1v2=dbt1met2(t1aux1v2,t2aux1v2,vP);
if(u3v2<d1v2)
{
tempo1v2[j][0]=t1aux1v2;
tempo2v2[j][0]=t2aux1v2;
j=j+1;
}
}
}
else
{
t1aux1v2=max(u1v2,u2v2);
t2aux1v2=min(u1v2,u2v2);
d1v2=dbt1mat2(t1aux1v2,t2aux1v2,vP);
if(u3v2<d1v2)
{
tempo1v2[j][0]=t1aux1v2;
tempo2v2[j][0]=t2aux1v2;
j=j+1;
}
}
}
}
j=0;
while(j<n3) // gera tempos na voltagem v3
{
V=3;
u1v3=ranu(1,1)*3;
u2v3=ranu(1,1)*3;
u3v3=ranu(1,1)*2.9;
berv3=ranbinomial(1,1,1,0.5);
if(berv3==0)
{
t1aux1v3=min(u1v3,u2v3);
t2aux1v3=max(u1v3,u2v3);
d1v3=dbt1met2(t1aux1v3,t2aux1v3,vP);
if(u3v3<d1v3)
{
tempo1v3[j][0]=t1aux1v3;
tempo2v3[j][0]=t2aux1v3;
j=j+1;
}
}
}
else
{
t1aux1v3=max(u1v3,u2v3);
t2aux1v3=min(u1v3,u2v3);
d1v3=dbt1mat2(t1aux1v3,t2aux1v3,vP);
if(u3v3<d1v3)
{
tempo1v3[j][0]=t1aux1v3;
tempo2v3[j][0]=t2aux1v3;
j=j+1;
}
}
}
}
return
{tempo1v1|tempo1v2|tempo1v3}~{tempo2v1|tempo2v2|tempo2v3}; //
amostra dos tempos
}
/*-----fim geração de (t1,t2) pelo método da rejeição -----*/

/*-----log da função de verossimilhança-----*/

```

```

logfveros(const vP, const l, const avScore, const amHess)
{
    decl logf=zeros(n,1);
    for(i=0;i<n1;++i)
        {V=2;
        if(t1[i][0]<t2[i][0])
            {
                logf[i][0]=logdbt1met2(t1[i][0],t2[i][0],vP);
            }
        else
            {
                logf[i][0]=logdbt1mat2(t1[i][0],t2[i][0],vP);
            }
        }
    for(i=n1;i<n1+n2;++i)
        {V=2.5;
        if(t1[i][0]<t2[i][0])
            {
                logf[i][0]=logdbt1met2(t1[i][0],t2[i][0],vP);
            }
        else
            {
                logf[i][0]=logdbt1mat2(t1[i][0],t2[i][0],vP);
            }
        }
    for(i=n1+n2;i<n1+n2+n3;++i)
        {V=3;
        if(t1[i][0]<t2[i][0])
            {
                logf[i][0]=logdbt1met2(t1[i][0],t2[i][0],vP);
            }
        else
            {
                logf[i][0]=logdbt1mat2(t1[i][0],t2[i][0],vP);
            }
        }
    l[0]=sumc(logf);
    if{vP[0][0]<0|| vP[1][0]<0|| vP[2][0]<0|| vP[3][0]<0||
        vP[4][0]<0}
        {
            return 0;
        }
    return 1;
}
/*-----fim log da função de verossimilhança-----*/

main()
{
    h=500;//total de amostras

    // declaração das variáveis
    decl e, e1, e2, e3, e4, e5, e6, e7, e8, r, q;
    e=0; e1=0; e2=0; e3=0; e4=0; e5=0; e6=0; e7=0; e8=0; r=0;
    q=0;
    decl vbeta01=0.0296296296296;
    decl vbeta02=0.0296296296296;
    decl vbeta03=0.0592592592593;
    decl vbeta04=0.148148148148;
    decl vbeta05=0.148148148148;
    decl vbeta1=3;
    decl vbeta2=3;
    decl vbeta3=3;
    decl vlambdal0=0.1;
    decl vlambdal20=0.1;
    decl vlambdal20=0.2;
    decl vs10=0.5;
    decl vs20=0.5;

    decl tlq=zeros(n,2);

    decl t1v1=zeros(n1,1);
    decl t2v1=zeros(n1,1);
    decl t1v2=zeros(n2,1);
    decl t2v2=zeros(n2,1);
    decl t1v3=zeros(n3,1);
    decl t2v3=zeros(n3,1);
    decl tlqv1=zeros(n1,2);
    decl tlqv2=zeros(n2,2);
    decl tlqv3=zeros(n3,2);

    decl beta01=zeros(h,1);
    decl beta02=zeros(h,1);
    decl beta03=zeros(h,1);
    decl beta04=zeros(h,1);
    decl beta05=zeros(h,1);
    decl beta1=zeros(h,1);
    decl beta2=zeros(h,1);
    decl beta3=zeros(h,1);
    decl lambdal0=zeros(h,1);
    decl lambdal20=zeros(h,1);
    decl lambdal20=zeros(h,1);
    decl s10=zeros(h,1);
    decl s20=zeros(h,1);

    decl beta01ord=zeros(h,1);

```

```

decl beta02ord=zeros(h,1);
decl beta03ord=zeros(h,1);
decl beta04ord=zeros(h,1);
decl beta05ord=zeros(h,1);
decl betalord=zeros(h,1);
decl beta2ord=zeros(h,1);
decl beta3ord=zeros(h,1);
decl lambda10ord=zeros(h,1);
decl lambda20ord=zeros(h,1);
decl lambda120ord=zeros(h,1);
decl s10ord=zeros(h,1);
decl s20ord=zeros(h,1);

decl avscore11=zeros(8,1);
decl avscore12=zeros(8,1);
decl avscore112=zeros(8,1);
decl avscores1=zeros(8,1);
decl avscores2=zeros(8,1);

decl varbeta01=zeros(h,1);
decl varbeta02=zeros(h,1);
decl varbeta03=zeros(h,1);
decl varbeta04=zeros(h,1);
decl varbeta05=zeros(h,1);
decl varbeta1=zeros(h,1);
decl varbeta2=zeros(h,1);
decl varbeta3=zeros(h,1);
decl varlambda10=zeros(h,1);
decl varlambda20=zeros(h,1);
decl varlambda120=zeros(h,1);
decl vars10=zeros(h,1);
decl vars20=zeros(h,1);

decl desvpbeta01=zeros(h,1);
decl desvpbeta02=zeros(h,1);
decl desvpbeta03=zeros(h,1);
decl desvpbeta04=zeros(h,1);
decl desvpbeta05=zeros(h,1);
decl desvpbeta1=zeros(h,1);
decl desvpbeta2=zeros(h,1);
decl desvpbeta3=zeros(h,1);
decl desvplambda10=zeros(h,1);
decl desvplambda20=zeros(h,1);
decl desvplambda120=zeros(h,1);
decl desvps10=zeros(h,1);
decl desvps20=zeros(h,1);

decl eqmbeta0i=zeros(h,1);

decl eqmbeta02=zeros(h,1);
decl eqmbeta03=zeros(h,1);
decl eqmbeta04=zeros(h,1);
decl eqmbeta05=zeros(h,1);
decl eqmbeta1=zeros(h,1);
decl eqmbeta2=zeros(h,1);
decl eqmbeta3=zeros(h,1);
decl eqmlambda10=zeros(h,1);
decl eqmlambda20=zeros(h,1);
decl eqmlambda120=zeros(h,1);
decl eqms10=zeros(h,1);
decl eqms20=zeros(h,1);

decl vetormediast1v1=zeros(h,1);
decl vetormediast2v1=zeros(h,1);
decl vetorvart1v1=zeros(h,1);
decl vetorvart2v1=zeros(h,1);
decl vetorcorrelacaov1=zeros(h,1);
decl mediat1v1, mediat2v1, vart1v1, vart2v1;
decl correlacaov1=zeros(2,2);
decl mediageralt1v1, mediageralt2v1, mediavart1v1,
mediavart2v1, mediacorr1v1;
decl desvpmediast1v1, desvpmediast2v1, desvpvart1v1,
desvpvart2v1, desvpcorr1v1;

decl vetormediast1v2=zeros(h,1);
decl vetormediast2v2=zeros(h,1);
decl vetorvart1v2=zeros(h,1);
decl vetorvart2v2=zeros(h,1);
decl vetorcorrelacaov2=zeros(h,1);
decl mediat1v2, mediat2v2, vart1v2, vart2v2;
decl correlacaov2=zeros(2,2);
decl mediageralt1v2, mediageralt2v2, mediavart1v2,
mediavart2v2, mediacorr2v2;
decl desvpmediast1v2, desvpmediast2v2, desvpvart1v2,
desvpvart2v2, desvpcorr2v2;

decl vetormediast1v3=zeros(h,1);
decl vetormediast2v3=zeros(h,1);
decl vetorvart1v3=zeros(h,1);
decl vetorvart2v3=zeros(h,1);
decl vetorcorrelacaov3=zeros(h,1);
decl mediat1v3, mediat2v3, vart1v3, vart2v3;
decl correlacaov3=zeros(2,2);
decl mediageralt1v3, mediageralt2v3, mediavart1v3,
mediavart2v3, mediacorr3v3;
decl desvpmediast1v3, desvpmediast2v3, desvpvart1v3,
desvpvart2v3, desvpcorr3v3;

```

```

decl MatInfObs=zeros(8,8);
decl invMatInfObs=zeros(8,8);
decl diag=zeros(8,1);
decl desvpad=zeros(8,1);
decl l,ir,mhess;
decl bb=zeros(8,8);

decl liminfbeta0190=zeros(h,1);
decl liminfbeta0290=zeros(h,1);
decl liminfbeta0390=zeros(h,1);
decl liminfbeta0490=zeros(h,1);
decl liminfbeta0590=zeros(h,1);
decl liminfbeta190=zeros(h,1);
decl liminfbeta290=zeros(h,1);
decl liminfbeta390=zeros(h,1);
decl liminflambda1090=zeros(h,1);
decl liminflambda2090=zeros(h,1);
decl liminflambda12090=zeros(h,1);
decl liminfs1090=zeros(h,1);
decl liminfs2090=zeros(h,1);

decl limsupbeta0190=zeros(h,1);
decl limsupbeta0290=zeros(h,1);
decl limsupbeta0390=zeros(h,1);
decl limsupbeta0490=zeros(h,1);
decl limsupbeta0590=zeros(h,1);
decl limsupbeta190=zeros(h,1);
decl limsupbeta290=zeros(h,1);
decl limsupbeta390=zeros(h,1);
decl limsuplambda1090=zeros(h,1);
decl limsuplambda2090=zeros(h,1);
decl limsuplambda12090=zeros(h,1);
decl limsupsl090=zeros(h,1);
decl limsupsl2090=zeros(h,1);

decl iiminfbeta0195=zeros(h,1);
decl iiminfbeta0295=zeros(h,1);
decl iiminfbeta0395=zeros(h,1);
decl iiminfbeta0495=zeros(h,1);
decl iiminfbeta0595=zeros(h,1);
decl iiminfbeta195=zeros(h,1);
decl iiminfbeta295=zeros(h,1);
decl iiminfbeta395=zeros(h,1);
decl iiminflambda1095=zeros(h,1);
decl iiminflambda2095=zeros(h,1);
decl iiminflambda12095=zeros(h,1);
decl iiminfs1095=zeros(h,1);

decl liminfs2095=zeros(h,1);
decl limsupbeta0195=zeros(h,1);
decl limsupbeta0295=zeros(h,1);
decl limsupbeta0395=zeros(h,1);
decl limsupbeta0495=zeros(h,1);
decl limsupbeta0595=zeros(h,1);
decl limsupbeta195=zeros(h,1);
decl limsupbeta295=zeros(h,1);
decl limsupbeta395=zeros(h,1);
decl limsuplambda1095=zeros(h,1);
decl limsuplambda2095=zeros(h,1);
decl limsuplambda12095=zeros(h,1);
decl limsupsl095=zeros(h,1);
decl limsupsl2095=zeros(h,1);

decl quantbeta0190=0;
decl quantbeta0290=0;
decl quantbeta0390=0;
decl quantbeta0490=0;
decl quantbeta0590=0;
decl quantbeta190=0;
decl quantbeta290=0;
decl quantbeta390=0;
decl quantlambda1090=0;
decl quantlambda2090=0;
decl quantlambda12090=0;
decl quants1090=0;
decl quants2090=0;

decl quantbeta0195=0;
decl quantbeta0295=0;
decl quantbeta0395=0;
decl quantbeta0495=0;
decl quantbeta0595=0;
decl quantbeta195=0;
decl quantbeta295=0;
decl quantbeta395=0;
decl quantlambda1095=0;
decl quantlambda2095=0;
decl quantlambda12095=0;
decl quants1095=0;
decl quants2095=0;

decl cobeturabeta0190=0;
decl cobeturabeta0290=0;
decl cobeturabeta0390=0;
decl cobeturabeta0490=0;

```

```

decl cobeturabeta0590=0;
decl cobeturabeta190=0;
decl cobeturabeta290=0;
decl cobeturabeta390=0;
decl cobeturalambda1090=0;
decl cobeturalambda2090=0;
decl cobeturalambda12090=0;
decl cobeturas1090=0;
decl cobeturas2090=0;

decl cobeturabeta0195=0;
decl cobeturabeta0295=0;
decl cobeturabeta0395=0;
decl cobeturabeta0495=0;
decl cobeturabeta0595=0;
decl cobeturabeta195=0;
decl cobeturabeta295=0;
decl cobeturabeta395=0;
decl cobeturalambda1095=0;
decl cobeturalambda2095=0;
decl cobeturalambda12095=0;
decl cobeturas1095=0;
decl cobeturas2095=0;

decl
  mediabeta01,mediabeta02,mediabeta03,mediabeta04,mediabeta05;
decl medianabeta1,medianabeta2,medianabeta3;
decl
  medialambda10,medialambda20,medialambda120,medias10,medias20;
decl
  medianabeta01,medianabeta02,medianabeta03,medianabeta04,media
  nabeta05;
decl medianabeta1,medianabeta2,medianabeta3;
decl
  medianalambda10,medianalambda20,medianalambda120,medianas10,m
  edianas20;

decl mediadpbeta01, mediadpbeta02, mediadpbeta03,
  mediadpbeta04, mediadpbeta05;
decl mediadpbeta1, mediadpbeta2,mediadpbeta3;
decl mediadplambda10, mediadplambda20, mediadplambda120,
  mediadps10, mediadps20;
decl mediaeqmbeta01, mediaeqmbeta02, mediaeqmbeta03,
  mediaeqmbeta04, mediaeqmbeta05;
decl mediaeqmbeta1, mediaeqmbeta2,mediaeqmbeta3;
decl mediaeqmlambda10, mediaeqmlambda20, mediaeqmlambda120,
  mediaeqms10, mediaeqms20;

decl
  desvpabeta01,desvpabeta02,desvpabeta03,desvpabeta04,desvpabet
  a05;
decl desvpabeta1, desvpabeta2,desvpabeta3;
decl
  desvpalambda10,desvpaiambda20,desvpalambda120,desvpas10,desvp
  as20;
decl dpdpbeta01, dpdpbeta02,dpdpbeta03,dpdpbeta04,dpdpbeta05;
decl dpdp1lambda10,
  dpdp1lambda20,dpdp1lambda120,dpdps10,dpdps20;
decl dpdpbeta1,dpdpbeta2,dpdpbeta3;
decl dpeqmbeta01, dpeqmbeta02, dpeqmbeta03, dpeqmbeta04,
  dpeqmbeta05;
decl dpeqmbeta1, dpeqmbeta2,dpeqmbeta3;
decl dpeqmlambda10, dpeqmlambda20, dpeqmlambda120, dpeqms10,
  dpeqms20;

decl time;
time=timer();

b=0;
while(b<h) obtenção das amostras
  {
  tlg=gera();
  t1=tlg*<1;0>;
  t2=tlg*<0;1>;
  for (i=0;i<n1;++i)
  {
  t1v1[i][0]=t1[i][0];
  t2v1[i][0]=t2[i][0];
  }
  tlgv1=t1v1-t2v1; // tempos na voltagem v1
  for (i=0;i<n2;++i)
  {
  t1v2[i][0]=t1[n1+i][0];
  t2v2[i][0]=t2[n1+i][0];
  }
  tlgv2=t1v2-t2v2; // tempos na voltagem v2
  for (i=0;i<n3;++i)
  {
  t1v3[i][0]=t1[n1+n2+i][0];
  t2v3[i][0]=t2[n1+n2+i][0];
  }
  tlgv3=t1v3-t2v3; // tempos na voltagem v3

  // calcula a média, a variância e a correlação dos tempos
  para cada amostra
  mediat1v1=mean(t1v1);
  
```



```

mediat2v1=meanc(t2v1);
vart1v1=varc(t1v1);
vart2v1=varc(t2v1);
correlacaov1=correlation(t1qv1);
vetormediast1v1[b][0]=mediat1v1;
vetormediast2v1[b][0]=mediat2v1;
vetorvart1v1[b][0]=vart1v1;
vetorvart2v1[b][0]=vart2v1;
vetorcorrelacaov1[b][0]=correlacaov1[1][0];

mediat1v2=meanc(t1v2);
mediat2v2=meanc(t2v2);
vart1v2=varc(t1v2);
vart2v2=varc(t2v2);
correlacaov2=correlation(t1qv2);
vetormediast1v2[b][0]=mediat1v2;
vetormediast2v2[b][0]=mediat2v2;
vetorvart1v2[b][0]=vart1v2;
vetorvart2v2[b][0]=vart2v2;
vetorcorrelacaov2[b][0]=correlacaov2[1][0];

mediat1v3=meanc(t1v3);
mediat2v3=meanc(t2v3);
vart1v3=varc(t1v3);
vart2v3=varc(t2v3);
correlacaov3=correlation(t1qv3);
vetormediast1v3[b][0]=mediat1v3;
vetormediast2v3[b][0]=mediat2v3;
vetorvart1v3[b][0]=vart1v3;
vetorvart2v3[b][0]=vart2v3;
vetorcorrelacaov3[b][0]=correlacaov3[1][0];

//maximização do log da verossimilhança
decl vP=<1;1;1;1;2;2;2>; // chutes iniciais
logfveros(vP,&l,0,0);
MaxControl(1000,1000);
mhess=0.00001*unit(8);
ir=MaxBFGS(logfveros,&vP,&l,&mhess,TRUE);
q=q+1;

if(ir==MAX_CONV)
    r=r+1;

if (ir==MAX_CONV && vP[0][0]<0.6)
    e1=e1+1;
if (ir==MAX_CONV && vP[1][0]<0.6)
    e2=e2+1;
if (ir==MAX_CONV && vP[2][0]<1.2)
    e3=e3+1;
if (ir==MAX_CONV && vP[3][0]<3)
    e4=e4+1;
if (ir==MAX_CONV && vP[4][0]<3)
    e5=e5+1;
if (ir==MAX_CONV && vP[5][0]>0)
    e6=e6+1;
if (ir==MAX_CONV && vP[6][0]>0)
    e7=e7+1;
if (ir==MAX_CONV && vP[7][0]>0)
    e8=e8+1;

if(ir==MAX_CONV && vP[0][0]<0.6 && vP[1][0]<0.6
&& vP[2][0]<1.2 && vP[3][0]<3 && vP[4][0]<3 && vP[5][0]>0
&& vP[6][0]>0 && vP[7][0]>0)
    {V=1.5;

/*estimativas de máxima verossimilhança*/
beta01[b][0]=vP[0][0];
beta02[b][0]=vP[1][0];
beta03[b][0]=vP[2][0];
beta04[b][0]=vP[3][0];
beta05[b][0]=vP[4][0];
beta1[b][0]=vP[5][0];
beta2[b][0]=vP[6][0];
beta3[b][0]=vP[7][0];
lambda10[b][0]=vP[0][0]*V^vP[5][0];
lambda20[b][0]=vP[1][0]*V^vP[6][0];
lambda120[b][0]=vP[2][0]*V^vP[7][0];
s10[b][0]=vP[3][0]*V^vP[7][0];
s20[b][0]=vP[4][0]*V^vP[7][0];

Num1Derivative(lambdalusual, vP, &avscore1);
Num1Derivative(lambda2usual, vP, &avscore2);
Num1Derivative(lambdal2usual, vP, &avscore12);
Num1Derivative(s1usual, vP, &avscore1);
Num1Derivative(s2usual, vP, &avscore2);

/*variâncias dos estimadores*/
Num2Derivative(logfveros,vP,&mhess);
MatInfObs=mhess;
invMatInfObs=invert(MatInfObs)*(-1);
bb=invMatInfObs;
e=e+1;

if (invMatInfObs[0][0]>0 && invMatInfObs[1][1]>0 &&
invMatInfObs[2][2]>0 &&

```

```

    invMatInfObs[3][3]>0 && invMatInfObs[4][4]>0 &&
    invMatInfObs[5][5]>0 &&
    invMatInfObs[6][6]>0 && invMatInfObs[7][7]>0)
    {
    varbeta01[b][0]=bb [0][0];
    varbeta02[b][0]=bb [1][1];
    varbeta03[b][0]=bb [2][2];
    varbeta04[b][0]=bb [3][3];
    varbeta05[b][0]=bb [4][4];
    varbeta1 [b][0]=bb[b][5];
    varbeta2[b][0]=bb[6][6];
    varbeta3[b][0]=bb[7][7];
    varlambda10[b][0]=(avscore11*bb*avscore11);
    varlambda20[b][0]=(avscore12*bb*avscore12);
    varlambda120[b][0]=(avscore12*bb*avscore12);
    vars10[b][0]=(avscores1*bb*avscores1);
    vars20[b][0]=(avscores2*bb*avscores2);
    b=b+1;
    }
}

//média das médias, variâncias e correlações dos tempos
nas voltagens v1, v2 e v3
mediageralt1v1=meanc(vetormediast1v1);
mediageralt2v1=meanc(vetormediast2v1);
mediavart1v1=meanc(vetorvart1v1);
mediavart2v1=meanc(vetorvart2v1);
mediacorrv1=meanc(vetorcorrelacaov1);

mediageralt1v2=meanc(vetormediast1v2);
mediageralt2v2=meanc(vetormediast2v2);
mediavart1v2=meanc(vetorvart1v2);
mediavart2v2=meanc(vetorvart2v2);
mediacorrv2=meanc(vetorcorrelacaov2);

mediageralt1v3=meanc(vetormediast1v3);
mediageralt2v3=meanc(vetormediast2v3);
mediavart1v3=meanc(vetorvart1v3);
mediavart2v3=meanc(vetorvart2v3);
mediacorrv3=meanc(vetorcorrelacaov3);

// desvios padrões das médias, das variâncias e das
correlações dos tempos nas voltagens v1, v2 e v3
desvpmediast1v1=sqrt(varc(vetormediast1v1));
desvpmediast2v1=sqrt(varc(vetormediast2v1));
desvpvart1v1=sqrt(varc(vetorvart1v1));
desvpvart2v1=sqrt(varc(vetorvart2v1));
desvpcorrv1=sqrt(varc(vetorcorrelacaov1));
desvpmediast1v2=sqrt(varc(vetormediast1v2));
desvpmediast2v2=sqrt(varc(vetormediast2v2));
desvpvart1v2=sqrt(varc(vetorvart1v2));
desvpvart2v2=sqrt(varc(vetorvart2v2));
desvpcorrv2=sqrt(varc(vetorcorrelacaov2));
desvpmediast1v3=sqrt(varc(vetormediast1v3));
desvpmediast2v3=sqrt(varc(vetormediast2v3));
desvpvart1v3=sqrt(varc(vetorvart1v3));
desvpvart2v3=sqrt(varc(vetorvart2v3));
desvpcorrv3=sqrt(varc(vetorcorrelacaov3));

// desvios padrões dos estimadores
desvpbeta01=sqrt(varbeta01);
desvpbeta02=sqrt(varbeta02);
desvpbeta03=sqrt(varbeta03);
desvpbeta04=sqrt(varbeta04);
desvpbeta05=sqrt(varbeta05);
desvpbeta1=sqrt(varbeta1);
desvpbeta2=sqrt(varbeta2);
desvpbeta3=sqrt(varbeta3);
desvplambda10=sqrt(varlambda10);
desvplambda20=sqrt(varlambda20);
desvplambda120=sqrt(varlambda120);
desvps10=sqrt(vars10);
desvps20=sqrt(vars20);

// erros quadráticos médios das estimativas
for(k=0;k<h;+k)
{
eqmbeta01[k][0]= varbeta01[k][0]+(beta01[k][0]-vbeta01).^2;
eqmbeta02[k][0]= varbeta02[k][0]+(beta02[k][0]-vbeta02).^2;
eqmbeta03[k][0]= varbeta03[k][0]+(beta03[k][0]-vbeta03).^2;
eqmbeta04[k][0]= varbeta04[k][0]+(beta04[k][0]-vbeta04).^2;
eqmbeta05[k][0]= varbeta05[k][0]+(beta05[k][0]-vbeta05).^2;
eqmbeta1[k][0]=varbeta1[k][0]+(beta1[k][0]-vbeta1).^2;
eqmbeta2[k][0]=varbeta2[k][0]+(beta2[k][0]-vbeta2).^2;
eqmbeta3[k][0]=varbeta3[k][0]+(beta3[k][0]-vbeta3).^2;
eqmlambda10[k][0]= varlambda10[k][0]+(lambda10[k][0]-
vlambda10).^2;
eqmlambda20[k][0]= varlambda20[k][0]+(lambda20[k][0]-
vlambda20).^2;
eqmlambda120[k][0]= varlambda120[k][0]+(lambda120[k][0]-
vlambda120).^2;
eqms10[k][0]= vars10[k][0]+(s10[k][0]-vs10).^2;
eqms20[k][0]= vars20[k][0]+(s20[k][0]-vs20).^2;
}

```

```
// intervalos de confiança 90% e 95% dos parâmetros
liminfbeta0190[k][0]=beta01[k][0]-1.6449*desvpbeta01[k][0];
liminfbeta0290[k][0]=beta02[k][0]-1.6449*desvpbeta02[k][0];
liminfbeta0390[k][0]=beta03[k][0]-1.6449*desvpbeta03[k][0];
liminfbeta0490[k][0]=beta04[k][0]-1.6449*desvpbeta04[k][0];
liminfbeta0590[k][0]=beta05[k][0]-1.6449*desvpbeta05[k][0];
liminfbeta190[k][0]=beta1[k][0]-1.6449*desvpbeta1[k][0];
liminfbeta290[k][0]=beta2[k][0]-1.6449*desvpbeta2[k][0];
liminfbeta390[k][0]=beta3[k][0]-1.6449*desvpbeta3[k][0];
liminflambda1090[k][0]=lambda10[k][0]-
1.6449*desvplambda10[k][0];
liminflambda2090[k][0]=lambda20[k][0]-
1.6449*desvplambda20[k][0];
liminflambda12090[k][0]=lambda120[k][0]-
1.6449*desvplambda120[k][0];
liminfs1090[k][0]=s10[k][0]-1.6449*desvps10[k][0];
liminfs2090[k][0]=s20[k][0]-1.6449*desvps20[k][0];

liminfbeta0195[k][0]=beta01[k][0]-1.96*desvpbeta01[k][0];
liminfbeta0295[k][0]=beta02[k][0]-1.96*desvpbeta02[k][0];
liminfbeta0395[k][0]=beta03[k][0]-1.96*desvpbeta03[k][0];
liminfbeta0495[k][0]=beta04[k][0]-1.96*desvpbeta04[k][0];
liminfbeta0595[k][0]=beta05[k][0]-1.96*desvpbeta05[k][0];
liminfbeta195[k][0]=beta1[k][0]-1.96*desvpbeta1[k][0];
liminfbeta295[k][0]=beta2[k][0]-1.96*desvpbeta2[k][0];
liminfbeta395[k][0]=beta3[k][0]-1.96*desvpbeta3[k][0];
liminflambda1095[k][0]=lambda10[k][0]-
1.96*desvplambda10[k][0];
liminflambda2095[k][0]=lambda20[k][0]-
1.96*desvplambda20[k][0];
liminflambda12095[k][0]=lambda120[k][0]-
1.96*desvplambda120[k][0];
liminfs1095[k][0]=s10[k][0]-1.96*desvps10[k][0];
liminfs2095[k][0]=s20[k][0]-1.96*desvps20[k][0];

limsupbeta0190[k][0]=beta01[k][0]+1.6449*desvpbeta01[k][0];
limsupbeta0290[k][0]=beta02[k][0]+1.6449*desvpbeta02[k][0];
limsupbeta0390[k][0]=beta03[k][0]+1.6449*desvpbeta03[k][0];
limsupbeta0490[k][0]=beta04[k][0]+1.6449*desvpbeta04[k][0];
limsupbeta0590[k][0]=beta05[k][0]+1.6449*desvpbeta05[k][0];
limsupbeta190[k][0]=beta1[k][0]+1.6449*desvpbeta1[k][0];
limsupbeta290[k][0]=beta2[k][0]+1.6449*desvpbeta2[k][0];
limsupbeta390[k][0]=beta3[k][0]+1.6449*desvpbeta3[k][0];
limsuplambda1090[k][0]=lambda10[k][0]+1.6449*desvplambda10[
k][0];
limsuplambda2090[k][0]=lambda20[k][0]+1.6449*desvplambda20[
k][0];

limsupbeta0195[k][0]=beta01[k][0]+1.96*desvpbeta01[k][0];
limsupbeta0295[k][0]=beta02[k][0]+1.96*desvpbeta02[k][0];
limsupbeta0395[k][0]=beta03[k][0]+1.96*desvpbeta03[k][0];
limsupbeta0495[k][0]=beta04[k][0]+1.96*desvpbeta04[k][0];
limsupbeta0595[k][0]=beta05[k][0]+1.96*desvpbeta05[k][0];
limsupbeta195[k][0]=beta1[k][0]+1.96*desvpbeta1[k][0];
limsupbeta295[k][0]=beta2[k][0]+1.96*desvpbeta2[k][0];
limsupbeta395[k][0]=beta3[k][0]+1.96*desvpbeta3[k][0];
limsuplambda1095[k][0]=lambda10[k][0]+1.96*desvplambda10[k]
[0];
limsuplambda2095[k][0]=lambda20[k][0]+1.96*desvplambda20[k]
[0];
limsuplambda12095[k][0]=lambda120[k][0]+1.96*desvplambda120
[k][0];
limsups1095[k][0]=s10[k][0]+1.96*desvps10[k][0];
limsups2095[k][0]=s20[k][0]+1.96*desvps20[k][0];
}

/* quantidade intervalos 90% que não contiveram o verdadeiro
valor dos parâmetros*/
for (k=0;k<h;+k)
{
if ((liminfbeta0190[k][0]>vbeta01||
limsupbeta0190[k][0]<vbeta01))
quantbeta0190=quantbeta0190+1;
if ((liminfbeta0290[k][0]>vbeta02||
limsupbeta0290[k][0]<vbeta02))
quantbeta0290=quantbeta0290+1;
if ((liminfbeta0390[k][0]>vbeta03||
limsupbeta0390[k][0]<vbeta03))
quantbeta0390=quantbeta0390+1;
if ((liminfbeta0490[k][0]>vbeta04||
limsupbeta0490[k][0]<vbeta04))
quantbeta0490=quantbeta0490+1;
if ((liminfbeta0590[k][0]>vbeta05||
limsupbeta0590[k][0]<vbeta05))
quantbeta0590=quantbeta0590+1;
if ((liminfbeta190[k][0]>3|| limsupbeta190[k][0]<3))
quantbeta190=quantbeta190+1;
if ((liminfbeta290[k][0]>3|| limsupbeta290[k][0]<3))
quantbeta290=quantbeta290+1;
if ((liminfbeta390[k][0]>3|| limsupbeta390[k][0]<3))
quantbeta390=quantbeta390+1;
}

```

```

if ((liminflambda1090[k][0]>0.1 ||
    limsuplambda1090[k][0]<0.1))
    quantlambda1090=quantlambda1090+1;
if ((liminflambda2090[k][0]>0.1 ||
    limsuplambda2090[k][0]<0.1))
    quantlambda2090=quantlambda2090+1;
if ((liminflambda12090[k][0]>0.2 ||
    limsuplambda12090[k][0]<0.2))
    quantlambda12090=quantlambda12090+1;
if ((liminfs1090[k][0]>0.5 || limsup1090[k][0]<0.5))
    quants1090=quants1090+1;
if ((liminfs2090[k][0]>0.5 || limsup2090[k][0]<0.5))
    quants2090=quants2090+1;
}

// cobertura dos intervalos de confiança 90%
cobeturabeta0190=(h-quantbeta0190)/h)*100;
cobeturabeta0290=(h-quantbeta0290)/h)*100;
cobeturabeta0390=(h-quantbeta0390)/h)*100;
cobeturabeta0490=(h-quantbeta0490)/h)*100;
cobeturabeta0590=(h-quantbeta0590)/h)*100;
cobeturabeta190=(h-quantbeta190)/h)*100;
cobeturabeta290=(h-quantbeta290)/h)*100;
cobeturabeta390=(h-quantbeta390)/h)*100;
cobeturambda1090=(h-quantlambda1090)/h)*100;
cobeturambda2090=(h-quantlambda2090)/h)*100;
cobeturambda12090=(h-quantlambda12090)/h)*100;
cobeturas1090=(h-quants1090)/h)*100;
cobeturas2090=(h-quants2090)/h)*100;

/* quantidade intervalos 95% que não contiveram o verdadeiro
valor dos parâmetros*/
for (k=0;k<h;++k)
{
if ((liminfbeta0195[k][0]>vbeta01 ||
    limsupbeta0195[k][0]<vbeta01))
    quantbeta0195=quantbeta0195+1;
if ((liminfbeta0295[k][0]>vbeta02 ||
    limsupbeta0295[k][0]<vbeta02))
    quantbeta0295=quantbeta0295+1;
if ((liminfbeta0395[k][0]>vbeta03 ||
    limsupbeta0395[k][0]<vbeta03))
    quantbeta0395=quantbeta0395+1;
if ((liminfbeta0495[k][0]>vbeta04 ||
    limsupbeta0495[k][0]<vbeta04))
    quantbeta0495=quantbeta0495+1;
if ((liminfbeta0595[k][0]>vbeta05 ||
    limsupbeta0595[k][0]<vbeta05))
    quantbeta0595=quantbeta0595+1;
if ((liminfbeta195[k][0]>3 || limsupbeta195[k][0]<3))
    quantbeta195=quantbeta195+1;
if ((liminfbeta295[k][0]>3 || limsupbeta295[k][0]<3))
    quantbeta295=quantbeta295+1;
if ((liminfbeta395[k][0]>3 || limsupbeta395[k][0]<3))
    quantbeta395=quantbeta395+1;
if ((liminflambda1095[k][0]>0.1 ||
    limsuplambda1095[k][0]<0.1))
    quantlambda1095=quantlambda1095+1;
if ((liminflambda2095[k][0]>0.1 ||
    limsuplambda2095[k][0]<0.1))
    quantlambda2095=quantlambda2095+1;
if ((liminflambda12095[k][0]>0.2 ||
    limsuplambda12095[k][0]<0.2))
    quantlambda12095=quantlambda12095+1;
if ((liminfs1095[k][0]>0.5 || limsup1095[k][0]<0.5))
    quants1095=quants1095+1;
if ((liminfs2095[k][0]>0.5 || limsup2095[k][0]<0.5))
    quants2095=quants2095+1;
}

// cobertura dos intervalos de confiança 95%
cobeturabeta0195=(h-quantbeta0195)/h)*100;
cobeturabeta0295=(h-quantbeta0295)/h)*100;
cobeturabeta0395=(h-quantbeta0395)/h)*100;
cobeturabeta0495=(h-quantbeta0495)/h)*100;
cobeturabeta0595=(h-quantbeta0595)/h)*100;
cobeturabeta195=(h-quantbeta195)/h)*100;
cobeturabeta295=(h-quantbeta295)/h)*100;
cobeturabeta395=(h-quantbeta395)/h)*100;
cobeturambda1095=(h-quantlambda1095)/h)*100;
cobeturambda2095=(h-quantlambda2095)/h)*100;
cobeturambda12095=(h-quantlambda12095)/h)*100;
cobeturas1095=(h-quants1095)/h)*100;
cobeturas2095=(h-quants2095)/h)*100;

/*Médias das estimativas, dos desvios padrões e dos EQM dos
estimadores */
mediabeta01=meanc(beta01);
mediabeta02=meanc(beta02);
mediabeta03=meanc(beta03);
mediabeta04=meanc(beta04);
mediabeta05=meanc(beta05);
mediabeta1=meanc(beta1);
mediabeta2=meanc(beta2);
mediabeta3=meanc(beta3);
medialambda10=meanc(lambda10);

```

```

media1ambda20=meanc(lambda20);
media1ambda120=meanc(lambda120);
medias10=meanc(s10);
medias20=meanc(s20);

beta01ord=sortc(beta01);
beta02ord=sortc(beta02);
beta03ord=sortc(beta03);
beta04ord=sortc(beta04);
beta05ord=sortc(beta05);
beta1ord=sortc(beta1);
beta2ord=sortc(beta2);
beta3ord=sortc(beta3);
lambda10ord=sortc(lambda10);
lambda20ord=sortc(lambda20);
lambda120ord=sortc(lambda120);
s10ord=sortc(s10);
s20ord=sortc(s20);

medianabeta01=beta01ord[(h/2)-1][0];
medianabeta02=beta02ord[(h/2)-1][0];
medianabeta03=beta03ord[(h/2)-1][0];
medianabeta04=beta04ord[(h/2)-1][0];
medianabeta05=beta05ord[(h/2)-1][0];
medianabeta1=beta1ord[(h/2)-1][0];
medianabeta2=beta2ord[(h/2)-1][0];
medianabeta3=beta3ord[(h/2)-1][0];
medianalambda10=lambda10ord[(h/2)-1][0];
medianalambda20=lambda20ord[(h/2)-1][0];
medianalambda120=lambda120ord[(h/2)-1][0];
medianas10=s10ord[(h/2)-1][0];
medianas20=s20ord[(h/2)-1][0];

mediadpbeta01=meanc(desvpbeta01);
mediadpbeta02=meanc(desvpbeta02);
mediadpbeta03=meanc(desvpbeta03);
mediadpbeta04=meanc(desvpbeta04);
mediadpbeta05=meanc(desvpbeta05);
mediadpbeta1=meanc(desvpbeta1);
mediadpbeta2=meanc(desvpbeta2);
mediadpbeta3=meanc(desvpbeta3);
mediadplambda10=meanc(desvplambda10);
mediadplambda20=meanc(desvplambda20);
mediadplambda120=meanc(desvplambda120);
mediadps10=meanc(desvps10);
mediadps20=meanc(desvps20);

mediaeqmbeta01=meanc(eqmbeta01);
mediaeqmbeta02=meanc(eqmbeta02);
mediaeqmbeta03=meanc(eqmbeta03);
mediaeqmbeta04=meanc(eqmbeta04);
mediaeqmbeta05=meanc(eqmbeta05);
mediaeqmbeta1=meanc(eqmbeta1);
mediaeqmbeta2=meanc(eqmbeta2);
mediaeqmbeta3=meanc(eqmbeta3);
mediaeqmlambda10=meanc(eqmlambda10);
mediaeqmlambda20=meanc(eqmlambda20);
mediaeqmlambda120=meanc(eqmlambda120);
mediaeqms10=meanc(eqms10);
mediaeqms20=meanc(eqms20);

/*desvios padrões das estimativas, dos desvios padrões e dos
EQM dos estimadores*/
desvpbeta01=sqrt(varc(beta01));
desvpbeta02=sqrt(varc(beta02));
desvpbeta03=sqrt(varc(beta03));
desvpbeta04=sqrt(varc(beta04));
desvpbeta05=sqrt(varc(beta05));
desvpbeta1=sqrt(varc(beta1));
desvpbeta2=sqrt(varc(beta2));
desvpbeta3=sqrt(varc(beta3));
desvpalambda10=sqrt(varc(lambda10));
desvpalambda20=sqrt(varc(lambda20));
desvpalambda120=sqrt(varc(lambda120));
desvps10=sqrt(varc(s10));
desvps20=sqrt(varc(s20));

dpdpbeta01=sqrt(varc(desvpbeta01));
dpdpbeta02=sqrt(varc(desvpbeta02));
dpdpbeta03=sqrt(varc(desvpbeta03));
dpdpbeta04=sqrt(varc(desvpbeta04));
dpdpbeta05=sqrt(varc(desvpbeta05));
dpdpbeta1=sqrt(varc(desvpbeta1));
dpdpbeta2=sqrt(varc(desvpbeta2));
dpdpbeta3=sqrt(varc(desvpbeta3));
dpdplambda10=sqrt(varc(desvplambda10));
dpdplambda20=sqrt(varc(desvplambda20));
dpdplambda120=sqrt(varc(desvplambda120));
dpdps10=sqrt(varc(desvps10));
dpdps20=sqrt(varc(desvps20));

dpeqmbeta01=sqrt(varc(eqmbeta01));
dpeqmbeta02=sqrt(varc(eqmbeta02));
dpeqmbeta03=sqrt(varc(eqmbeta03));
dpeqmbeta04=sqrt(varc(eqmbeta04));

```



```

dpeqmbeta05=sqrt(varc(eqmbeta05));
dpeqmbeta1=sqrt(varc(eqmbeta1));
dpeqmbeta2=sqrt(varc(eqmbeta2));
dpeqmbeta3=sqrt(varc(eqmbeta3));
dpeqmlambda10=sqrt(varc(eqmlambda10));
dpeqmlambda20=sqrt(varc(eqmlambda20));
dpeqmlambda120=sqrt(varc(eqmlambda120));
dpeqms10=sqrt(varc(eqms10));
dpeqms20=sqrt(varc(eqms20));

//SAIDA
print("\n medias gerais dos tempos, das variâncias e
      correlação na voltagem v1",
      mediageralt1v1~mediageralt2v1~mediavart1v1~mediavart2v1
      ~mediacorr1);

print("\n medias gerais dos tempos, das variâncias e
      correlação na voltagem v2",
      mediageralt1v2~mediageralt2v2~mediavart1v2~mediavart2v2
      ~mediacorr2);

print("\n medias gerais dos tempos, das variâncias e
      correlação na voltagem v3",
      mediageralt1v3~mediageralt2v3~mediavart1v3~mediavart2v3
      ~mediacorr3);

print("\n dp das medias, variancias e correlacoes na voltagem
      v1",
      desvpmediast1v1~desvpmediast2v1~desvpvart1v1~
      desvpvart2v1~desvpcorr1);

print("\n dp das medias, variancias e correlacoes na voltagem
      v2",
      desvpmediast1v2~desvpmediast2v2~desvpvart1v2~
      desvpvart2v2~desvpcorr2);

print("\n dp das medias, variancias e correlacoes na voltagem
      v3",
      desvpmediast1v3~desvpmediast2v3~desvpvart1v3~
      desvpvart2v3~desvpcorr3);

print("\n\n EMV ", beta01~beta02~beta03~beta04~beta05~
      beta1~beta2~beta3~
      lambda10~lambda20~lambda120~s10~s20);

print ("\n\n desvios padroes emv",
      desvpbeta01~desvpbeta02~desvpbeta03~desvpbeta04~
      desvpbeta05~desvpbeta1~desvpbeta2~desvpbeta3~
      desvplambda10~desvplambda20~desvplambda120~
      desvps10~desvps20);

print ("\n IC 90% das emv:" ,
      (liminfbeta0190~limsupbeta0190),
      (liminfbeta0290~limsupbeta0290),
      (liminfbeta0390~limsupbeta0390),
      (liminfbeta0490~limsupbeta0490),
      (liminfbeta0590~limsupbeta0590),
      (liminfbeta190~limsupbeta190),
      (liminfbeta290~limsupbeta290),
      (liminfbeta390~limsupbeta390),
      (liminflambda1090~limsuplambda1090),
      (liminflambda2090~limsuplambda2090),
      (liminflambda12090~limsuplambda12090),
      (liminfps1090~limsupps1090),
      (liminfps2090~limsupps2090));

print ("\n IC 95% das emv:" ,
      (liminfbeta0195~limsupbeta0195),
      (liminfbeta0295~limsupbeta0295),
      (liminfbeta0395~limsupbeta0395),
      (liminfbeta0495~limsupbeta0495),
      (liminfbeta0595~limsupbeta0595),
      (liminfbeta195~limsupbeta195),
      (liminfbeta295~limsupbeta295),
      (liminfbeta395~limsupbeta395),
      (liminflambda1095~limsuplambda1095),
      (liminflambda2095~limsuplambda2095),
      (liminflambda12095~limsuplambda12095),
      (liminfps1095~limsupps1095),
      (liminfps2095~limsupps2095));

print ("\n coberturas IC 90%:",
      cobeturabeta0190~
      cobeturabeta0290~
      cobeturabeta0390~
      cobeturabeta0490~
      cobeturabeta0590~
      cobeturabeta190~
      cobeturabeta290~
      cobeturabeta390~
      cobeturlambda1090~
      cobeturlambda2090~
      cobeturlambda12090~
      coberturas1090~
      coberturas2090);

```

```

print ("\n coberturas IC 95%:",
cobeturabeta0195~
cobeturabeta0295~
cobeturabeta0395~
cobeturabeta0495~
cobeturabeta0595~
cobeturabeta195~
cobeturabeta295~
cobeturabeta395~
cobaturalambda1095~
cobaturalambda2095~
cobaturalambda12095~
coberturas1095~
coberturas2095);

print ("\n\n medias EMV ",
mediabeta01~mediabeta02~mediabeta03~
mediabeta04~mediabeta05~mediabeta1~mediabeta2~
mediabeta3~medialambda10~medialambda20~medialambda120~
medias10~medias20);

print ("\n\n medianas EMV ",
medianabeta01~medianabeta02~medianabeta03~
medianabeta04~medianabeta05~medianabeta1~medianabeta2~
medianabeta3~medianalambda10~medianalambda20~
medianalambda120~medianas10~medianas20);

print ("\n\n medias desvios padroes",
mediadpbeta01~mediadpbeta02~mediadpbeta03~
mediadpbeta04~mediadpbeta05~mediadpbeta1~mediadpbeta2~
mediadpbeta3~mediadplambda10~mediadplambda20~
mediadplambda120~mediadps10~mediadps20);

print ("\n\n medias EQM das emv",
mediaeqmbeta01~mediaeqmbeta02~mediaeqmbeta03~
mediaeqmbeta04~mediaeqmbeta05~mediaeqmbeta1~
mediaeqmbeta2~mediaeqmbeta3~mediaeqmlambda10~
mediaeqmlambda20~mediaeqmlambda120~
mediaeqms10~mediaeqms20);

print ("\n\n desvios padroes das emv",
desvpabeta01~desvpabeta02~desvpabeta03~
desvpabeta04~desvpabeta05~desvpabeta1~desvpabeta2~
desvpabeta3~desvpalambda10~desvpalambda20~
desvpalambda120~desvpas10~desvpas20);

print ("\n\n desvios padroes dp das emv",
dpdpbeta01~dpdpbeta02~dpdpbeta03~
dpdpbeta04~dpdpbeta05~dpdpbeta1~dpdpbeta2~dpdpbeta3~
dpdp1lambda10~dpdp1lambda20~dpdp1lambda120~
dpdps10~dpdps20);

print ("\n\n desvios padroes eqm ",
dpeqmbeta01~dpeqmbeta02~dpeqmbeta03~
dpeqmbeta04~dpeqmbeta05~dpeqmbeta1~
dpeqmbeta2~dpeqmbeta3~
dpeqmlambda10~dpeqmlambda20~
dpeqmlambda120~dpeqms10~dpeqms20);

print ("\n\n amostras que não convergiram: ", q-r);

print ("\n\n amostras que convergiram e não satisfizeram a
condicao imposta nas estimativas: ", r-e);

print ("\n\n a condicao de beta01 nao foi satisfeita: ",
r-e1);
print ("\n\n a condicao de beta02 nao foi satisfeita: ",
r-e2);
print ("\n\n a condicao de beta03 nao foi satisfeita: ",
r-e3);
print ("\n\n a condicao de beta04 nao foi satisfeita: ",
r-e4);
print ("\n\n a condicao de beta05 nao foi satisfeita: ",
r-e5);
print ("\n\n a condicao de beta1 nao foi satisfeita: ",
r-e6);
print ("\n\n a condicao de beta2 nao foi satisfeita: ",
r-e7);
print ("\n\n a condicao de beta3 nao foi satisfeita: ",
r-e8);

print ("\n\n amostras com problemas nas matrizes de
covariâncias: ", e-h);

print ("\n\n tempo de execucao:",timespan(time));
|

```

REFERÊNCIAS BIBLIOGRÁFICAS

- BARLOW, R. & PROSCHAN, F. *Statistical Theory of Reliability and Life Testing*. New York: Holt, Rinehart and Winston, 1981. 290p.
- BEST, N., COWLES, M. K. , VINES, K. *CODA: Convergence Diagnosis and Output Analysis Software for Gibbs Sampling Output Version 0.30*. Cambridge: University of Nebraska , 1996. 41p.
- BLOCK, H. W. & BASU, A. P. A Continuous Bivariate Exponential Extension. *Journal of the American Statistical Association*, 69(348): 1031-37, dec, 1974.
- BOX, G.E.P. & TIAO, G.C. *Bayesian Inference in Statistical Analysis*. Philippines: Addison-Wesley, 1973. 588p.
- CASELLA, G. & GEORGE, E. I. Explaining the Gibbs Sampler. *The American Statistician*, 46(3): 167-74, aug, 1992.
- CHIB, S. & GREENBERG, E. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4), 327-335, nov, 1995.
- DOORNIK, J. A. *Ox Version 2.10*. London, 1999.
- DOWNTON, F. Bivariate Exponential Distributions in Reliability Theory. *Journal of the Royal Statistical Society*, series B, 32(3): 408-17, 1970.

- FLETCHER, R. *Practical Methods of Optimization*. 2nd ed. New York: John Wiley & Sons, 1987. 436p.
- FREUND, J. E. A Bivariate Extension of the Exponential Distribution. *Journal of the American Statistical Association*, 56: 971-77, 1961.
- GAMERMAN, D. *Simulação Estocástica Via Cadeias de Markov*. São Paulo: ABE- Associação Brasileira de Estatística, 1996. 196p.
- GELMAN, A. & RUBIN, D.B. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4): 457-72, nov, 1992.
- GEMAN, S. & GEMAN, D. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741, 1984.
- GEWEKE, J. Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In *Bayesian Statistics 4*. (J.M. Bernardo, J.O. Berger, A. P. Dawid e A. F. M. Smith, eds), Oxford Univ. Press. 169-193, 1992.
- GUMBEL, E. J. Bivariate Exponential Distributions. *Journal of the American Statistical Association*, 55: 698-707, 1960.
- HASTINGS, W. K. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57: 97-109, 1970.
- HAWKES, A. G. A Bivariate Exponential Distribution with Applications to Reliability. *Journal of the Royal Statistical Society*, series B, 34(1): 129-31, 1972.
- HEIDELBERGER, P. & WELCH, P.H. Simulation Run Length Control in the Presence of an Initial Transient. *Operations Research*, 31:1109-1144,1983.

- JAMES, B. R. *Probabilidade: Um Curso em Nível Intermediário*. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1981. 304p.
- KASS, R. E. & SLATE, E. H. Some Diagnostics of Maximum Likelihood and Posterior Nonnormality. *The Annals of Statistics*, 22: 668-695, 1994.
- KENNEDY, W. E., Jr. & GENTLE, J. E. *Statistical Computing*. New York: Marcel Dekker, 1980. 591p.
- LAWLESS, J. F. *Statistical Models and Methods for Lifetime Data*. New York: John Wiley & Sons, 1982. 580p.
- LEITE, J. G. & SINGER, J. M. *Métodos Assintóticos em Estatística: Fundamentos e Aplicações*. São Paulo: IME – USP, 1990. 130p.
- MANN, N. R., SCHAFER, R. E., SINGPURWALLA, N. D. *Methods for Statistical Analysis of Reliability and Life Data*. New York: John Wiley & Sons, 1974. 564p.
- MARSHALL, A. W. & OLKIN, I. A Multivariate Exponential Distribution. *Journal of the American Statistical Association*, 62: 30-44, mar, 1967.
- METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. e TELLER, E. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21: 1087-1092, 1953.
- MOOD, A. M., GRAYBILL, F. A., BOES, D. C. *Introduction to the Theory of Statistics*. 3rd ed. New York: McGraw-Hill, 1974. 564p.
- NELSON, W. *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*. New York: John Wiley & Sons, 1990. 601p.

- PAULSON, A. S. A Characterization of the Exponential Distribution and a Bivariate Exponential Distribution. *Sankhyā*, series A, 35(1): 69-78, mar, 1973.
- RAFTERY, A. E. e Lewis, S. How Many Iterations in the Gibbs Sampler? In *Bayesian Statistics 4*. (J.M. Bernardo, J.O. Berger, A. P. Dawid e A. F. M. Smith, eds), Oxford Univ. Press. 763-773, 1992.
- ROHATGI, V. K. *An Introduction to Probability Theory and Mathematical Statistics*. New York: John Wiley & Sons, 1976. 684p.
- RYU, K. An Extention of Marshall and Olkin's Bivariate Exponential Distribution. *Journal of the American Statistical Association*, 88(424): 1458-65, dec, 1993.
- SARKAR, S.K. A Continuous Bivariate Exponential Distribution. *Journal of the American Statistical Association*, 82(398): 667-76, jun, 1987.
- VON NEUMANN, J. *Various Techniques Used in Connection with Random Digits*. Washington: National Bureau of Standards, 1951.