



**Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Estatística**

## **Discriminante para mistura de modelos bivariados**

**Erique Pereira Neto**

Dissertação apresentada ao Departamento de Estatística da Universidade de Brasília, como requisito parcial para obtenção do título de Mestre em Estatística

**Brasília  
2018**



**Erique Pereira Neto**

## **Discriminante para mistura de modelos bivariados**

Dissertação submetida ao Programa de Pós-Graduação em Estatística do Departamento de Estatística da Universidade de Brasília como parte dos requisitos para a obtenção do grau de Mestre em Estatística.

Orientadora: Professora Dra. **Cira Etheowalda Guevara Otiniano**

**Brasília, 7 de dezembro de 2018**

Dissertação submetida ao Programa de Pós-Graduação em Estatística do Departamento de Estatística da Universidade de Brasília como parte dos requisitos para a obtenção do grau de Mestre em Estatística.

Banca:

Prof<sup>ª</sup>. Dra. Cira Etheowalda Guevara Otiniano  
Orientador, EST/UnB

Prof. Dr. Carlos Henrique Ribeiro Lima  
FT/ UnB

Prof<sup>ª</sup>. Dra. Verónica Andrea González-López  
Unicamp

Prof<sup>ª</sup>. Dra. Juliana Betini Fachini Gomes  
Suplente-EST/UnB

# Dedicatória

Ao meu tio Adão e à memória da minha tia Maria, que me deram a oportunidade e todo apoio para prosseguir com meus estudos. E por ter sido exemplo de determinação e coragem.

# Agradecimentos

Agradeço ao meu tio Adão, por ter dado a oportunidade de poder avançar com meus estudos e por todos os ensinamentos valiosos.

Aos meus pais e irmãos, por acreditarem e torcerem por mim.

À minha orientadora Profa. Cira, pela oportunidade, pela paciência e apoio na realização desse trabalho. Sou muito grato!

Agradeço imensamente a Lenara Mendes, que sempre esteve junto a mim, me apoiando durante os momentos mais difíceis e também os mais felizes! Com você tudo foi mais fácil.

Às minhas e meus colegas do mestrado, alguns de vida mesmo, por terem me apoiado nos estudos e na vivência no ambiente acadêmico. Mas em especial à Geiziane, que é minha amiga desde o tempo de graduação e que muito me apoiou nesse percurso, à Alessandra que tanto apoiou durante os estudos no mestrado e ao Alisson pelo apoio.

Agradeço ao meu amigo Cleyverson, pelo apoio dado quando meu chefe no BB, e também a todos os meus colegas da Agência BB UnB.

Ao Júlio César pelo incentivo.

À todos que torceram por mim e me apoiaram direta ou indiretamente.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Obrigado!

# Resumo

Nas últimas décadas a utilização de modelos de mistura para fins de agrupamento vem sendo consideravelmente aumentada devido principalmente à existência de métodos computacionais eficientes que facilitam a estimativa dos parâmetros. Por outro lado, as cópulas se tornaram modelos populares para modelar dependências em dados multivariados. Neste trabalho aplicam-se misturas de cópulas bivariadas visando usá-las para análise de dados de misturas com dois *clusters*. Foram propostas funções discriminantes não-lineares para identificar a procedência de uma dada observação, isto é, para discriminar a qual *cluster* ela pertence, considerando misturas de cópulas arquimedianas bivariadas. Para estimação dos parâmetros foi utilizado o método de máxima verossimilhança, via algoritmo EM, baseado na abordagem padrão para modelos de mistura.

**Palavras-chave:** Cópulas, Análise Discriminante, Mistura de Cópulas, Algoritmo EM.

# Abstract

In the last decades the use of mixing models for grouping purposes has been considerably increased due mainly to the existence of efficient computational methods that facilitate the estimation of the parameters. On the other hand, copulas have become popular models for modeling dependencies in multivariate data. In this work we apply mixtures of bivariate copulas to use them to analyze data of mixtures with two clusters. Non-linear discriminant functions were proposed to identify the origin of a given observation, that is, to discriminate to which cluster it belongs, considering mixtures of bivariate Archimedean copulas. For the estimation of the parameters, the maximum likelihood method was used, via the EM algorithm, based on the standard approach for mixing models.

**Key Words:** Copulas, Discriminant Analysis, Copulas Mixture, EM Algorithm.



# Sumário

<b>Introdução</b>	<b>1</b>
<b>1 Conceitos Preliminares</b>	<b>3</b>
1.1 Cópulas . . . . .	4
1.2 Cópulas Arquimedianas . . . . .	6
1.3 Dependência . . . . .	9
1.4 Mistura de Distribuições . . . . .	11
<b>2 Mistura de Cópulas</b>	<b>14</b>
2.1 Modelo . . . . .	14
2.2 Estrutura de Dependência de Mistura de Cópulas . . . . .	15
<b>3 Estimação</b>	<b>22</b>
3.1 Estimação Por Máxima Verossimilhança . . . . .	22
3.2 Estimação usando Algoritmo EM . . . . .	23
3.3 Matriz de Informação . . . . .	24
<b>4 Análise Discriminante</b>	<b>26</b>
4.1 Função discriminante para misturas de cópulas . . . . .	26
4.2 Erros de classificação . . . . .	31
<b>5 Resultados</b>	<b>33</b>
5.1 Simulação . . . . .	33
5.2 Aplicação . . . . .	46
5.2.1 Dados de erupções de Geysers Old Faithful em Yellowstone National Park, Wyoming, EUA. . . . .	46
<b>6 Conclusão</b>	<b>52</b>
<b>Referências Bibliográficas</b>	<b>53</b>
<b>A Códigos R</b>	<b>55</b>

# Lista de Figuras

1	Simulação de mistura de dados bivariados formando dois <i>clusters</i> . . . . .	1
1.1	Curvas de nível da densidade da cópula de Clayton para $\theta = -0.1$ , (a), $\theta = 1.0$ , (b), e $\theta = 3.0$ , (c). . . . .	7
1.2	Curvas de nível da densidade da cópula de Gumbel para $\theta = 1.1$ , (a), $\theta = 1.5$ , (b), e $\theta = 3.0$ , (c). . . . .	7
1.3	Curvas de nível da densidade da cópula de Frank para $\theta = -5.0$ , (a), $\theta = 0.1$ , (b), e $\theta = 5.0$ , (c). . . . .	7
1.4	Curvas de nível da densidade da cópula de Joe para $\theta = 1.0$ , (a), $\theta = 2.0$ , (b), e $\theta = 4.0$ , (c). . . . .	8
1.5	Gráficos de diferentes misturas de distribuições Normais: a) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 2, \mu_2 = 8, \sigma_1 = 2, \sigma_2) = 3$ ; b) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 0, \mu_2 = 8, \sigma_1 = 1, \sigma_2) = 2$ e c) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = -4, \mu_2 = 2, \sigma_1 = 3, \sigma_2) = 1$ . . . . .	12
1.6	Gráficos de diferentes misturas de distribuições GEV: a) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 6, \mu_2 = 15, \sigma_1 = 3, \sigma_2 = 10, \alpha_1 = 0.5, \alpha_2 = 0.8)$ ; b) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 4, \mu_2 = 8, \sigma_1 = 3, \sigma_2 = 3, \alpha_1 = -0.8, \alpha_2 = 0.6)$ e c) $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 10, \mu_2 = 25, \sigma_1 = 2, \sigma_2 = 10, \alpha_1 = 0.0, \alpha_2 = 0.8)$ . . . . .	13
2.1	Curvas de níveis para mistura de cópula de Frank para $\theta_1 = -5.0$ e cópula de Clayton para $\theta_2 = 1.0$ , e diferentes valores de $p$ e marginais normais padrões ( $C(u, v) = p * Frank + (1 - p) * Clayton$ ). . . . .	17
2.2	Curvas de níveis para mistura de cópula de Frank para $\theta_1 = -5.0$ e cópula de Joe para $\theta_2 = 2.0$ , e diferentes valores de $p$ e marginais normais padrões ( $C(u, v) = p * Frank + (1 - p) * Joe$ ). . . . .	18
2.3	Curvas de níveis para diferentes misturas de cópulas: a-c) $p * Gumbel(\theta_1 = 3.0) + (1 - p) * Frank(\theta_2 = -5.0)$ , d-f) $p * Joe(\theta_1 = 1.1) + (1 - p) * Gumbel(\theta_2 = 3.0)$ , g-i) $p * Gumbel(\theta_1 = 3.0) + (1 - p) * Clayton(\theta_2 = 1.1)$ , todas com marginais normais padrões . . . . .	19

2.4	Curvas de níveis para mistura de cópulas Clayton e Gumbel: $p * Clayton(\theta_1 = -0.1) + (1 - p) * Gumbel(\theta_2 = 3.0)$ , $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ). Veja definição da distribuição Gumbel na equação (5.1), com $\alpha = 0$ . . . . .	20
2.5	Curvas de níveis para mistura de cópulas Clayton e Gumbel: $p * Clayton(\theta_1 = -0.1) + (1 - p) * Gumbel(\theta_2 = 1.1)$ , $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ). . . . .	20
2.6	Curvas de níveis para mistura de cópulas Clayton e Gumbel: $p * Clayton(\theta_1 = 1.0) + (1 - p) * Gumbel(\theta_2 = 3.0)$ , $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ). . . . .	21
2.7	Curvas de níveis para mistura de cópulas Joe e Frank: $p * Joe(\theta_1 = 2.0) + (1 - p) * Frank(\theta_2 = -5.0)$ , $p = 0.75$ , marginais da Joe com distribuição Gumbel( $\mu = 3, \sigma = 2$ ) e marginais da cópula Frank com distribuição Gumbel( $\mu = 3, \sigma = 1$ ). . . . .	21
5.1	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m1 a m4 da Tabela 5.1. . . . .	37
5.2	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m5 a m8 da Tabela 5.1. . . . .	38
5.3	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m9 a m12 da Tabela 5.1. . . . .	39
5.4	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m13 a m16 da Tabela 5.1. . . . .	40
5.5	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m17 a m20 da Tabela 5.1. . . . .	41
5.6	Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m21 a m24 da Tabela 5.1. . . . .	42
5.7	Histograma das variáveis X e Y simuladas, as densidades estimadas e suas respectivas densidades empíricas. . . . .	44
5.8	Gráfico de dispersão de X e Y. . . . .	44
5.9	Curvas de níveis da densidade de X e Y, com separação de <i>clusters</i> pela função discriminante da mistura. . . . .	45
5.10	Histograma das variáveis, (a):"tempo até próxima erupção"e (b):"tempo de duração da erupção"e suas respectivas densidades empíricas. . . . .	46
5.11	Gráfico de dispersão do "tempo até próxima erupção"e "tempo de duração da erupção". . . . .	46
5.12	Densidade das variáveis, (a):"tempo até próxima erupção"e (b):"tempo de duração da erupção"ajustadas aos dados. . . . .	47

5.13	Curvas de níveis da densidade conjunta dos dados considerando mistura de duas cópulas Gumbel. . . . .	49
5.14	Curvas de níveis da densidade conjunta dos dados considerando misturas de cópulas: a) Gumbel-Gumbel; b) Frank-Frank; c) Joe-Joe; e d) Gumbel, sem mistura. Os parâmetros das misturas estão na Tabela 5.9. . . . .	50
5.15	Curvas de níveis da densidade conjunta plotada contra os dados, com separação de <i>cluster</i> pela função discriminante da mistura . . . . .	51

# Lista de Tabelas

1.1	Principais Cópulas arquimedianas e seus respectivos geradores. . . . .	6
1.2	Principais Cópulas arquimedianas e medidas de dependência . . . . .	11
4.1	Mistura de cópulas e suas funções discriminantes . . . . .	28
5.1	Parâmetros usados nas simulações. . . . .	34
5.2	Estimadores dos parâmetros das marginas das misturas simuladas mostradas na Tabela 5.1, com os parâmetros originais $\mu_1 = 4$ , $\mu_2 = 0$ , $\sigma_1 = 1$ e $\sigma_2 = 1$ , considerando distribuição Gumbel. . . . .	35
5.3	Estimadores dos parâmetros das misturas, coeficientes de correlação de Kendal estimado usando-se a equação (2.2) e amostral, coeficientes de correlação de Spearman estimado conforme equação (2.3) e amostral, e coeficientes de dependência caudal inferior e superior. . . . .	36
5.4	Erros-padrões dos estimadores dos parâmetros das misturas via Bootstrap. . . . .	36
5.5	Erros do discriminante e viés em relação ao discriminante ótimo . . . . .	43
5.6	Parâmetros estimados das distribuições marginais de X e Y, considerando distribuição GEV com $\mu_i$ parâmetro de locação, $\sigma_i$ parâmetro de escala e $\alpha_i$ parâmetro de forma, com $i = 1, 2$ . . . . .	44
5.7	Tabela comparativa de valores da função de log verossimilhança, AIC e BIC para algumas misturas de cópulas estimadas. . . . .	45
5.8	Parâmetros das distribuições do "tempo até próxima erupção", $g_1$ , e "tempo de duração da erupção", $g_2$ ; em que $g_1$ e $g_2$ seguem a forma da equação (5.3). . . . .	47
5.9	Tabela comparativa de valores da função de log verossimilhança, AIC e BIC para algumas misturas de cópulas estimadas. . . . .	49

# Introdução

Nas últimas décadas, o agrupamento baseado em modelos de probabilidade é uma alternativa amplamente aceita. A representação de uma mistura finita como uma função de densidade populacional permite a identificação de diferentes *clusters* através da aplicação de metodologias estatísticas padrões e faz inferência baseada em um modelo sólido com forte embasamento teórico. Para uma visão geral sobre a aplicação de modelos de mistura finita para fins de agrupamento ver, McLachlan e Peel (2000), Böhning (1999) e Melnykov et al. (2010).

Considere uma amostra de pares de variáveis aleatórias  $(X_i, Y_i)$ ,  $i = 1, \dots, n$ , em que as observações deste vetor advêm de dois *clusters* e com estruturas de dependência podendo ser diferentes de um *cluster* para o outro, como mostrado na figura abaixo.

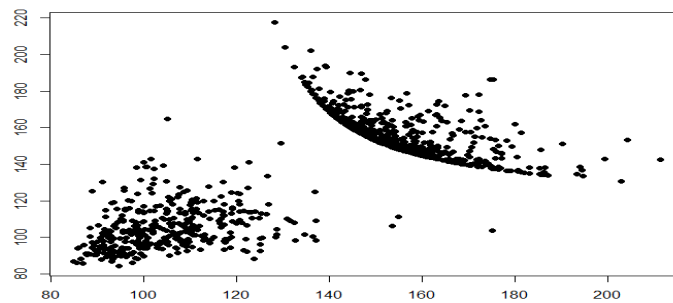


Figura 1: Simulação de mistura de dados bivariados formando dois *clusters*.

Para modelar esses pares com modelos de mistura finita, emprega-se uma mistura de distribuições bivariadas com diferentes matrizes de correlação. A maioria dos modelos propostos tem como base densidades normais ou t-student bivariadas (componentes normais ou t-student bivariadas), ou seja, os *clusters* são caracterizados por uma distribuição normal ou t-student bivariada. Nestes casos, o modelo populacional é normal heterocedástico, cuja estrutura de covariância difere entre os *clusters*. Isso é bastante restritivo, pois assume uma estrutura elíptica nas componentes, o que não é verdade em certas aplicações, como em finanças, onde as assimetrias são tipicamente observadas. Para ir além da estrutura elíptica dos *clusters*, outras densidades multivariadas foram empregadas, ver Karlis e Santourian (2009), que não têm flexibilidade.

Para superar essa desvantagem, um modelo baseado em mistura de cópulas é utilizado, visto que ele permite certas associações mais flexíveis. O interesse principal de cópulas reside no fato de que elas separam a associação das marginais e fornecem uma ampla gama de estruturas de associação. Misturas de cópulas para definir novas famílias não é nova. Lai et al. (2009) utilizaram mistura de cópulas para dados em finanças, Arakelian e Karlis (2014) usaram a mistura de cópulas para a modelagem de *clusters*.

Neste trabalho, foram utilizadas misturas de duas cópulas arquimedianas para agrupar observações com diferentes estruturas de associação (dois *clusters*) e foram propostas funções discriminantes não-lineares para identificar a procedência de uma dada observação, isto é, para discriminar a qual *cluster* ela pertence, considerando mistura de cópulas arquimedianas bivariadas.

Sendo assim, este trabalho está organizado em cinco capítulos. No Capítulo 1 é feita uma revisão sobre cópulas e suas estruturas de dependência, tratando com ênfase as cópulas bivariadas arquimedianas. O Capítulo 2 contém a definição do modelo de mistura de cópulas e sua estrutura de dependência, em que é descrito o cálculo de medidas de dependência como o  $\tau$  de Kendall,  $\rho$  de Spearman e os parâmetros de dependência de cauda inferior e superior. No Capítulo 3 é descrito o procedimento de estimação do vetor de parâmetros da mistura de cópulas, a obtenção da matriz de informação de Fisher e do erro padrão dos parâmetros estimados. No Capítulo 4 define-se a função de discriminante não-linear de Bayes e obtêm-se expressões para modelos específicos de misturas. Os resultados da performance dos estimadores da mistura de cópulas bivariadas e o cálculo dos erros ao utilizar o discriminante proposto para classificar uma observação da mistura são apresentados no Capítulo 5. Finalmente, uma aplicação de mistura de cópulas bivariadas em um conjunto de dados reais é apresentado no Capítulo 5.

# Capítulo 1

## Conceitos Preliminares

O presente capítulo apresenta conceitos e propriedades importantes sobre cópulas que serão úteis ao longo deste trabalho. Inicialmente, faz-se necessária a introdução do conceito de volume sobre um cubo  $n$ -dimensional e de distribuição multivariada, que será utilizado na definição de cópulas.

**Definição 1.1.** *Seja  $S_1 \times \dots \times S_n$  um subconjunto não vazio de  $[-\infty, \infty]$ ,  $H$  uma função real  $n$ -variada com  $\text{Dom}H = S_1 \times \dots \times S_n$ . E seja  $B = [\mathbf{a}, \mathbf{b}]$  um cubo  $n$ -dimensional com todos os vértices contidos em  $\text{Dom}H$ . Então o  $H$ -volume de  $B$  é dado por*

$$V_{H(B)} = \sum_{\mathbf{c}} \text{ sinal}(\mathbf{c}) H(\mathbf{c}), \quad (1.1)$$

em que a soma é feita em todos os vértices  $\mathbf{c}$  de  $B$ , e  $\text{ sinal}(\mathbf{c})$  é dada por

$$\text{ sinal}(\mathbf{c}) = \begin{cases} 1, & \text{se } c_k = a_k \text{ para um número par de } k\text{'s.} \\ -1, & \text{se } c_k = a_k \text{ para um número ímpar de } k\text{'s.} \end{cases} \quad (1.2)$$

Por exemplo, seja o cubo tridimensional  $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ , temos 8 vértices. No vértice  $(b_1, b_2, b_3)$  temos uma quantidade 0, par, de  $a_k$ , então a função sinal assume o valor 1 nesse vértice. Já no vértice  $(b_1, b_2, a_2)$  temos uma quantidade 1, ímpar, de  $a_k$  e a função sinal neste vértice toma o valor  $-1$ . E assim por diante, em cada vértice que contenha um número par de  $a_k$ 's a função sinal assume o valor 1 e assume  $-1$  se o vértice contiver um número ímpar de  $a_k$ 's. Assim, a função volume aplicada nesse cubo é dada por

$$V_{H(B)} = H(b_1, b_2, b_3) - H(b_1, b_2, a_3) - H(b_1, a_2, b_3) -$$

$$-H(a_1, b_2, b_3) + H(b_1, a_2, a_3) + H(a_1, b_2, a_3) + H(a_1, a_2, b_3) - H(a_1, a_2, a_3).$$

**Definição 1.2.** *Considere  $\bar{\mathbf{R}}$  como sendo a reta real estendida,  $[-\infty, \infty]$ . Considere também  $G(-\infty) = \lim_{x \rightarrow -\infty} G(x)$  e  $G(\infty) = \lim_{x \rightarrow \infty} G(x)$ , com  $G$  uma função real. Uma função  $H : \bar{\mathbf{R}}^n \rightarrow [0, 1]$  é chamada uma função de distribuição  $n$ -variada se satisfaz as seguintes propriedades:*

1.  $H(x_1, \dots, x_n)$  é não-decrescente em cada uma das variáveis.
2.  $H(x_1, \dots, x_n)$  é contínua à direita em cada uma das variáveis.



3.  $H(\mathbf{x}) = 0$  para todo  $\mathbf{x} \in \bar{\mathbf{R}}^n$  tal que  $x_k = -\infty$  para pelo menos um  $k = 1, \dots, n$ , e  $H(\infty, \infty, \dots, \infty) = 1$
4.  $V_H(B) \geq 0$  para todo  $B = [x_1, y_1] \times \dots \times [x_k, y_k] \times \dots \times [x_n, y_n]$  com  $[x_k, y_k] \subseteq \bar{\mathbf{R}}$ .

## 1.1 Cópulas

Cópulas são funções que ligam funções de distribuições multivariadas às suas marginais. Usualmente, a cópula é definida como uma função de distribuição  $C : [0, 1]^n \rightarrow [0, 1]$ . Formalmente, podemos definir uma cópula da seguinte forma,

**Definição 1.3.** Uma cópula é uma função  $C : [0, 1]^n \rightarrow [0, 1]$  com as seguintes propriedades:

1. Para todo  $\mathbf{u} = (u_1, \dots, u_n)$  em  $[0, 1]^n$ ,

$$C(\mathbf{u}) = 0 \text{ se pelo menos uma das coordenadas de } \mathbf{u} \text{ é } 0,$$

e

$$C(\mathbf{u}) = u_k \text{ se todas as coordenadas de } \mathbf{u} \text{ são iguais a } 1 \text{ exceto } u_k.$$

2. Para todo  $\mathbf{a}$  e  $\mathbf{b}$  em  $[0, 1]^n$  tal que  $\mathbf{a} \leq \mathbf{b}$  ( $a_i \leq b_i, \forall i \in \{1, \dots, n\}$ ),

$$V_C([\mathbf{a}, \mathbf{b}]) \geq 0.$$

**Exemplo 1.1.** Considere a função  $\Pi(u, v) = uv$ , com  $u, v$  em  $[0, 1]$  temos  $\Pi(0, v) = \Pi(u, 0) = 0$  e  $\Pi(1, v) = v$  e  $\Pi(u, 1) = u$ . Seja  $u_1, u_2, v_1, v_2$  em  $[0, 1]$  tal que  $u_1 \leq u_2$  e  $v_1 \leq v_2$  temos que

$$V_C = \Pi(u_2, v_2) - \Pi(u_2, v_1) - \Pi(u_1, v_2) + \Pi(u_1, v_1)$$

$$V_C = u_2v_2 - u_2v_1 - u_1v_2 + u_1v_1 = v_2(u_2 - u_1) - v_1(u_2 - u_1) = (v_2 - v_1)(u_2 - u_1) \geq 0.$$

Portanto,  $\Pi(u, v) = uv$  é uma cópula bivariada chamada de cópula produto.

**Exemplo 1.2. (Limitantes de Fréchet)** Alguns exemplos básicos de cópulas são:

$$M^n = \min(u_1, u_2, \dots, u_n);$$

$$W^n = \max(u_1 + u_2 + \dots + u_n - n + 1, 0)$$

A função  $W^n$  não é uma cópula para  $n > 2$ , mesmo assim vale o seguinte resultado cuja demonstração pode ser encontrada em Nelsen (2006) p. 48.

**Teorema 1.1.** Se  $C$  é uma cópula, então para todo  $\mathbf{u}$  em  $[0, 1]^n$

$$W^n(\mathbf{u}) \leq C(\mathbf{u}) \leq M^n(\mathbf{u}).$$

Devido a essa relação as funções  $W^n$  e  $M^n$  são conhecidas como limitantes de Fréchet. Um teorema importante no estudo de cópulas e que faz a conexão entre cópula e função de distribuição conjunta de variáveis aleatórias e suas marginais é o Teorema de Sklar descrito a seguir e cuja prova pode ser encontrada em Sklar (1973).

**Teorema 1.2. (Teorema de Sklar)** *Seja  $H$  uma função de distribuição  $n$ -variada com marginais  $F_1, F_2, \dots, F_n$ . Então existe uma cópula  $C$  tal que para todo  $\mathbf{x}$  em  $\bar{\mathbf{R}}^n$ ,*

$$H(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n)).$$

*Se  $F_1, F_2, \dots, F_n$  são todas contínuas, então  $C$  é única, caso contrário,  $C$  é unicamente determinada em  $\text{Ran}F_1 \times \text{Ran}F_2 \times \dots \times \text{Ran}F_n$ . No sentido inverso, se  $C$  é uma cópula e  $F_1, F_2, \dots, F_n$  são funções de distribuição então  $H$  definida acima é função de distribuição  $n$ -variada com marginais  $F_1, F_2, \dots, F_n$ .  $\text{Ran}F_k$  denota o conjunto imagem de  $F_k$ ,  $k = 1, \dots, n$ .*

O corolário 1 fornece um método para se construir cópulas a partir de uma função de distribuição. Antes de enunciar o corolário definimos a função inversa generalizada.

**Definição 1.4.** *Seja  $F$  uma função de distribuição. Então uma inversa generalizada de  $F$  é uma função  $F^{(-1)}$  com domínio em  $[0, 1]^n$  tal que*

1. *Se  $t$  estiver em  $\text{Ran}F$ , então  $F^{(-1)}(t)$  é um número  $x$  em  $\bar{\mathbf{R}}$  tal que  $F(x) = t$ , isto é, para  $t$  em  $\text{Ran}F$ ,*

$$F(F^{(-1)}(t)) = t;$$

2. *Se  $t$  não estiver em  $\text{Ran}F$ , então*

$$F^{(-1)}(t) = \inf\{x | F(x) \geq t\} = \sup\{x | F(x) \leq t\} \quad (1.3)$$

*Se  $F$  é estritamente crescente, então a inversa generalizada é a inversa da função e usamos neste caso a notação  $F^{-1}$ .*

**Corolário 1.1.** *Seja  $H, C, F_1, F_2, \dots, F_n$  como no Teorema de Sklar e seja  $F_1^{(-1)}, F_2^{(-1)}, \dots, F_n^{(-1)}$  as inversas generalizadas de  $F_1, F_2, \dots, F_n$ , respectivamente. Então para qualquer  $\mathbf{u} = (u_1, \dots, u_n)$  em  $[0, 1]^n$ ,*

$$C(u_1, u_2, \dots, u_n) = H(F_1^{(-1)}(u_1), F_2^{(-1)}(u_2), \dots, F_n^{(-1)}(u_n)). \quad (1.4)$$

Cópula é uma função uniformemente contínua em seu domínio, como mostra o teorema a seguir, cuja prova pode ser encontrada em Nelsen (2006) p. 46, isso significa que  $C$  possui derivadas de toda ordem em seu domínio.

**Teorema 1.3.** *Seja  $C$  uma cópula. Então para  $\mathbf{u}$  e  $\mathbf{v}$  em  $[0, 1]^n$ ,*

$$|C(\mathbf{v}) - C(\mathbf{u})| \leq \sum_{k=1}^n |v_k - u_k|. \quad (1.5)$$

Muito da utilidade de cópulas no estudo de estatísticas não-paramétricas deriva do fato de que cópulas são invariantes a transformações monótonas crescentes nas variáveis. O teorema 1.4 mostra essa propriedade,

**Teorema 1.4.** *Sejam  $X$  e  $Y$  variáveis aleatórias contínuas com cópula  $C_{XY}$ . Se  $\alpha$  e  $\beta$  são funções estritamente crescentes em  $\text{Ran}X$  e  $\text{Ran}Y$ , respectivamente, então  $C_{\alpha(X)\beta(Y)} = C_{XY}$*

Veja a prova em Nelsen (2006) p. 25. Na literatura existem diversas famílias de cópulas, sendo as principais: cópula gaussiana, t-student, arquimedianas, vine cópulas, cópulas extremas. Neste trabalho será dada ênfase às cópulas arquimedianas bivariadas. Ou seja, de agora em diante cópula significará somente cópula bivariada arquimediana.

## 1.2 Cópulas Arquimedianas

As cópulas arquimedianas encontram um grande número de aplicações pela facilidade com que podem ser construídas, por possuir uma grande variedade de famílias nessa classe de cópulas e pelas boas propriedades que os membros dessa família possuem.

**Definição 1.5.** Uma cópula Arquimediana é uma função  $C : [0, 1]^2 \rightarrow [0, 1]$ , dada por  $C(u, v) = \varphi^{[-1]}(\varphi(u) + \varphi(v))$  em que  $\varphi$  é uma função contínua convexa estritamente decrescente de  $[0, 1] \rightarrow [0, \infty]$  tal que  $\varphi(1) = 0$  em que  $\varphi^{[-1]}$  denota a "pseudo-inversa" de  $\varphi$ , isto é,

$$\varphi^{[-1]}(t) = \begin{cases} \varphi^{-1}(t) & \text{se } t \in [0, \varphi(0)) \\ 0 & \text{se } t \geq \varphi(0) \end{cases}$$

Quando  $\varphi(0) = \infty$ ,  $\varphi$  é dito ser estrito e  $\varphi^{[-1]} \equiv \varphi^{-1}$ .  $\varphi$  é chamado de gerador.

A tabela a seguir mostra as principais cópulas arquimedianas e seus respectivos geradores.

Tabela 1.1: Principais Cópulas arquimedianas e seus respectivos geradores.

Cópula	$C_\theta(u, v)$	$\varphi_\theta(t)$	$\theta \in$
Clayton	$[\max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-\frac{1}{\theta}}$	$\frac{t^{-\theta} - 1}{\theta}$	$[-1, \infty] \setminus \{0\}$
Gumbel	$\exp(-[(-\ln u)^\theta + (-\ln v)^\theta]^{\frac{1}{\theta}})$	$(\ln t)^{-\theta}$	$[1, \infty)$
Frank	$-\frac{1}{\theta} \ln \left( 1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$	$-\ln \frac{e^{-\theta t} - 1}{e^{-\theta} - 1}$	$(-\infty, \infty) \setminus \{0\}$
Joe	$1 - [(1 - u)^\theta + (1 - v)^\theta - (1 - u)^\theta(1 - v)^\theta]^{\frac{1}{\theta}}$	$-\ln[1 - (1 - t)^\theta]$	$[1, \infty)$

As figuras a seguir mostram curvas de níveis para diferentes valores do parâmetro  $\theta$  das cópulas arquimedianas mostradas na tabela acima, com marginais F e G, tais que ambas são funções de distribuição de uma variável aleatória  $N(0, 1)$ .

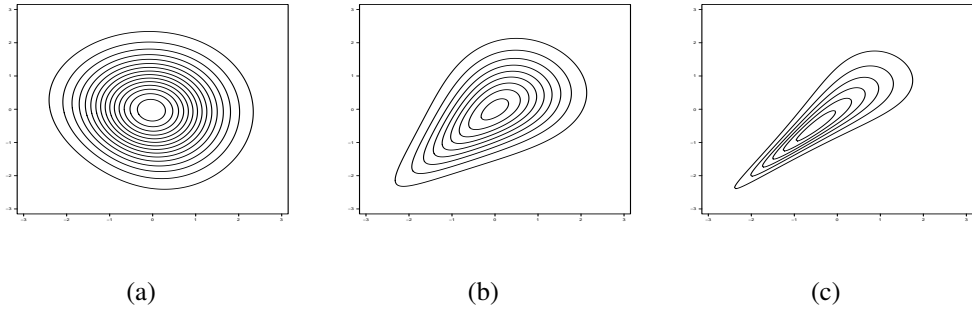


Figura 1.1: Curvas de nível da densidade da cópula de Clayton para  $\theta = -0.1$ , (a),  $\theta = 1.0$ , (b), e  $\theta = 3.0$ , (c).

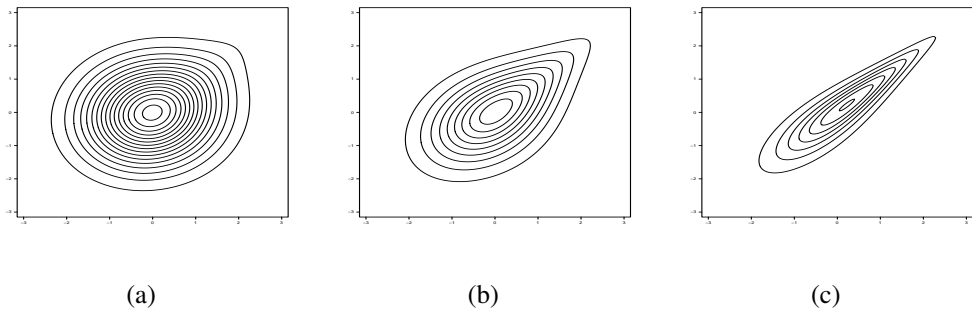


Figura 1.2: Curvas de nível da densidade da cópula de Gumbel para  $\theta = 1.1$ , (a),  $\theta = 1.5$ , (b), e  $\theta = 3.0$ , (c).

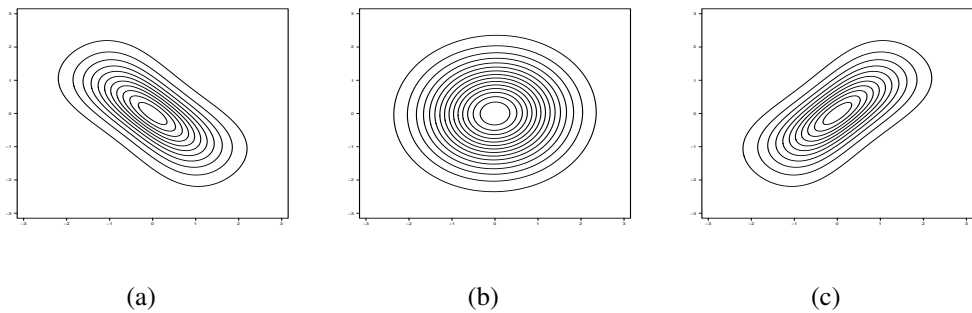


Figura 1.3: Curvas de nível da densidade da cópula de Frank para  $\theta = -5.0$ , (a),  $\theta = 0.1$ , (b), e  $\theta = 5.0$ , (c).

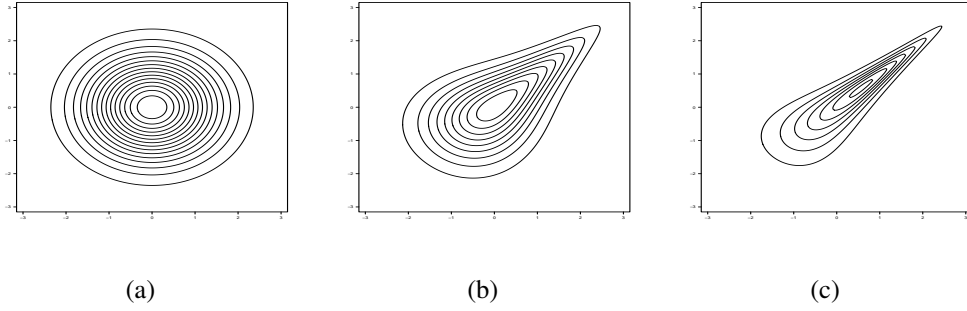


Figura 1.4: Curvas de nível da densidade da cópula de Joe para  $\theta = 1.0$ , (a),  $\theta = 2.0$ , (b), e  $\theta = 4.0$ , (c).

Algumas propriedades algébricas que caracterizam a família de cópulas arquimedianas são dadas a seguir:

Seja  $C$  uma cópula arquimediana com gerador  $\varphi$ . Então:

1.  $C$  é simétrica, isto é,  $C(u, v) = C(v, u)$  para todo  $u$  e  $v$  em  $[0, 1]$ ;
2.  $C$  é associativa, isto é,  $C(C(u, v), w) = C(u, C(v, w))$  para todo  $u$  e  $v$  em  $[0, 1]$ ;
3. Dada qualquer constante  $\alpha > 0$ , tem-se que  $\alpha\varphi$  é também um gerador de  $C$ .

Uma função bastante útil em processos de estimação de cópulas arquimedianas é a função de Kendall, denotada por  $K_C$  que apresenta uma expressão fechada para cópulas arquimedianas conforme teorema 1.5.

**Teorema 1.5.** *Seja  $C$  uma cópula arquimediana com gerador  $\varphi$  e seja*

$$K_C(t) = V_C((u, v) \in [0, 1]^n | C(u, v) \leq t),$$

então para todo  $t$  em  $[0, 1]$ , temos

$$K_C(t) = t - \frac{\varphi(t)}{\varphi'(t^+)}, \quad (1.6)$$

em que  $\varphi'(t^+) = \lim_{u \rightarrow t} \varphi'(u)$ , com  $u > t$ .

A prova desse teorema pode ser vista em Nelsen (2006), p.127.

Os seguintes resultados fornecem uma metodologia para gerar um vetor aleatório  $(U, V)$  tal que sua função de distribuição conjunta é uma cópula arquimediana  $C$  com gerador  $\varphi$ .

**Corolário 1.2.** *Se  $(U, V)$  tem função de distribuição  $C$ , com  $C$  uma cópula arquimediana gerada por  $\varphi$ , então a variável aleatória  $C(U, V)$  tem função de distribuição igual a  $K_C(t)$ .*

**Teorema 1.6.** *sob as hipóteses do corolário 1.2, a função de distribuição conjunta  $H(s, t)$  das variáveis aleatórias  $S = \frac{\varphi(U)}{\varphi(U) + \varphi(V)}$  e  $T = C(U, V)$  é dada por  $H(s, t) = sK_C(t)$  para todo  $(s, t)$  em  $[0, 1]^2$ . Consequentemente,  $S$  e  $T$  são independentes e  $S$  é uniformemente distribuído em  $[0, 1]$ .*

Então, um algoritmo para gerar  $(U, V)$  é:

**Algoritmo 1.1.**

- *Simular duas variáveis uniformes,  $U(0, 1)$ ,  $s$  e  $q$ ;*
- *Calcular  $t = K_C^{-1}(q)$ ;*
- *Fazer  $u = \varphi^{[-1]}(s\varphi(t))$ ; e  $v = \varphi^{[-1]}((1 - s)\varphi(t))$ .*

Seja  $C_k(u_1, \dots, u_k) = C(u_1, \dots, u_k, 1 \dots 1)$ ,  $k = 2, \dots, n - 1$ , a marginal de dimensão  $k$  de  $C$ , com  $C_1(u_1) = u_1$  e  $C_n(u_1, \dots, u_n) = C(u_1, \dots, u_k)$ , em que  $U_1, \dots, U_n$  é um vetor aleatório com distribuição  $C$ . Então, a distribuição condicional de  $C_k$  dado os valores de  $U_1, \dots, U_{k-1}$ , é dada por

$$C_k(u_k | u_1, \dots, u_{k-1}) = \mathbb{P}(U_k \leq u_k | U_1 = u_1, \dots, U_{k-1} = u_{k-1})$$

$$= \frac{\partial^{k-1} C_k(u_1, \dots, u_k)}{\partial u_1 \dots \partial u_{k-1}} / \frac{\partial^{k-1} C_{k-1}(u_1, \dots, u_{k-1})}{\partial u_1 \dots \partial u_{k-1}}.$$

Assim, outra forma de gerar um vetor aleatório  $(U_1, \dots, U_n)$ , cuja distribuição conjunta é  $C$ , é dada abaixo:

**Algoritmo 1.2.**

- *Simular uma variável aleatória  $u_1$  de  $U(0, 1)$ .*
- *Simular uma variável aleatória  $u_2$  de  $C_2(\cdot | u_1)$ .*
- $\vdots$
- *Simular uma variável aleatória  $u_n$  de  $C_n(\cdot | u_1, \dots, u_{n-1})$ .*

Para simular  $u_k$  de  $C_k(\cdot | u_1, \dots, u_{k-1})$ , deve-se simular  $q$  de  $U(0, 1)$  e fazer  $u_k = C_k^{-1}(q | u_1, \dots, u_{k-1})$ .

### 1.3 Dependência

Esta seção trata das principais medidas de dependência entre variáveis aleatórias que são coeficiente de correlação linear de Pearson,  $\tau$  de Kendall e  $\rho$  de Spearman, além de dependência caudal. Na literatura o coeficiente de correção linear de Pearson é uma das medidas de dependência mais conhecidas, definida a seguir.

**Definição 1.6.** *Seja  $(X, Y)$  um vetor de variáveis aleatórias com variância finita. O coeficiente de correlação linear de Pearson é dado por*

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}},$$

em que  $Cov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$  é a covariância de  $(X, Y)$ , e  $Var(X)$ ,  $Var(Y)$  são as variâncias de  $X$  e  $Y$ , respectivamente.

Algumas das desvantagens de se usar a correlação de Pearson é que essa medida é fortemente influenciada por *outliers* e a não-linearidade dos dados. Então devido a essas desvantagens outras medidas de dependência são mais usadas.

**Definição 1.7. (Tau de Kendall)** *Sejam  $(X_1, Y_1)$  e  $(X_2, Y_2)$  vetores aleatórios independentes e identicamente distribuídos com função de distribuição conjunta  $H$ . Então a versão populacional do coeficiente tau de Kendall é expressa pela probabilidade de concordância menos a probabilidade de discordância,*

$$\tau = \tau_{X,Y} = P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0] \quad (1.7)$$

Essa medida é invariante a transformações monótonas crescentes nos vetores aleatórios como evidenciado no seguinte expressão, já que a função cópula é invariante a transformações monótonas crescentes (teorema 4), veja Nelsen (2006) e Embrechts et al. (2001):

$$\tau = 4 \int \int_{I^2} C(u, v) dC(u, v) - 1 = 4\mathbb{E}C(U, V) - 1, \quad (1.8)$$

em que  $I^2 = [0, 1] \times [0, 1]$ . Pelo teorema 1.5, nas cópulas Arquimedianas temos que o coeficiente tau de Kendall pode ser calculado da seguinte forma:

**Teorema 1.7.** *Sejam  $X$  e  $Y$  variáveis aleatórias com cópula arquimediana  $C$  gerada por  $\varphi$ . A versão populacional do coeficiente  $\tau$  de Kendall é dada pelo seguinte teorema, Nelsen (2006) p. 127.*

$$\tau = 1 + 4 \int_0^1 \frac{\varphi(t)}{\varphi'(t)} dt \quad (1.9)$$

**Definição 1.8. ( $\rho$  de Spearman)** *Sejam  $(X_1, Y_1)$ ,  $(X_2, Y_2)$  e  $(X_3, Y_3)$  três vetores aleatórios independentes e identicamente distribuídos com função de distribuição conjunta  $H$ , cópula  $C$  e ambas as marginais iguais a  $F$  e  $G$ . Então a versão populacional do  $\rho$  de Spearman é expressa por, Nelsen (2006) e Embrechts et al. (2001),*

$$\rho_s = 3P[(X_1 - X_2)(Y_1 - Y_3) > 0] - P[(X_1 - X_2)(Y_1 - Y_3) < 0] = 12 \int \int_{I^2} C(u, v) dudv - 3.$$

Os valores de  $\tau$  e  $\rho_s$  estão relacionados, mas essa relação muda em cada família de cópula. Mas vale a seguinte desigualdade universal para essas duas medidas, cuja prova pode ser vista em Nelsen (2006) p. 175.

**Teorema 1.8.** *Sejam  $X$  e  $Y$  variáveis aleatórias contínuas. Então*

$$-1 \leq 3\tau - 2\rho_s \leq 1.$$

Para dados bivariados relativos a eventos extremos,  $(X, Y) \sim H$  e  $(F(X), G(Y)) \sim C$  cópula extremal, por exemplo a cópula de Gumbel, o conhecimento do comportamento caudal dos dados é de grande importância. A função a seguir permite investigar a dependência caudal de dados bivariados.

**Definição 1.9. (Dependência caudal)** Sejam  $X$  e  $Y$  variáveis aleatórias contínuas com função de distribuição  $F$  e  $G$ , respectivamente. O parâmetro de dependência de cauda superior  $\lambda_U$  é o limite (se existir) da probabilidade condicional de  $Y$  ser maior que o  $100t$ -ésimo percentil de  $G(y)$  dado que  $X$  é maior ou igual ao  $100t$ -ésimo percentil de  $F(x)$  quando  $t$  se aproxima de 1,

$$\lambda_U = \lim_{t \rightarrow 1^-} P[Y > G^{(-1)}(t) | X > F^{(-1)}(t)]. \quad (1.10)$$

De forma similar, o parâmetro de dependência de cauda inferior  $\lambda_L$  é o limite (se existir) da probabilidade condicional de  $Y$  ser menor ou igual que o  $100t$ -ésimo percentil de  $G(y)$  dado que  $X$  é menor ou igual ao  $100t$ -ésimo percentil de  $F(x)$  quando  $t$  se aproxima de 0,

$$\lambda_L = \lim_{t \rightarrow 0^+} P[Y \leq G^{(-1)}(t) | X \leq F^{(-1)}(t)]. \quad (1.11)$$

Estes parâmetros dependem apenas da cópula de  $X$  e  $Y$ , e podem ser escritos como nas fórmulas a seguir, Nelsen (2006).

$$\lambda_U = \lim_{t \rightarrow 1^-} \frac{1 - 2t + C(t, t)}{1 - t}. \quad (1.12)$$

e

$$\lambda_L = \lim_{t \rightarrow 0^+} \frac{C(t, t)}{t}. \quad (1.13)$$

A Tabela 1.2 mostra a relação entre o parâmetro da cópula e o coeficiente  $\tau$  de Kendall e coeficientes caudais, para as cópulas arquimedianas apresentadas na Tabela 1.1.

Tabela 1.2: Principais Cópulas arquimedianas e medidas de dependência

Cópula	$C_\theta(u, v)$	$\tau$	$\lambda_L$	$\lambda_U$
Clayton	$[\max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-\frac{1}{\theta}}$	$\frac{\theta}{\theta + 2}$	$2^{-\frac{1}{\theta}}$	0
Gumbel	$\exp(-[(-\ln u)^\theta + (-\ln v)^\theta]^{\frac{1}{\theta}})$	$\frac{\theta}{\theta - 1}$	0	$2 - 2^{\frac{1}{\theta}}$
Frank	$-\frac{1}{\theta} \ln \left( 1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$	$\frac{1 - 4(1 - D_1(\theta))^*}{\theta}$	0	0
Joe	$1 - [(1 - u)^\theta + (1 - v)^\theta - (1 - u)^\theta(1 - v)^\theta]^{\frac{1}{\theta}}$	Não tem forma fechada	0	$2 - 2^{\frac{1}{\theta}}$

$$*D_k(x) = \frac{k}{x^k} \int_0^x \frac{t^k}{e^t - 1} dt \text{ é a função de Debye.}$$

## 1.4 Mistura de Distribuições

Uma função de densidade de probabilidade  $h$  de uma variável aleatória  $X$  é um modelo de mistura finita de  $k$  densidades (componentes) de uma família  $\mathcal{F}$ , se para  $f_1, \dots, f_k \in \mathcal{F}$ ,

$$h(x; \theta) = \sum_{i=1}^k \pi_i f_i(x, \theta_i), \quad (1.14)$$



em que os  $\pi_i$ 's são os pesos da mistura tal que  $\sum_{i=1}^k \pi_i = 1$ . O modelo de mistura densidades é bastante utilizado para modelar dados de uma população que apresenta assimetria ou heterogeneidade de subpopulações.

Exemplos de mistura de densidades:

a) mistura de Normais

$$h(x; \theta) = \pi_1 f_1(x, \mu_1, \sigma_1) + \pi_2 f_2(x, \mu_2, \sigma_2),$$

sendo  $\theta = (\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2)$ ,  $f_1$  e  $f_2$  densidades de uma distribuição Normal( $\mu, \sigma^2$ ).

b) mistura de distribuições de valor extremo generalizadas (GEV), definidas na equação (5.1).

$$h(x; \theta) = \pi_1 g_1(x, \mu_1, \sigma_1, \alpha_1) + \pi_2 g_2(x, \mu_2, \sigma_2, \alpha_2),$$

sendo  $\theta = (\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2, \alpha_1, \alpha_2)$ ,  $g_1$  e  $g_2$  densidades de uma distribuição GEV( $\mu, \sigma, \alpha$ ).

As figuras abaixo mostram alguns exemplos de mistura de distribuições.

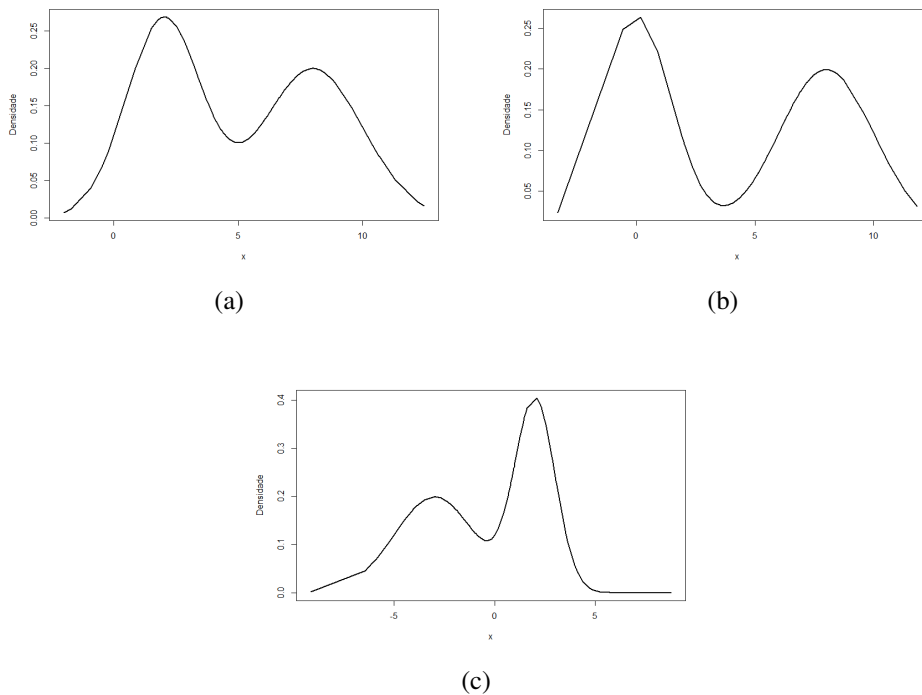


Figura 1.5: Gráficos de diferentes misturas de distribuições Normais: a)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 2, \mu_2 = 8, \sigma_1 = 2, \sigma_2) = 3$ ; b)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 0, \mu_2 = 8, \sigma_1 = 1, \sigma_2) = 2$  e c)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = -4, \mu_2 = 2, \sigma_1 = 3, \sigma_2) = 1$ .

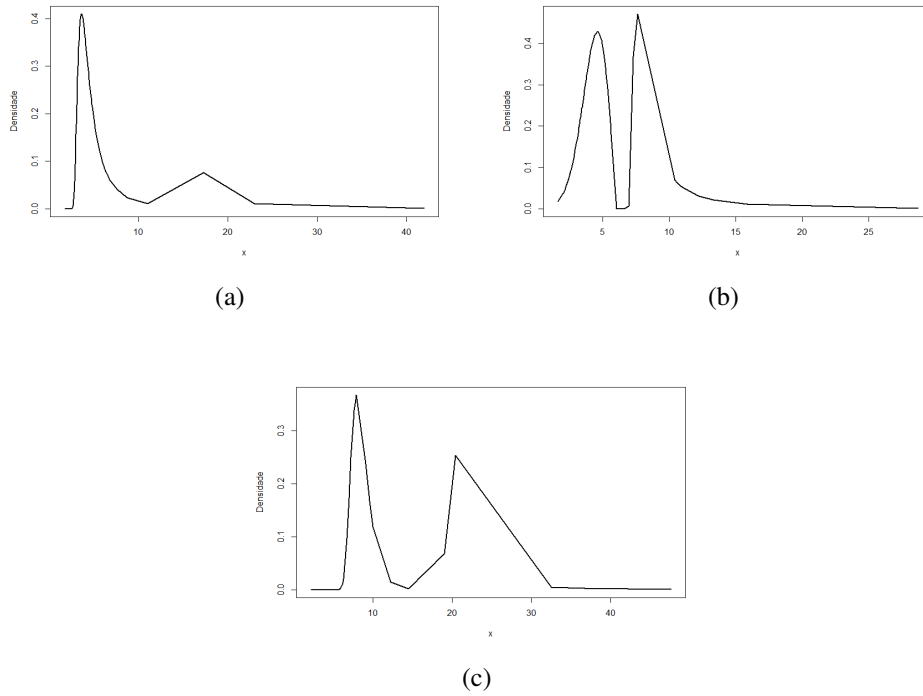


Figura 1.6: Gráficos de diferentes misturas de distribuições GEV: a)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 6, \mu_2 = 15, \sigma_1 = 3, \sigma_2 = 10, \alpha_1 = 0.5, \alpha_2 = 0.8)$ ; b)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 4, \mu_2 = 8, \sigma_1 = 3, \sigma_2 = 3, \alpha_1 = -0.8, \alpha_2 = 0.6)$  e c)  $\theta = (\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = 10, \mu_2 = 25, \sigma_1 = 2, \sigma_2 = 10, \alpha_1 = 0.0, \alpha_2 = 0.8)$ .

# Capítulo 2

## Mistura de Cópulas

Modelos de mistura de distribuições são muito vantajosos por sua extrema flexibilidade para modelar diversos fenômenos aleatórios nas mais diversas áreas, como por exemplo, em finanças, astronomia, biologia, genética, engenharia e medicina, ver McLachlan e Peel, 2000. Neste trabalho é estudado o modelo de mistura de cópulas no caso bivariado. A mistura de cópulas permite a obtenção de estruturas de dependência mais flexíveis do que uma única cópula, possibilitando a junção de características distintas de cópulas numa mesma cópula, Arakelian e Karlis (2014).

### 2.1 Modelo

Dadas  $C_1$  e  $C_2$  duas cópulas e os números  $p_1, p_2 \in [0, 1]$  tais que  $p_1 + p_2 = 1$ , como definido em Arakelian e Karlis (2014), o modelo de mistura de cópulas com componentes  $C_1$  e  $C_2$ , é dado por

$$C(u, v) = pC_1(u, v; \theta_1) + (1 - p)C_2(u, v; \theta_2), \quad (2.1)$$

é de fato uma cópula, pois

1. Para todo  $u, v$  em  $[0, 1]$ ,

$$pC_1(u, 0; \theta_1) + (1 - p)C_2(u, 0; \theta_2) = p \cdot 0 + (1 - p) \cdot 0 = 0 = pC_1(0, v; \theta_1) + (1 - p)C_2(0, v; \theta_2)$$

e

$$pC_1(u, 1; \theta_1) + (1 - p)C_2(u, 1; \theta_2) = pu + (1 - p)u = u$$

e

$$pC_1(1, v; \theta_1) + (1 - p)C_2(1, v; \theta_2) = pv + (1 - p)v = v;$$

2. Para todo  $u_1, u_2, v_1, v_2$  em  $[0, 1]$  tal que  $u_1 \leq u_2$  e  $v_1 \leq v_2$ ,

$$pC_1(u_2, v_2; \theta_1) + (1 - p)C_2(u_2, v_2; \theta_2) - [pC_1(u_2, v_1; \theta_1) + (1 - p)C_2(u_2, v_1; \theta_2)]$$

$$= p[C_1(u_2, v_2; \theta_1) - C_1(u_2, v_1; \theta_1)] + (1 - p)[C_2(u_2, v_2; \theta_2) - C_2(u_2, v_1; \theta_2)];$$

e

$$\begin{aligned} & -[pC_1(u_1, v_2; \theta_1) + (1 - p)C_2(u_1, v_2; \theta_2)] + pC_1(u_1, v_1; \theta_1) + (1 - p)C_2(u_1, v_1; \theta_2) \\ & = p[-C_1(u_1, v_2; \theta_1) + C_1(u_1, v_1; \theta_1)] + (1 - p)[-C_2(u_1, v_2; \theta_2) + C_2(u_1, v_1; \theta_2)]. \end{aligned}$$

Juntando os termos que multiplicam  $p$  temos

$$p[C_1(u_2, v_2; \theta_1) - C_1(u_2, v_1; \theta_1) - C_1(u_1, v_2; \theta_1) + C_1(u_1, v_1; \theta_1)] \geq 0;$$

E fazendo o mesmo para os termos que contêm  $(1 - p)$  temos

$$(1 - p)[C_1(u_2, v_2; \theta_1) - C_1(u_2, v_1; \theta_1) - C_1(u_1, v_2; \theta_1) + C_1(u_1, v_1; \theta_1)] \geq 0.$$

O que demonstra que a mistura de cópulas é também uma cópula.

## 2.2 Estrutura de Dependência de Mistura de Cópulas

Na mistura de cópulas o parâmetro do modelo,  $\theta$ , nem sempre é de fácil interpretação. Para contornar isto, supondo que as marginais são contínuas, pode se interpretar  $\theta$  com relação ao coeficiente  $\tau$  de Kendall, que é limitado no intervalo  $[-1, 1]$ , tendo-se assim uma medida de dependência comparável. O coeficiente  $\tau$  de Kendall da mistura de cópulas é expresso por

$$\begin{aligned} \tau_C &= 4 \int \int_{I^2} (pC_1(u, v; \theta_1) + (1 - p)C_2(u, v; \theta_2))(pdC_1(u, v; \theta_1) + (1 - p)dC_2(u, v; \theta_2)) - 1 \\ &= 4 \int \int_{I^2} p^2 C_1(u, v; \theta_1) dC_1(u, v; \theta_1) + 4 \int \int_{I^2} (1 - p)^2 C_2(u, v; \theta_2) dC_2(u, v; \theta_2) + \\ &+ 4 \int \int_{I^2} p(1 - p) C_1(u, v; \theta_1) dC_2(u, v; \theta_2) + 4 \int \int_{I^2} p(1 - p) C_2(u, v; \theta_2) dC_1(u, v; \theta_1) - 1 \\ &= 4 \int \int_{I^2} p^2 C_1(u, v; \theta_1) dC_1(u, v; \theta_1) - p^2 + p^2 + \\ &+ 4 \int \int_{I^2} (1 - p)^2 C_2(u, v; \theta_2) dC_2(u, v; \theta_2) - (1 - p)^2 + (1 - p)^2 + \\ &+ 2[4 \int \int_{I^2} p(1 - p) C_1(u, v; \theta_1) dC_2(u, v; \theta_2) - p(1 - p) + p(1 - p)] - 1 \\ &= p^2 \tau_{C_1} + (1 - p)^2 \tau_{C_2} + 2p(1 - p)Q + [p^2 + (1 - p)^2 + 2p(1 - p) - 1] \\ &= p^2 \tau_{C_1} + (1 - p)^2 \tau_{C_2} + 2p(1 - p)Q, \end{aligned} \tag{2.2}$$

em que  $Q = Q(C_1, C_2) = 4 \int \int_{I^2} C_2(u, v; \theta_2) dC_1(u, v; \theta_1) - 1$  é a função de concordância de-

finida para as cópulas  $C_1(u, v; \theta_1)$  e  $C_2(u, v; \theta_2)$  e  $\tau_{C_1}, \tau_{C_2}$  são os coeficientes  $\tau$  correspondentes a cada uma das cópulas, não tendo forma fechada para as cópulas consideradas neste trabalho e sendo, então, estimadas numericamente. Nelsen(2006), mostra que  $Q(C_1, C_2) = Q(C_2, C_1)$ . Ou seja, podemos ver que, mesmo que cada uma das cópulas componentes sejam da independência, a mistura pode sugerir dependência. O valor do  $\rho$  de Spearman é dado por

$$\begin{aligned}
\rho_C &= 12 \int \int_{I^2} (pC_1(u, v; \theta_1) + (1-p)C_2(u, v; \theta_2)) dudv - 3 \\
&= 12 \int \int_{I^2} pC_1(u, v; \theta_1) dudv + 12 \int \int_{I^2} (1-p)C_2(u, v; \theta_2) dudv - 3 \\
&= p[12 \int \int_{I^2} C_1(u, v; \theta_1) dudv - 3] + 3p + (1-p)[12 \int \int_{I^2} C_2(u, v; \theta_2) dudv - 3] + \\
&3(1-p) - 3 = p\rho_{C_1} + (1-p)\rho_{C_2}.
\end{aligned} \tag{2.3}$$

De (2.3), pode-se observar que o  $\rho$  da combinação convexa das cópulas é uma combinação convexa dos  $\rho$  de cada uma das cópulas. Por outro lado, a dependência caudal da mistura de cópulas é dada pelas expressões a seguir, dependência de cauda inferior

$$\begin{aligned}
\lambda_L &= \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} \\
&= \lim_{u \rightarrow 0^+} \frac{pC_1(u, v; \theta_1) + (1-p)C_2(u, v; \theta_2)}{u} \\
&= \lim_{u \rightarrow 0^+} \frac{pC_1(u, v; \theta_1)}{u} + \lim_{u \rightarrow 0^+} \frac{(1-p)C_2(u, v; \theta_2)}{u} \\
&= p\lambda_L^{\theta_1} + (1-p)\lambda_L^{\theta_2},
\end{aligned} \tag{2.4}$$

e dependência de cauda superior

$$\begin{aligned}
\lambda_U &= \lim_{u \rightarrow 1^-} \frac{1 - 2u + C(u, u)}{1 - u} \\
&= \lim_{u \rightarrow 1^-} \frac{1 - 2u + pC_1(u, v; \theta_1) + (1-p)C_2(u, v; \theta_2)}{1 - u} \\
&= \lim_{u \rightarrow 1^-} \frac{1 - 2u + C_2(u, v; \theta_2) + p[C_1(u, v; \theta_1) - C_2(u, v; \theta_2)]}{1 - u} \\
&= \lim_{u \rightarrow 1^-} \frac{1 - 2u + C_2(u, v; \theta_2)}{1 - u} + \lim_{u \rightarrow 1^-} \frac{p[1 - 2u + C_1(u, v; \theta_1) - (1 - 2u + C_2(u, v; \theta_2))]}{1 - u} \\
&= \lambda_U^{\theta_2} + p\lambda_U^{\theta_1} - p\lambda_U^{\theta_2} = p\lambda_U^{\theta_1} + (1-p)\lambda_U^{\theta_2}.
\end{aligned} \tag{2.5}$$

De (2.4) e (2.5), tem-se que o índice de dependência caudal inferior e superior, respectivamente, da mistura de cópulas é a combinação convexa dos índices das componentes da mistura. Isso permite a formação de uma nova cópula contendo dependência de cauda inferior e superior.

As figuras abaixo mostram contornos de curvas de níveis modelos de mistura de cópulas ilustrando os diversos padrões de dependência que se pode obter a partir de estruturas de pendências

distintas.

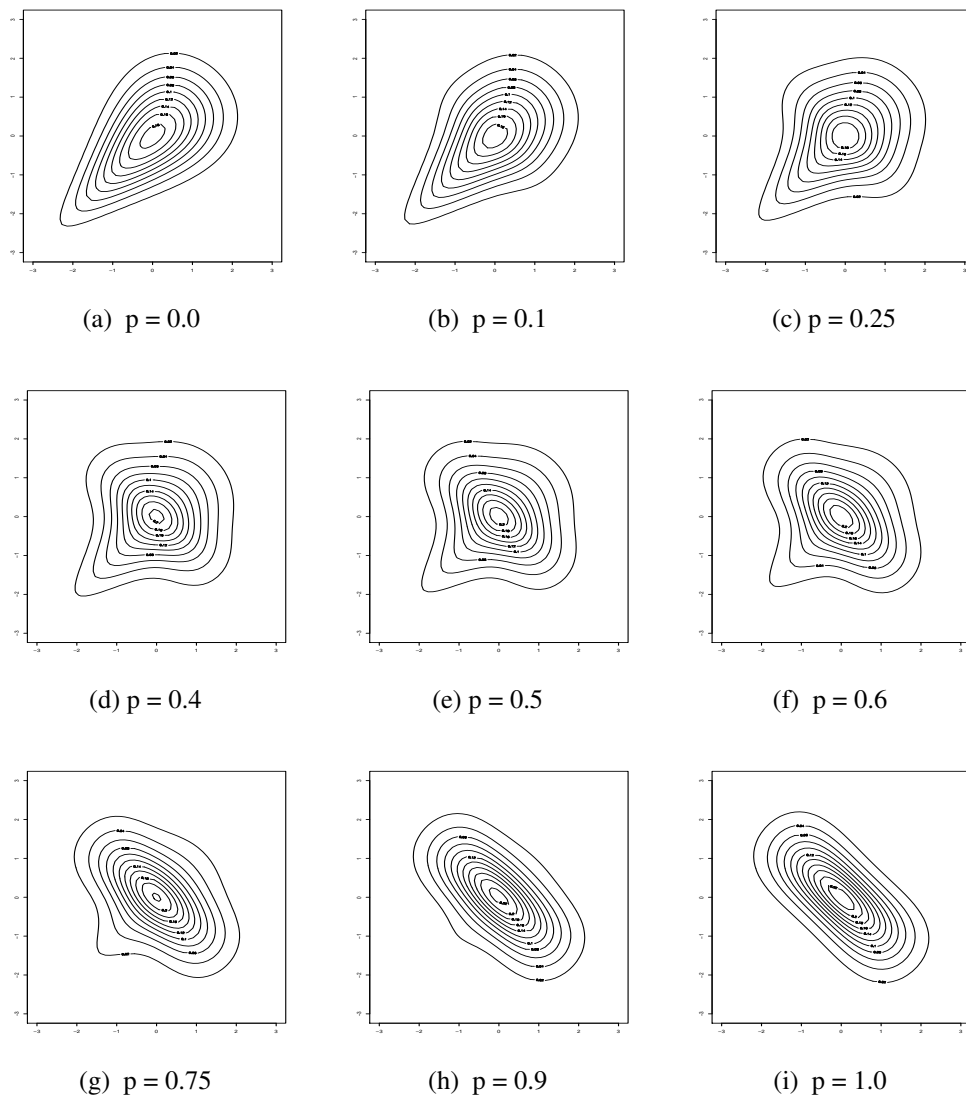


Figura 2.1: Curvas de níveis para mistura de cópula de Frank para  $\theta_1 = -5.0$  e cópula de Clayton para  $\theta_2 = 1.0$ , e diferentes valores de  $p$  e marginais normais padrões ( $C(u, v) = p * Frank + (1 - p) * Clayton$ ).

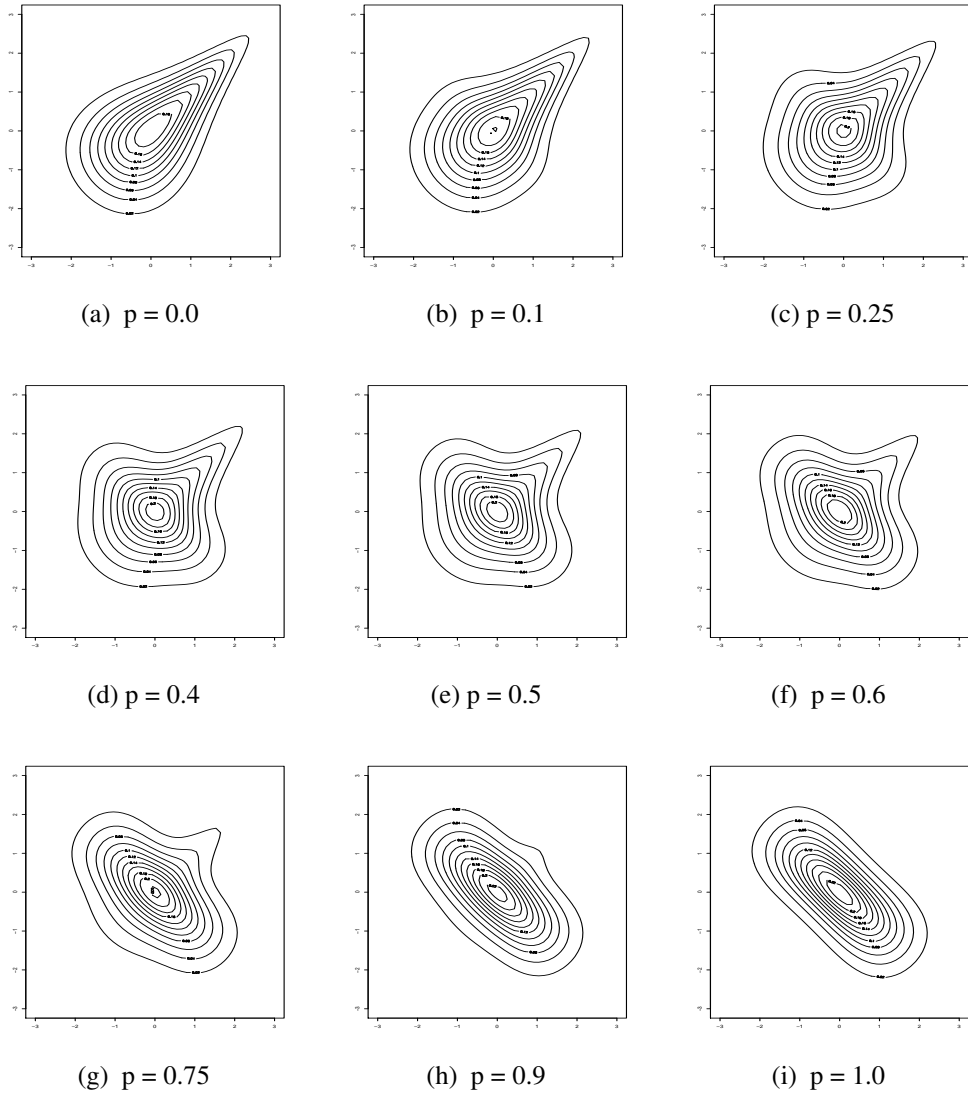


Figura 2.2: Curvas de níveis para mistura de cópula de Frank para  $\theta_1 = -5.0$  e cópula de Joe para  $\theta_2 = 2.0$ , e diferentes valores de  $p$  e marginais normais padrões ( $C(u, v) = p * Frank + (1 - p) * Joe$ ).

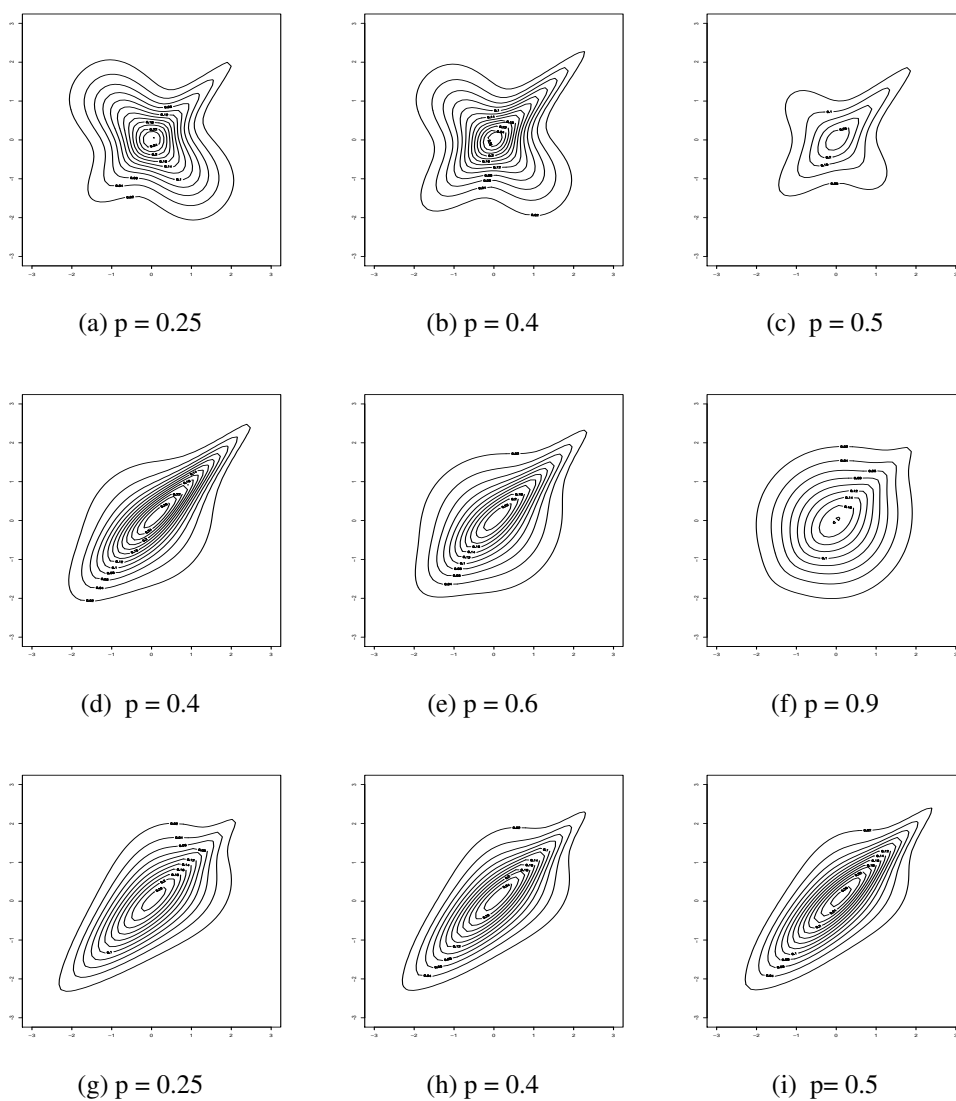


Figura 2.3: Curvas de níveis para diferentes misturas de cópulas: a-c)  $p * Gumbel(\theta_1 = 3.0) + (1 - p) * Frank(\theta_2 = -5.0)$ , d-f)  $p * Joe(\theta_1 = 1.1) + (1 - p) * Gumbel(\theta_2 = 3.0)$ , g-i)  $p * Gumbel(\theta_1 = 3.0) + (1 - p) * Clayton(\theta_2 = 1.1)$ , todas com marginais normais padrões



É importante ressaltar que além do papel da estrutura de dependência, a distribuição marginal dos dados influenciam no formato das curvas de níveis. As próximas figuras correspondem a mistura de cópulas, sendo que cada distribuição marginal é a mistura de distribuições GEV, conforme definido na seção 1.4. Note que, a mistura das distribuições marginais permite a separação dos *clusters*.

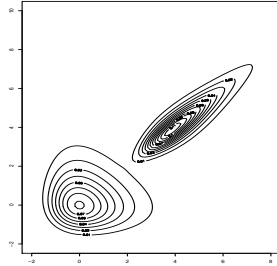


Figura 2.4: Curvas de níveis para mistura de cópulas Clayton e Gumbel:  $p * Clayton(\theta_1 = -0.1) + (1 - p) * Gumbel(\theta_2 = 3.0)$ ,  $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ). Veja definição da distribuição Gumbel na equação (5.1), com  $\alpha = 0$ .

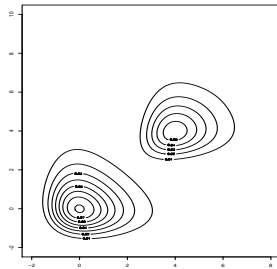


Figura 2.5: Curvas de níveis para mistura de cópulas Clayton e Gumbel:  $p * Clayton(\theta_1 = -0.1) + (1 - p) * Gumbel(\theta_2 = 1.1)$ ,  $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ).

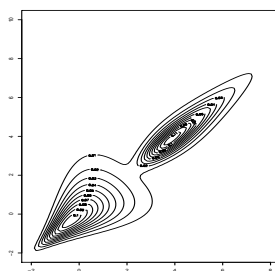


Figura 2.6: Curvas de níveis para mistura de cópulas Clayton e Gumbel:  $p * Clayton(\theta_1 = 1.0) + (1-p) * Gumbel(\theta_2 = 3.0)$ ,  $p = 0.6$ , marginais da Clayton com distribuição Gumbel( $\mu = 0, \sigma = 1$ ) e marginais da cópula Gumbel com distribuição Gumbel( $\mu = 4, \sigma = 1$ ).

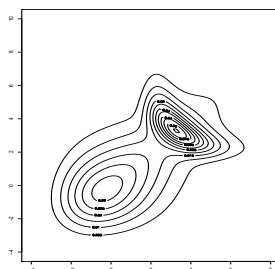


Figura 2.7: Curvas de níveis para mistura de cópulas Joe e Frank:  $p * Joe(\theta_1 = 2.0) + (1-p) * Frank(\theta_2 = -5.0)$ ,  $p = 0.75$ , marginais da Joe com distribuição Gumbel( $\mu = 3, \sigma = 2$ ) e marginais da cópula Frank com distribuição Gumbel( $\mu = 3, \sigma = 1$ ).

# Capítulo 3

## Estimação

Esta seção trata dos procedimentos de estimação dos parâmetros da mistura de cópulas e da matriz de informação de Fisher para os mesmos. Para a contextualização do modelo de mistura, considere um vetor aleatório  $(X, Y)$ , em que suas observações advêm de dois *clusters*, formando uma mistura com distribuição conjunta dada pela equação (2.1), com  $u = F_X(x)$  e  $v = G_Y(y)$ , sendo  $F$  e  $G$  as funções de distribuição de  $X$  e  $Y$ , respectivamente.

Para fazer a estimação dos parâmetros do modelo, devemos considerar a densidade da mistura de cópulas. Denotando  $c(u, v)$  como a densidade da mistura, que é obtida ao calcular a derivada de (2.1) e é dada por:

$$c(u, v) = pc_1(u, v; \theta_1) + (1 - p)c_2(u, v; \theta_2). \quad (3.1)$$

### 3.1 Estimação Por Máxima Verossimilhança

Sejam  $X$  e  $Y$  variáveis aleatórias com funções de densidade marginais  $f(x; \phi_1)$  e  $g(y; \phi_2)$ , respectivamente, e funções de distribuição acumulada  $F(x; \phi_1)$  e  $G(y; \phi_2)$ , respectivamente. Seja  $H$  a distribuição conjunta de  $(X, Y)$ , em que

$$H(x, y) = pC_1((F(x; \phi_1), G(y; \phi_2))) + (1 - p)C_2((F(x; \phi_1), G(y; \phi_2))). \quad (3.2)$$

Então a densidade dessa mistura é dada por:

$$h(x, y; \phi_1, \phi_2, p, \theta_1, \theta_2) = f(x; \phi_1)g(y; \phi_2)[pc_1(F(x; \phi_1), G(y; \phi_2); \theta_1) + (1 - p)c_2(F(x; \phi_1), G(y; \phi_2); \theta_2)]. \quad (3.3)$$

A inferência para os parâmetros  $(\phi_1, \phi_2, p, \theta_1, \theta_2)$  pode ser feita baseada na função de log-

verossimilhança

$$L(\phi_1, \phi_2, p, \theta_1, \theta_2) = \sum_{i=1}^n \log h(x_i, y_i; \phi_1, \phi_2, p, \theta_1, \theta_2), \quad (3.4)$$

definida para os valores de uma amostra aleatória  $(X_1, Y_1), \dots, (X_n, Y_n)$  de  $(X, Y)$ . Como a maximização de (3.3) pode ter um custo computacional alto, um método alternativo consiste em estimar separadamente os parâmetros das distribuições marginais,  $\phi_1$  e  $\phi_2$ . Maximiza-se os  $\phi_j$ 's via verossimilhança univariada substituindo as estimativas na equação (3.3), então a função é maximizada para  $\theta = (\theta_1, \theta_2)$ . Esse método é conhecido como Inferência por Marginais (IFM) criado por Joe (2005).

## 3.2 Estimação usando Algoritmo EM

Estimar os parâmetros da mistura de cópulas pode não ser simples, uma vez que, ao haver vários máximos locais, o algoritmo utilizado para obter as estimativas de máxima verossimilhança (EMV) poderá não estimar corretamente os parâmetros do modelo da mistura. Neste caso, calcular as EMV dos parâmetros da mistura via algoritmo EM substitui a difícil maximização da verossimilhança, por uma sequência de maximizações mais fáceis, em que o limite é o ótimo da função de máxima verossimilhança, Casella e Berger (2002).

Considere uma amostra observada de um vetor aleatório  $(X, Y)$  e, associado a este vetor, variáveis  $Z_{ij}$ 's,  $i = 1, \dots, n$ , assumindo o valor 1, se a  $i$ -ésima observação pertence  $j$ -ésima componente da mistura, e 0 caso contrário. Contudo, não se conhece os valores de  $Z_{ij}$ . O algoritmo EM é aplicado a este problema tratando  $Z_{ij}$  como dados faltantes. E procede iterativamente em dois passos, E (para expectância) e M (para maximização). Se conhecêssemos os valores de  $Z_{ij}$  a função de log verossimilhança para os dados completos seria dada por

$$\log L_c(\Psi) = \sum_{i=1}^n z_{ij} \log(h(x_i, y_i; \phi_1, \phi_2, p, \theta_1, \theta_2)),$$

em que  $\Psi = (p, \phi_1, \phi_2, \theta_1, \theta_2)$ .

Os dados não observados são lançados no problema usando a esperança condicional. Assim, com o intuito de encontrar uma distribuição para os dados completos  $(X, Y, Z)$ , estimamos a função de log verossimilhança conforme equação abaixo.

$$\log L(\Psi) = \sum_{i=1}^n w_{ji} \log(h(x_i, y_i; \phi_1, \phi_2, p, \theta_1, \theta_2)),$$

tal que  $w_{ji} = \mathbb{E}(Z_{ij} | (x, y)) = \mathbb{P}(Z_{ij} = 1 | (x, y))$ .

Considerando que os parâmetros das marginais já foram estimados num primeiro momento, usando o método IFM, o algoritmo EM consiste em iterar os dois seguintes passos até atingir um critério de convergência especificado, partindo de chutes iniciais para cada um dos parâmetros.

1. Expectância:

$$w_{1i} = \frac{pf(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_1(F(x_i; \hat{\phi}_1), G(y_i; \hat{\phi}_2); \theta_1)}{pf(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_1(F(x_i; \hat{\phi}_1), G(y_i; \hat{\phi}_2); \theta_1) + (1-p)f(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_2(F(x_i; \hat{\phi}_1), G(y_i; \hat{\phi}_2); \theta_2)}$$

e  $w_{2i} = 1 - w_{1i}$ .

2. Maximização: Atualização dos parâmetros usando

$$p = \sum_{i=1}^n \frac{w_{1i}}{n},$$

e os outros parâmetros podem ser atualizados via maximização da log-verossimilhança da forma

$$L_j = \sum_{i=1}^n w_{ij} [\log f(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_j(F(x_i; \hat{\phi}_1), G(y_i; \hat{\phi}_2); \theta_j)], j = 1, 2. \quad (3.4)$$

### 3.3 Matriz de Informação

A matriz de informação de Fisher esperada do vetor de parâmetros  $\Psi = (p_1, p_2, \phi_1, \phi_2, \theta_1, \theta_2)$  é definida por (McLachlan e Peel (2000))

$$\mathcal{I}(\Psi) = E_{\Psi}[U(\Psi)U^T(\Psi)], \quad (3.5)$$

em que

$$U(\Psi) = \frac{\partial \log L(\Psi)}{\partial \Psi}. \quad (3.6)$$

Na prática é comum usarmos a matriz de informação observada ao invés da esperada, definida por  $I(\hat{\Psi})$ , ou seja, sem tomar a esperança. A matriz de covariância assintótica do estimador de máxima verossimilhança  $\hat{\Psi}$  é aproximada pela inversa da matriz de informação observada  $I(\hat{\Psi})$ .

Uma forma alternativa para aproximar o erro padrão de  $\hat{\Psi}$  é através da técnica Bootstrap que foi introduzida por Efron (1979). Para mais conhecimento da técnica veja Efron (1982) e Efron e Tibshirani (1994). O procedimento a seguir descrito em McLachlan e Peel (2000) pode ser usado para aproximar o erro padrão de  $\hat{\Psi}$ .

- Passo 1. Gera-se B subamostras da amostra obtida.
- Passo 2. O algoritmo é EM é aplicado a cada amostra bootstrap para calcular o estimador de máxima verossimilhança para os dados da amostra bootstrap,  $\hat{\Psi}^*$ .
- Passo 3. A matriz de covariância bootstrap de  $\hat{\Psi}^*$  é dada por

$$cov^*(\hat{\Psi}^*) = \frac{\sum_{b=1}^B (\hat{\Psi}_b^* - \bar{\hat{\Psi}}^*)(\hat{\Psi}_b^* - \bar{\hat{\Psi}}^*)^T}{B - 1},$$

com

$$\bar{\hat{\Psi}}^* = \frac{\sum_{b=1}^B \hat{\Psi}_b^*}{B}$$

O erro padrão do  $i$ -ésimo elemento de  $\hat{\Psi}$  é estimado pela raiz quadrada do  $i$ -ésimo elemento da diagonal de  $cov^*(\hat{\Psi}^*)$ .

# Capítulo 4

## Análise Discriminante

A noção de análise de discriminante foi introduzida em 1920 por Karl Pearson, chamado de coeficiente de semelhança racial. O conceito de distância entre grupos de variáveis foi ampliado por Fisher na década de 1930. A análise de discriminante é uma técnica que trata da classificação de um elemento amostral a um grupo ou população conhecidos de acordo com algum conceito de distância. Neste trabalho, a classificação é baseada no princípio de máxima verossimilhança: atribui-se o elemento amostral ao grupo ou população  $\Pi_j$ ,  $j = 1, 2$  em que o vetor observado tenha máxima probabilidade de ocorrência, Huberty (1994).

### 4.1 Função discriminante para misturas de cópulas

Considere uma amostra de um vetor aleatório  $(X, Y)$  de uma mistura de duas populações  $\Pi_1$  e  $\Pi_2$  com funções de distribuição marginais  $F(x; \phi_1)$  e  $G(y; \phi_2)$  e funções de densidades marginais  $f(x; \phi_1)$  e  $g(y; \phi_2)$  e densidades conjuntas

$$f(x; \phi_1)g(y; \phi_2)c_j(F(x; \phi_1), G(y; \phi_2); \theta_j),$$

com  $j = 1, 2$  e  $c_j(u, v) = \frac{\partial^2 C_J(u, v)}{dudv}$ . Então a probabilidade de uma observação  $(x, y)$  de origem populacional desconhecida pertencer à população  $\Pi_1$  é expressa por

$$w_{1i} = \frac{pf(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_1(u, v)}{pf(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_1(u, v) + (1 - p)f(x_i; \hat{\phi}_1)g(y_i; \hat{\phi}_2)c_2(u, v)}, \quad (4.1)$$

em que  $u = F(x_i; \hat{\phi}_1)$  e  $v = G(y_i; \hat{\phi}_2)$ . Podemos simplificar a expressão acima de tal forma que

$$P((x, y) \in \pi_1) = \frac{1}{1 + \exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x; \hat{\phi}_1), G(y; \hat{\phi}_2); \theta_2)}{c_1(F(x; \hat{\phi}_1), G(y; \hat{\phi}_2); \theta_1)}])}. \quad (4.2)$$

A probabilidade de uma observação do vetor aleatório pertencer à população  $\Pi_2$  é o complementar da probabilidade de pertencer a  $\Pi_1$ , ou seja,

$$\begin{aligned}
P((x, y) \in \Pi_2) &= 1 - \frac{1}{1 + \exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])]} \\
&= \frac{\exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])}{1 + \exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])}.
\end{aligned}$$

Assim, a classificação é feita atribuindo a observação à população  $\Pi_1$  se

$$P((x, y) \in \Pi_1) \geq P((x, y) \in \Pi_2),$$

isto é, se

$$\frac{1}{1 + \exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])} \geq \frac{\exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])}{1 + \exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}])} \quad (4.3)$$

que corresponde a

$$\exp(\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}]) \leq 1,$$

ou

$$\log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}] \leq 0.$$

Caso contrário o vetor observado é classificado à população  $\Pi_2$ . Logo, a função discriminante para mistura de cópulas (2.1) é definida por

$$NL(x, y) = \log[\frac{1-p}{p}] + \log[\frac{c_2(F(x;\phi_1), G(y;\phi_2); \theta_2)}{c_1(F(x;\phi_1), G(y;\phi_2); \theta_1)}]. \quad (4.4)$$

Portanto, uma dada observação  $(x, y)$  do vetor  $(X, Y)$  com densidade (3.3) será classificada em  $\Pi_1$  se  $NL(x, y) \leq 0$  e será classificada em  $\Pi_2$  se  $NL(x, y) > 0$ .

Dependendo das famílias de cópulas  $C_1(F(x;\phi_1), G(y;\phi_2); \theta_1)$  e  $C_2(F(x;\phi_1), G(y;\phi_2); \theta_2)$  a função  $NL(x, y)$  terá diferentes formas. Na Tabela 4.1. apresentamos as expressões de  $NL(x, y)$  para diferentes misturas de cópulas  $C_1(F(x;\phi_1), G(y;\phi_2); \theta_1)$  e  $C_2(F(x;\phi_1), G(y;\phi_2); \theta_2)$ .



Tabela 4.1: Mistura de cópulas e suas funções discriminantes

Mistura	$C_1(F(x; \phi_1), G(y; \phi_2); \theta_1)$	$C_2(F(x; \phi_1), G(y; \phi_2); \theta_2)$	$NL(x, y)$
i	Clayton	Clayton	$\alpha + \beta z + z_3$
ii	Gumbel	Gumbel	$\alpha + \beta z + z_3$
iii	Frank	Frank	$\alpha + \beta z + z_3$
iv	Joe	Joe	$\alpha + \beta z + z_3$
v	Gumbel	Clayton	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$
vi	Gumbel	Frank	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$
vii	Joe	Gumbel	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$
viii	Clayton	Frank	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$
ix	Clayton	Joe	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$
x	Frank	Joe	$\alpha + \beta_1 z_1 + \beta_2 z_2 + z_3$

Com  $\alpha = \alpha(p, \theta_1, \theta_2)$ ,  $\beta = \beta(\theta_1, \theta_2)$ ,  $z = h(x, y, \theta_1, \theta_2)$ ,  $\beta_1 = \beta_1(\theta_1)$ ,  $\beta_2 = \beta_2(\theta_2)$ ,  $z_1 = h_1(x, y, \theta_1)$ ,  $z_2 = h_2(x, y, \theta_2)$  e  $z_3 = h_3(x, y, \theta_1, \theta_2)$ .

**Caso (i).** Suponha que as cópulas  $C_1(\cdot, \cdot; \theta_1)$  e  $C_2(\cdot, \cdot; \theta_2)$  sejam ambas Clayton, então temos

$$\alpha = \log\left(\frac{p_2}{p_1}\right) + \log\left(\frac{\theta_2 + 1}{\theta_1 + 1}\right),$$

$$\beta = \theta_1 - \theta_2,$$

$$z = \log(F(x; \phi_1)G(y; \phi_2))$$

e

$$z_3 = \frac{1 + 2\theta_1}{\theta_1} \log(F(x; \phi_1)^{-\theta_1} + G(y; \phi_2)^{-\theta_1} - 1) - \frac{1 + 2\theta_2}{\theta_2} \log(F(x; \phi_1)^{-\theta_2} + G(y; \phi_2)^{-\theta_2} - 1).$$

**Caso (v).** Considerando  $C_1(\cdot, \cdot; \theta_1)$  cópula de Gumbel e  $C_2(\cdot, \cdot; \theta_2)$  cópula de Clayton, segue que

$$\alpha = \log\left(\frac{p_2}{p_1}\right) + \log(\theta_2 + 1),$$

$$\beta_1 = 1 - \theta_1,$$

$$z_1 = \log[\log(F(x; \phi_1)) \log(G(y; \phi_2))],$$

$$\beta_2 = -\theta_2,$$

$$z_2 = \log[F(x; \phi_1)G(y; \phi_2)]$$

e

$$\begin{aligned}
z_3 = & [(-\log(F(x; \phi_1)))^{\theta_1} + (-\log(G(y; \phi_2)))^{\theta_1}]^{\frac{1}{\theta_1}} - \\
& - \log\{(\theta_1 - 1)[(-\log(F(x; \phi_1)))^{\theta_1} + (-\log(G(y; \phi_2)))^{\theta_1}]^{\frac{1}{\theta_1}-2} + \\
& + [(-\log(F(x; \phi_1)))^{\theta_1} + (-\log(G(y; \phi_2)))^{\theta_1}]^{\frac{2}{\theta_1}-2}\} - \\
& - \frac{1 + 2\theta_2}{\theta_2} \log(F(x; \phi_1)^{-\theta_2} + G(y; \phi_2)^{-\theta_2} - 1).
\end{aligned}$$

As funções discriminantes são encontradas a partir da densidade das cópulas e das respectivas marginais F e G. A seguir são listadas a densidade de cada uma das cópulas tratadas neste trabalho. Fazendo  $u = F(x)$  e  $v = G(y)$ :

Clayton, considere a distribuição de cópula

$$C(u, v; \theta) = (u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}}$$

e então,

$$\frac{\partial^2 C(u, v; \theta)}{dudv} = u^{-(\theta+1)} v^{-(\theta+1)} (\theta + 1) (u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}}. \quad (4.5)$$

Gumbel, considere a distribuição de cópula

$$C(u, v; \theta) = e^{-[(-\log(u))^\theta + (-\log(v))^\theta]^{\frac{1}{\theta}}},$$

logo,

$$\begin{aligned}
\frac{\partial^2 C(u, v; \theta)}{dudv} = & \frac{1}{u} \frac{1}{v} (\log(u) \log(v))^{\theta-1} e^{-[(-\log(u))^\theta + (-\log(v))^\theta]^{\frac{1}{\theta}}} \\
& [(\theta - 1)[(-\log(u))^\theta + (-\log(v))^\theta]^{\frac{1}{\theta}-2} + [(-\log(u))^\theta + (-\log(v))^\theta]^{\frac{2}{\theta}-2}]. \quad (4.6)
\end{aligned}$$

Frank, considere a distribuição de cópula

$$C(u, v; \theta) = -\frac{1}{\theta} \log \left( 1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right),$$

assim,

$$\frac{\partial^2 C(u, v; \theta)}{dudv} = \theta e^{-\theta u} e^{-\theta v} \frac{1 - e^{-\theta}}{[e^{-\theta} - 1 + (e^{-\theta u} - 1)(e^{-\theta v} - 1)]^2}. \quad (4.7)$$

Joe, considere a distribuição de cópula

$$C(u, v; \theta) = 1 - [(1 - u)^\theta + (1 - v)^\theta - (1 - u)^\theta (1 - v)^\theta]^{\frac{1}{\theta}},$$

portanto,

$$\begin{aligned} \frac{\partial^2 C(u, v; \theta)}{dudv} &= (1-u)^{\theta-1}(1-v)^{\theta-1}[(1-\theta)[(1-u)^\theta + \\ &(1-v)^\theta - (1-u)^\theta(1-v)^\theta]^{\frac{1}{\theta}-2} [1 - (1-u)^\theta - (1-v)^\theta + (1-u)^\theta(1-v)^\theta] + \\ &[(1-u)^\theta + (1-v)^\theta - (1-u)^\theta(1-v)^\theta]^{\frac{1}{\theta}-1}. \end{aligned} \quad (4.8)$$

Pela equação (4.4), temos que a função discriminante é dada por

$$NL(x, y) = \log\left[\frac{1-p}{p}\right] + \log\left[\frac{c_2(F(x; \phi_1), G(y; \phi_2); \theta_2)}{c_1(F(x; \phi_1), G(y; \phi_2); \theta_1)}\right]$$

então, basta substituir a densidade das cópulas em cada um dos casos que obtemos o formato de  $NL(x, y)$  na Tabela 4.1. Vejamos o caso (iv):

Pela propriedade do logaritmo, tem-se que a segunda parte do lado direito da equação (4.4) é igual a

$$\log f(x)g(y)c_2(F(x; \phi_1), G(y; \phi_2); \theta_2) - \log f(x)g(y)c_1(F(x; \phi_1), G(y; \phi_2); \theta_1),$$

substituindo a densidade da cópula de Joe, resulta

$$NL(x, y) = \log\left(\frac{1-p}{p}\right) + (\theta_2 - \theta_1) * \log((1-u)(1-v)) + \log(J(x, y; \phi_1, \phi_2, \theta_1, \theta_2)), \quad (4.9)$$

em que

$$J(x, y; \phi_1, \phi_2, \theta_1, \theta_2) = \frac{J_2(x, y; \phi_1, \phi_2, \theta_1, \theta_2)}{J_1(x, y; \phi_1, \phi_2, \theta_1, \theta_2)}. \quad (4.10)$$

$$\begin{aligned} J_2(x, y; \phi_1, \phi_2, \theta_1, \theta_2) &= f(x)g(y)[(1-\theta_2)[(1-u)^{\theta_2} + \\ &(1-v)^{\theta_2} - (1-u)^{\theta_2}(1-v)^{\theta_2}]^{\frac{1}{\theta_2}-2} \\ &[[1 - (1-u)^{\theta_2} - (1-v)^{\theta_2} + (1-u)^{\theta_2}(1-v)^{\theta_2}] + \\ &[(1-u)^{\theta_2} + (1-v)^{\theta_2} - (1-u)^{\theta_2}(1-v)^{\theta_2}]^{\frac{1}{\theta_2}-1}, \end{aligned}$$

e

$$\begin{aligned}
J_1(x, y; \phi_1, \phi_2, \theta_1, \theta_2) &= f(x)g(y)[(1 - \theta_1)[(1 - u)^{\theta_1} + \\
&\quad (1 - v)^{\theta_1} - (1 - u)^{\theta_1}(1 - v)^{\theta_1}]^{\frac{1}{\theta_1} - 2} \\
&][1 - (1 - u)^{\theta_1} - (1 - v)^{\theta_1} + (1 - u)^{\theta_1}(1 - v)^{\theta_1}] + \\
&[(1 - u)^{\theta_1} + (1 - v)^{\theta_1} - (1 - u)^{\theta_1}(1 - v)^{\theta_1}]^{\frac{1}{\theta_1} - 1}
\end{aligned}$$

Ou seja, tem-se

$$\alpha = \log\left(\frac{1 - p}{p}\right),$$

$$\beta = \theta_2 - \theta_1,$$

$$z = \log((1 - F(x; \phi_1))(1 - G(y; \phi_2)))$$

e

$$z_3 = \log(J(x, y; \phi_1, \phi_2, \theta_2)).$$

No caso de as marginais das duas componentes serem diferentes, então a parte  $\beta z$  se modificará para

$$\beta z = \beta_1 z_1 + \beta_2 z_2, \quad (4.11)$$

tal que

$$\beta_1 z_1 + \beta_2 z_2 = (\theta_2 - 1) \log((1 - F_2(x; \phi_1))(1 - G_2(y; \phi_2))) - (\theta_1 - 1) \log((1 - F_1(x; \phi_1))(1 - G_1(y; \phi_2))).$$

## 4.2 Erros de classificação

Ao usarmos o procedimento de classificação, estamos sujeitos a incorrer em dois tipos de erros. Um erro ocorrerá quando não classificarmos uma observação  $(x, y)$  à população  $\Pi_1$ , se a função discriminante  $NL(x, y) > 0$ , sendo que  $(x, y)$  pertence a  $\Pi_1$ . Ou seja, a probabilidade de atribuímos de forma errada uma observação  $(x, y)$  à população  $\Pi_2$  usando  $NL(x, y)$  é

$$e_1 = P[NL(x, y) > 0 \cap (x, y) \in \pi_1], \quad (4.12)$$

$$e_1 = P((x, y) \in \pi_1)P[NL(x, y) > 0 | (x, y) \in \pi_1] = p_1 P[NL(x, y) > 0 | (x, y) \in \pi_1]$$

De forma análoga, a probabilidade de atribuímos erroneamente uma observação  $(x, y)$  à popu-

lação  $\pi_1$  usando  $NL(x, y)$  é dada por

$$e_2 = p_2 P[NL(x, y) \leq 0 | (x, y) \in \pi_2]. \quad (4.13)$$

Assim a probabilidade total de erro de classificação é

$$e = e_1 + e_2. \quad (4.14)$$

# Capítulo 5

## Resultados

Este capítulo traz os resultados de simulações de diversas misturas de cópulas arquimedianas e uma aplicação em dados reais referente a erupções do Geysir Old Faithful em Yellowstone National Park, Wyoming, EUA. A base de dados possui informações do tempo de duração de erupção do geysir e o tempo até a próxima erupção do geysir. As simulações foram feitas usando-se o software R Core Team (2018).

### 5.1 Simulação

Foram realizadas simulações de misturas das cópulas arquimedianas listadas na Tabela 1.1, com o intuito de analisar a capacidade das misturas de captarem os diversos formatos de dependência nos dados e o desempenho de função discriminante baseada na mistura de cópulas em classificar corretamente as observações. A simulação foi realizada para 24 misturas, por meio da combinação do valor de  $p$  e dos parâmetros das cópulas, conforme Tabela 5.1, com amostras de tamanho  $n = 1000$ . Para cada simulação foi realizada a estimação dos erros-padrões dos estimadores via procedimento Bootstrap de tamanho igual a 100, de acordo com a seção 3.3. Os resultados das simulações são apresentados nas Tabelas 5.2 e 5.3.

Para gerar amostras de tamanho  $N$  de mistura de cópulas, cuja distribuição é dada em (2.1), utiliza-se o seguinte algoritmo:

#### Algoritmo 5.1.

- 1. Gera-se uma amostra  $u_0$  de uma distribuição uniforme no intervalo  $[0, 1]$ ;
- 2. Se  $u_0 \leq p$ , então gera-se uma observação da primeira cópula, ou seja, de  $C_1$  usando o algoritmo 1.2, por ser mais fácil de usar e mais rápido para simular. Se  $u_0 > p$ , então gera-se uma observação da segunda cópula, ou seja, de  $C_2$  usando o algoritmo 1.2;
- 3. Gera-se as observações  $x = F^{-1}(u)$  e  $y = G^{-1}(v)$ , tal que  $X \sim F$  e  $Y \sim G$ ;
- 4. Repete-se os passos 1, 2 e 3,  $N$  vezes.

Tabela 5.1: Parâmetros usados nas simulações.

Mistura	$p$	Cópula $C_1$	$\theta_1$	Cópula $C_2$	$\theta_2$
$m_1$	0.3	Frank	-5.0	Gumbel	1.0
$m_2$	0.5	Frank	-5.0	Gumbel	1.0
$m_3$	0.7	Frank	-5.0	Gumbel	1.0
$m_4$	0.3	Frank	-5.0	Gumbel	10
$m_5$	0.5	Frank	-5.0	Gumbel	10
$m_6$	0.7	Frank	-5.0	Gumbel	10
$m_7$	0.1	Clayton	-0.8	Joe	4.0
$m_8$	0.6	Clayton	-0.8	Joe	4.0
$m_9$	0.9	Clayton	-0.8	Joe	4.0
$m_{10}$	0.1	Clayton	3.0	Joe	2.0
$m_{11}$	0.6	Clayton	3.0	Joe	2.0
$m_{12}$	0.9	Clayton	3.0	Joe	2.0
$m_{13}$	0.25	Frank	-20.0	Joe	15.0
$m_{14}$	0.5	Frank	-20.0	Joe	15.0
$m_{15}$	0.75	Frank	-20.0	Joe	15.0
$m_{16}$	0.25	Gumbel	3	Clayton	15.0
$m_{17}$	0.5	Gumbel	3	Clayton	15.0
$m_{18}$	0.75	Gumbel	3	Clayton	15.0
$m_{19}$	0.25	Frank	-5	Frank	5.0
$m_{20}$	0.4	Frank	-5	Frank	5.0
$m_{21}$	0.8	Frank	-5	Frank	5.0
$m_{22}$	0.25	Frank	-15	Frank	-5.0
$m_{23}$	0.5	Frank	-15	Frank	-5.0
$m_{24}$	0.75	Frank	-15	Frank	-5.0

As distribuições marginais simuladas foram  $X \sim Gumbel(\mu_1 = 4, \sigma_1 = 1)$  e  $Y \sim Gumbel(\mu_2 = 0, \sigma_2 = 1)$ , sendo  $\mu_1$  e  $\mu_2$  parâmetros de locação e  $\sigma_1$  e  $\sigma_2$  parâmetros de escala. Os parâmetros das distribuições marginais foram estimados usando-se o método de máxima verossimilhança. A estimação dos parâmetros das misturas foi feita por meio do algoritmo EM, sendo que o algoritmo usado para maximização da log-verossimilhança foi o PSO (*Particle Swarm Optimization*), mais conhecido como "Exame de Partículas" ou "Algoritmo dos Pássaros". O critério de convergência utilizado foi  $|L[k] - L[k - 1]| < 0.0001$ , em que  $L[k]$  é o valor da função de verossimilhança na iteração  $k$ .

As Tabelas 5.2 e 5.3 apresentam os valores das estimativas dos parâmetros simulados, sendo  $\hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2$  e  $\hat{\sigma}_2$  as estimativas para os parâmetros das marginais, considerando como funções de distribuições marginais a distribuição Gumbel, em que  $\mu_1$  e  $\mu_2$  parâmetros de locação e  $\sigma_1$  e  $\sigma_2$  parâmetros de escala, apresentada nas equações 5.1 e 5.2. Na Tabela 5.3,  $\hat{p}$  é a estimativa para o percentual da mistura,  $\hat{\theta}_1$  e  $\hat{\theta}_2$  as estimativas para os parâmetros das cópulas  $C_1$  e  $C_2$ , respectivamente. Já  $\hat{\tau}$  é o coeficiente *tau* de Kendall estimado pelos parâmetros da mistura, calculado tomando-se como base a equação (2.2),  $\hat{\tau}_{am}$  é o valor amostral do *tau* de Kendall,  $\hat{\rho}$  é a estimativa do coeficiente de correlação de Spearman dada pelos parâmetros da

mistura estimada, utilizando-se da equação (2.3),  $\hat{\rho}_{am}$  é o seu correspondente amostral,  $\hat{\lambda}_L$  e  $\hat{\lambda}_U$  são as estimativas dos coeficientes de dependência caudal inferior e superior, encontrados de acordo com as equações (2.4) e (2.5), respectivamente.

Observou-se que, em geral, as estimativas encontradas estão próximas dos parâmetros simulados e dos correspondentes amostrais, o que indica que os valores foram estimados adequadamente. Foi verificada também a qualidade dos estimadores de máxima verossimilhança graficamente, fez-se a plotagem das curvas de níveis das misturas nos parâmetros simulados e nos estimados. As Figuras 5.1 a 5.5 mostram os gráficos das curvas de níveis teórica e estimada, plotadas contra os dados, constatando-se que os ajustes dos parâmetros estão adequados, visto que as curvas são muito semelhantes.

Tabela 5.2: Estimadores dos parâmetros das marginas das misturas simuladas mostradas na Tabela 5.1, com os parâmetros originais  $\mu_1 = 4$ ,  $\mu_2 = 0$ ,  $\sigma_1 = 1$  e  $\sigma_2 = 1$ , considerando distribuição Gumbel.

Mistura	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$
m1	3.9971	0.9709	0.0630	1.0009
m2	3.9947	1.0069	0.0133	0.9690
m3	3.9733	0.9717	0.0578	1.0216
m4	3.9983	0.9762	-0.0447	0.9817
m5	3.9800	0.9900	0.0100	1.0100
m6	4.0400	1.0000	0.0000	0.9800
m7	3.9300	0.9500	-0.0700	0.9600
m8	.9845	0.9769	0.0177	0.9802
m9	4.0112	0.9654	0.0164	0.9897
m10	3.9963	1.0021	0.0012	1.0093
m11	4.0758	1.0401	0.0725	1.0615
m12	3.9635	0.9694	-0.0535	0.9809
m13	3.9544	0.9993	-0.0388	1.0065
m14	4.0326	1.0167	-0.0333	1.0006
m15	3.9599	0.9860	0.0117	1.0401
m16	3.9609	1.0215	-0.0505	1.0162
m17	3.9623	0.9889	-0.0165	0.9916
m18	3.9914	1.0044	0.0046	0.9947
m19	4.0765	0.9744	0.0589	1.0060
m20	4.0311	0.9275	0.0000	0.9917
m21	3.9952	1.0159	-0.0129	0.9957
m22	3.9145	0.9770	0.0412	1.0390
m23	3.9844	1.0181	0.0402	0.9912
m24	3.9536	0.9741	0.0484	1.0085

A Tabela 5.4 contém os erros-padrões dos estimadores dos parâmetros calculados através do método Bootstrap, descrito na seção 3.3, com  $B = 100$  amostras de tamanho 1000.

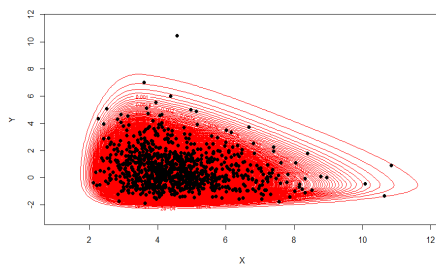


Tabela 5.3: Estimadores dos parâmetros das misturas, coeficientes de correlação de Kendal estimado usando-se a equação (2.2) e amostral, coeficientes de correlação de Spearman estimado conforme equação (2.3) e amostral, e coeficientes de dependência caudal inferior e superior.

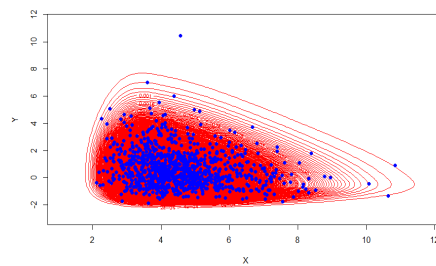
Mistura	$\hat{\rho}$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\tau}$	$\hat{\tau}_{am}$	$\hat{\rho}$	$\hat{\rho}_{am}$	$\lambda_L$	$\lambda_U$
$m_1$	0.2970	-5.238	1.0000	-0.1245	-0.1324	-0.1963	-0.1953	0.0000	0.0000
$m_2$	0.4961	-5.7921	1.0141	-0.2332	-0.2269	-0.3359	-0.3257	0.0000	0.0097
$m_3$	0.6875	-4.4211	1.0000	-0.2832	-0.2881	-0.4098	-0.4172	0.0000	0.0000
$m_4$	0.3091	-4.7983	10.0558	0.4298	0.4381	0.4869	0.4993	0.0000	0.6416
$m_5$	0.5093	5.4416	9.4046	0.1313	0.1452	0.1387	0.1572	0.0000	0.4531
$m_6$	0.7275	-4.9840	9.0973	-0.1379	-0.1438	-0.1994	-0.2096	0.0000	0.251
$m_7$	0.0919	-0.7039	3.6249	0.4807	0.456	0.6342	0.5954	0.246	0.7167
$m_8$	0.5872	-0.6013	3.9074	-0.0111	-0.134	-0.0119	-0.1488	1.8594	0.3327
$m_9$	0.8764	-0.7618	2.9806	-0.4678	-0.5172	-0.5723	-0.6113	2.1768	0.0913
$m_{10}$	0.2181	2.3622	2.0986	0.4093	0.4102	0.574	0.5784	0.1626	0.4759
$m_{11}$	0.6094	3.1458	2.0097	0.5045	0.5109	0.6839	0.6894	0.4889	0.2297
$m_{12}$	0.9066	3.3849	1.4819	0.5853	0.5832	0.7664	0.7633	0.7388	0.0377
$m_{13}$	0.2443	-21.2923	13.6860	0.4462	0.4509	0.4998	0.5095	0.0000	0.7164
$m_{14}$	0.4986	-19.1737	13.9451	0.0194	0.0217	0.0122	0.0119	0.0000	0.4759
$m_{15}$	0.7524	-20.8170	10.2393	-0.4156	-0.4073	-0.4866	-0.4694	0.0000	0.2303
$m_{16}$	0.2350	3.3625	14.9914	0.8318	0.8303	0.955	0.9536	0.1912	0.7288
$m_{17}$	0.4984	3.1296	15.2339	0.7694	0.7663	0.9200	0.9123	0.479	0.3749
$m_{18}$	0.7350	2.9018	13.5129	0.702	0.6964	0.875	0.8657	0.2518	0.5367
$m_{19}$	0.2885	-3.7956	5.3787	0.2291	0.2352	0.3225	0.3325	0.0000	0.0000
$m_{20}$	0.4461	-4.4667	6.3832	0.0999	0.0899	0.138	0.1215	0.0000	0.0000
$m_{21}$	0.7092	-5.8660	2.9918	-0.2624	-0.2647	-0.3682	-0.3716	0.0000	0.0000
$m_{22}$	0.2936	-15.7680	-5.1883	-0.5478	-0.5525	-0.7391	-0.7447	0.0000	0.0000
$m_{23}$	0.5514	-15.9834	-4.2467	-0.5941	-0.5967	-0.777	-0.778	0.0000	0.0000
$m_{24}$	0.6970	-15.5595	-6.2271	-0.689	-0.6904	-0.8702	-0.8712	0.0000	0.0000

Tabela 5.4: Erros-padrões dos estimadores dos parâmetros das misturas via Bootstrap.

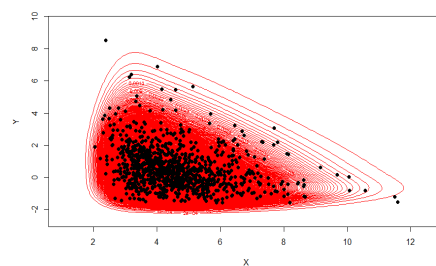
Mistura	EP( $\hat{\mu}_1$ )	EP( $\hat{\mu}_2$ )	EP( $\hat{\sigma}_1$ )	EP( $\hat{\sigma}_2$ )	EP( $\hat{\rho}$ )	EP( $\hat{\theta}_1$ )	EP( $\hat{\theta}_2$ )
m1	0.0273	0.0210	0.0307	0.0274	0.0198	0.6737	0.0040
m2	0.0370	0.0187	0.0352	0.0261	0.0175	0.5358	0.0359
m3	0.0351	0.0225	0.0351	0.0231	0.0137	0.3516	0.0082
m4	0.0324	0.0244	0.0234	0.0325	0.0227	0.0213	0.0171
m5	0.0394	0.0172	0.0381	0.0243	0.0189	0.3739	1.1361
m6	0.0313	0.0245	0.0306	0.0238	0.0160	0.2871	1.3738
m7	0.0286	0.0203	0.0321	0.0229	0.0193	0.0646	0.2162
m8	0.0313	0.0267	0.0317	0.0233	0.0235	0.0039	0.3038
m9	0.0353	0.0233	0.0376	0.0216	0.0304	0.0099	0.8191
m10	0.0349	0.0253	0.0338	0.0248	0.0315	0.5639	0.1087
m11	0.0348	0.0247	0.0371	0.0245	0.0420	0.2815	0.1295
m12	0.0321	0.0208	0.0334	0.0235	0.0224	0.2158	0.3392
m13	0.0357	0.0265	0.0345	0.0285	0.0158	1.8370	1.0025
m14	0.0297	0.0248	0.0317	0.0242	0.0163	0.8619	2.1752
m15	0.0303	0.0214	0.0366	0.0228	0.0154	0.7308	2.1020
m16	0.0332	0.0240	0.0331	0.0231	0.0293	0.2914	0.8389
m17	0.0338	0.0271	0.0357	0.0237	0.0262	0.1679	1.3145
m18	0.0385	0.0233	0.0370	0.0217	0.0325	0.1095	3.1701
m19	0.0325	0.0252	0.0280	0.0250	0.0227	0.8470	0.3792
m20	0.0318	0.0231	0.0318	0.0254	0.0583	1.0670	1.0913
m21	0.0330	0.0232	0.0326	0.0241	0.0231	0.4658	0.6291
m22	0.0294	0.0245	0.0314	0.0225	0.0518	2.4622	0.4430
m23	0.0325	0.0302	0.0322	0.0227	0.0422	1.1307	0.5720
m24	0.0316	0.0239	0.0322	0.0251	0.0131	0.8161	0.5705



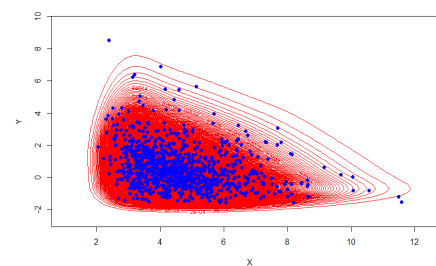
(a) Curvas de níveis de m1 teórica



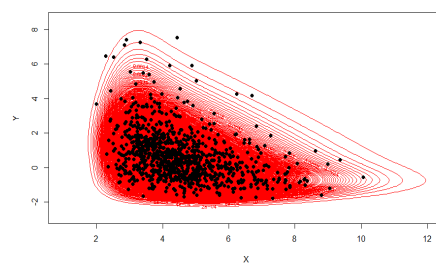
(b) Curvas de níveis de m1 estimada



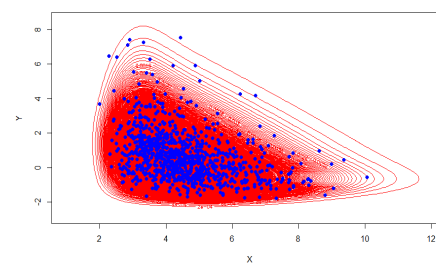
(c) Curvas de níveis de m2 teórica



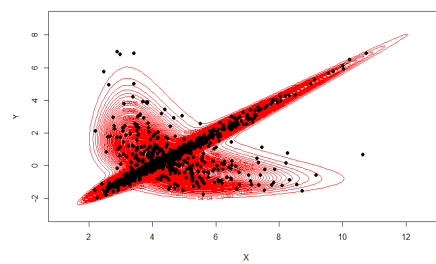
(d) Curvas de níveis de m2 estimada



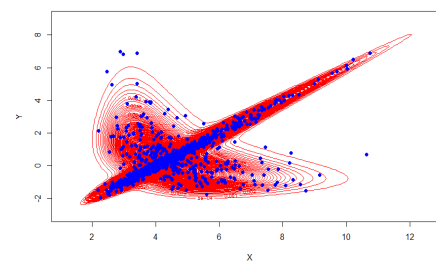
(e) Curvas de níveis de m3 teórica



(f) Curvas de níveis de m3 estimada

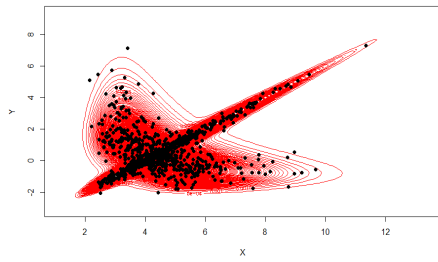


(g) Curvas de níveis de m4 teórica

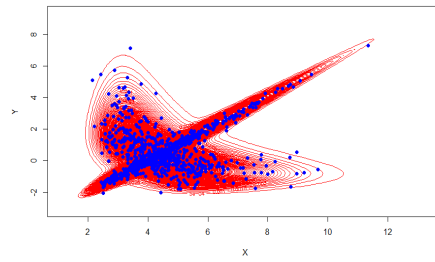


(h) Curvas de níveis de m4 estimada

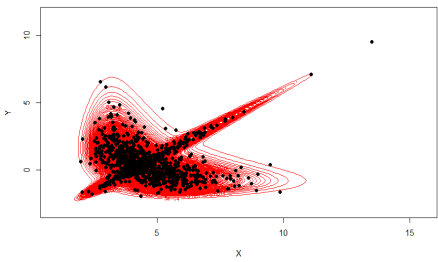
Figura 5.1: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m1 a m4 da Tabela 5.1.



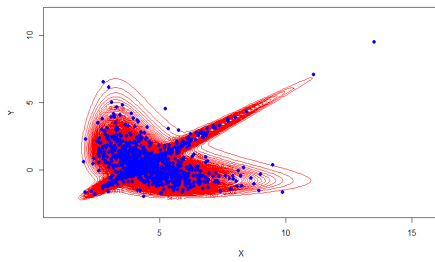
(a) Curvas de níveis de m5 teórica



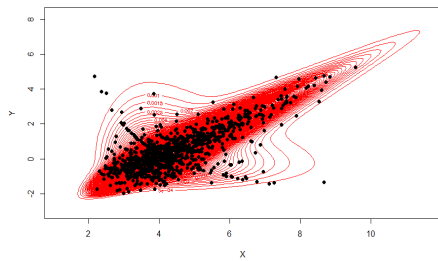
(b) Curvas de níveis de m5 estimada



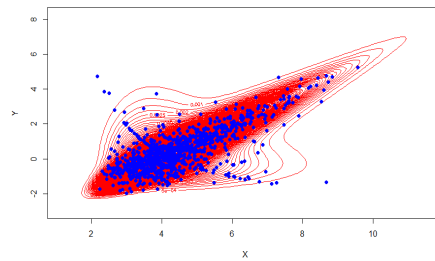
(c) Curvas de níveis de m6 teórica



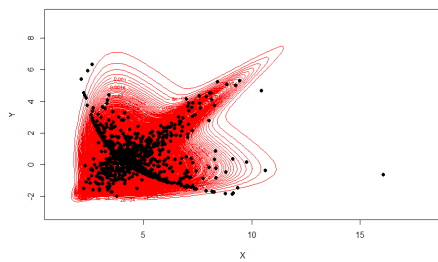
(d) Curvas de níveis de m6 estimada



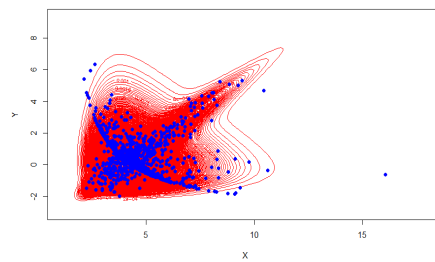
(e) Curvas de níveis de m7 teórica



(f) Curvas de níveis de m7 estimada

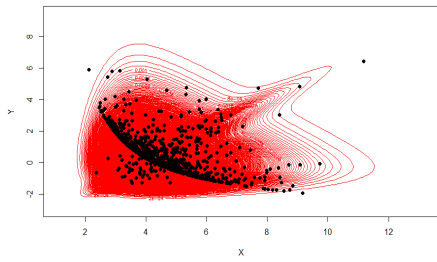


(g) Curvas de níveis de m8 teórica

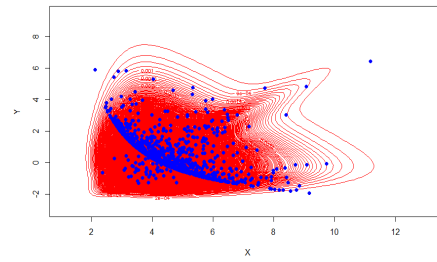


(h) Curvas de níveis de m8 estimada

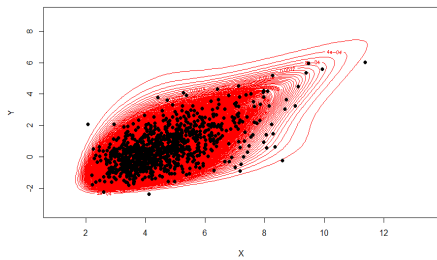
Figura 5.2: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m5 a m8 da Tabela 5.1.



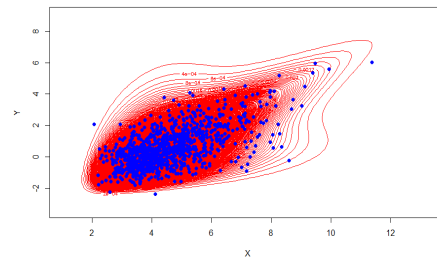
(a) Curvas de níveis de m9 teórica



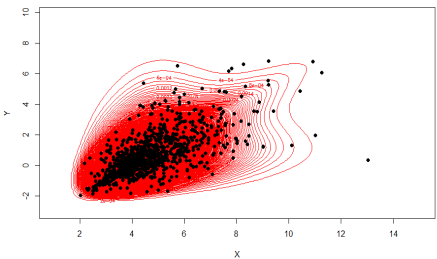
(b) Curvas de níveis de m9 estimada



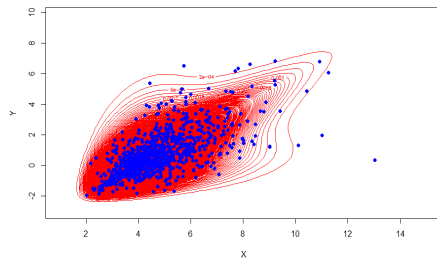
(c) Curvas de níveis de m10 teórica



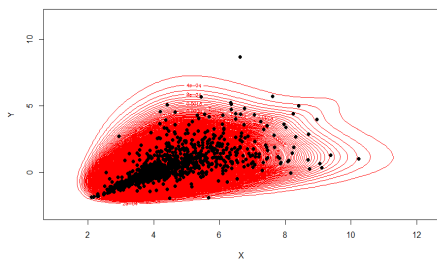
(d) Curvas de níveis de m10 estimada



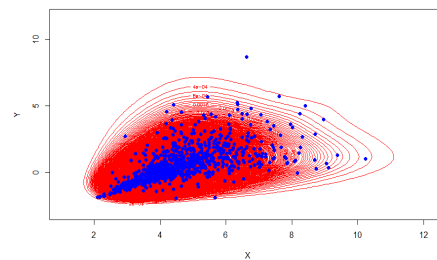
(e) Curvas de níveis de m11 teórica



(f) Curvas de níveis de m11 estimada

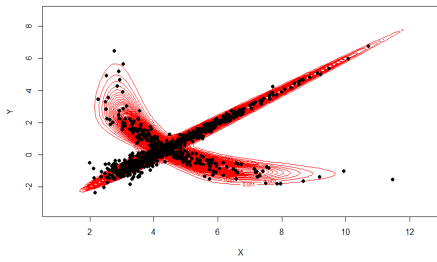


(g) Curvas de níveis de m12 teórica

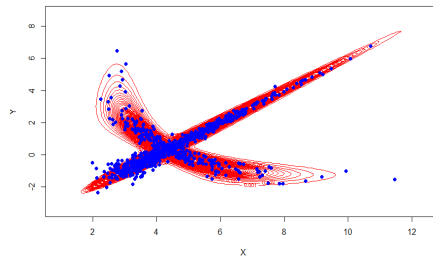


(h) Curvas de níveis de m12 estimada

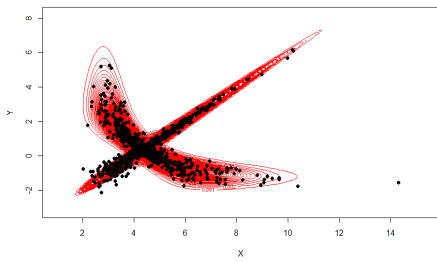
Figura 5.3: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m9 a m12 da Tabela 5.1.



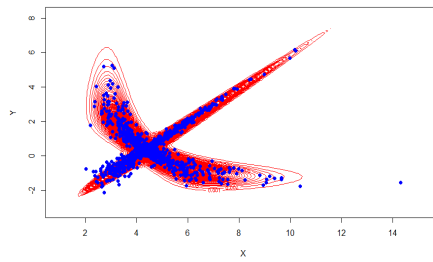
(a) Curvas de níveis de m13 teórica



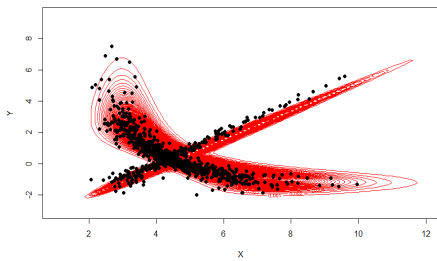
(b) Curvas de níveis de m13 estimada



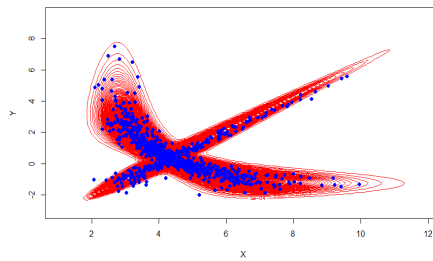
(c) Curvas de níveis de m14 teórica



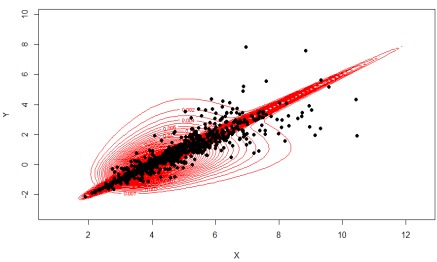
(d) Curvas de níveis de m14 estimada



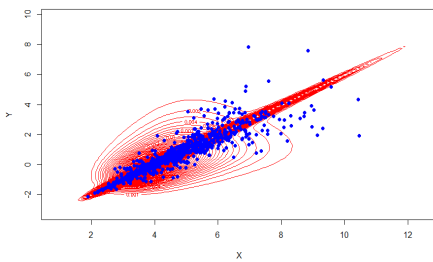
(e) Curvas de níveis de m15 teórica



(f) Curvas de níveis de m15 estimada

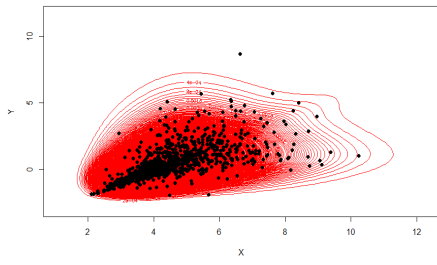


(g) Curvas de níveis de m16 teórica

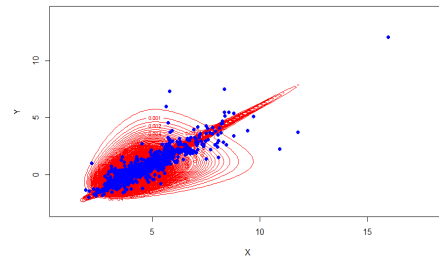


(h) Curvas de níveis de m16 estimada

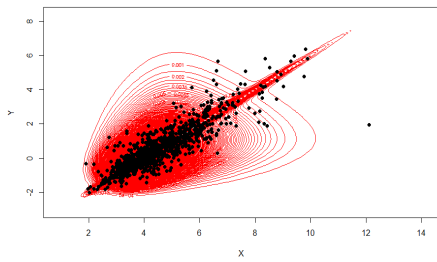
Figura 5.4: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m13 a m16 da Tabela 5.1.



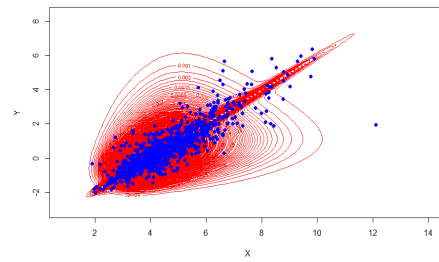
(a) Curvas de níveis de m17 teórica



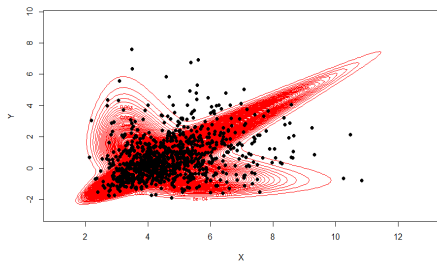
(b) Curvas de níveis de m17 estimada



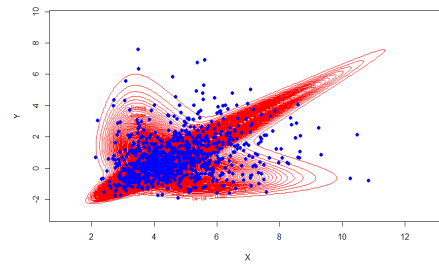
(c) Curvas de níveis de m18 teórica



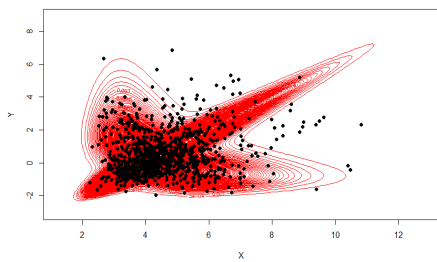
(d) Curvas de níveis de m18 estimada



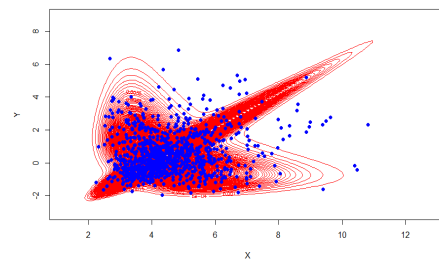
(e) Curvas de níveis de m19 teórica



(f) Curvas de níveis de m19 estimada

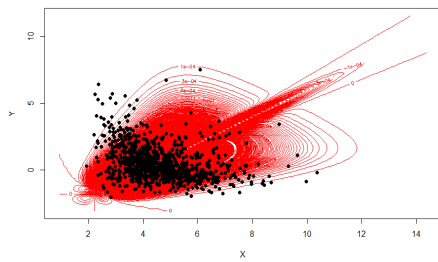


(g) Curvas de níveis de m20 teórica

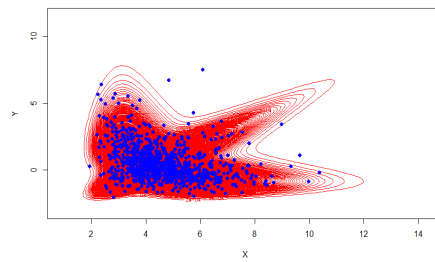


(h) Curvas de níveis de m20 estimada

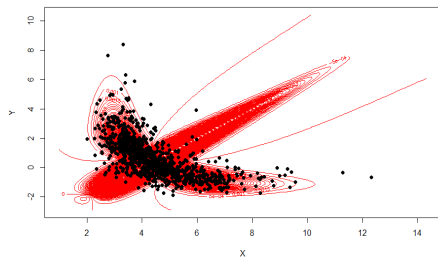
Figura 5.5: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m17 a m20 da Tabela 5.1.



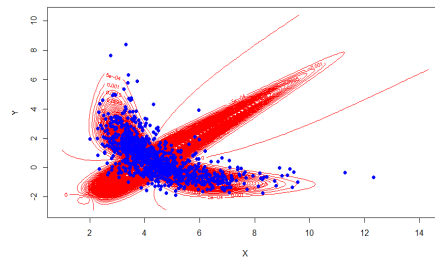
(a) Curvas de níveis de m21 teórica



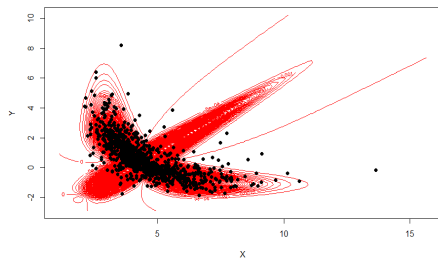
(b) Curvas de níveis de m21 estimada



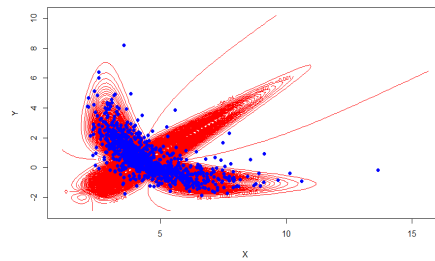
(c) Curvas de níveis de m22 teórica



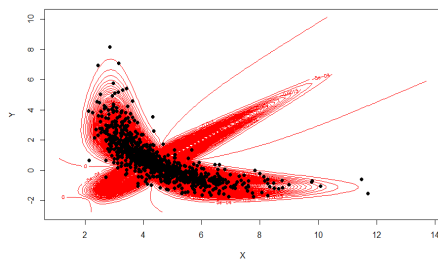
(d) Curvas de níveis de m22 estimada



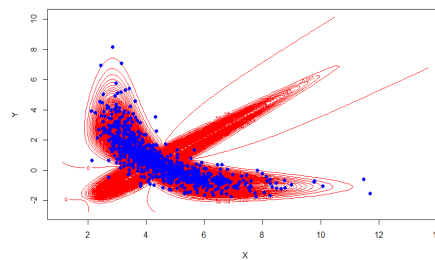
(e) Curvas de níveis de m23 teórica



(f) Curvas de níveis de m23 estimada



(g) Curvas de níveis de m24 teórica



(h) Curvas de níveis de m24 estimada

Figura 5.6: Curvas de níveis para mistura de Cópulas, teórica e estimada, misturas m21 a m24 da Tabela 5.1.

Para verificar a qualidade da função discriminante, foram calculados os erros de classificação para cada mistura da Tabela 5.1. Os resultados são mostrados na Tabela 5.5, juntamente com o viés para o erro teórico (ótimo) da função discriminante. O viés foi calculado via Bootstrap com  $B=100$  amostras. Em geral, as funções discriminantes para as misturas tiveram

desempenho adequado, apresentando baixos erros de classificação.

Tabela 5.5: Erros do discriminante e viés em relação ao discriminante ótimo

Mistura	$e_1$	$e_2$	$e$	Viés
$m_1$	0.2705	0.0437	0.3142	0.0130
$m_2$	0.1351	0.2361	0.3712	0.0060
$m_3$	0.0369	0.2233	0.2602	0.0080
$m_4$	0.1404	0.0002	0.1406	0.0884
$m_5$	0.0636	0.0200	0.0836	0.0015
$m_6$	0.0612	0.0350	0.0962	0.0006
$m_7$	0.0480	0.0105	0.0585	0.0097
$m_8$	0.0554	0.0951	0.1505	0.0024
$m_9$	0.0160	0.0538	0.0698	0.0096
$m_{10}$	0.1939	0.0211	0.2150	0.0993
$m_{11}$	0.0579	0.2401	0.2980	0.0173
$m_{12}$	0.0093	0.0770	0.0863	0.0143
$m_{13}$	0.0293	0.0046	0.0339	0.0015
$m_{14}$	0.0312	0.0265	0.0577	0.0002
$m_{15}$	0.0102	0.0407	0.0509	0.0020
$m_{16}$	0.1730	0.0128	0.1858	0.0078
$m_{17}$	0.1653	0.0999	0.2652	0.0024
$m_{18}$	0.0744	0.1518	0.2262	0.0194
$m_{19}$	0.1344	0.0513	0.1857	0.0181
$m_{20}$	0.1410	0.0914	0.2324	0.0078
$m_{21}$	0.0573	0.1408	0.1981	0.0207
$m_{22}$	0.1871	0.0090	0.1961	0.0790
$m_{23}$	0.0811	0.2251	0.3062	0.0124
$m_{24}$	0.0389	0.1978	0.2367	0.0251

Para ilustrar melhor o uso do discriminante baseado na mistura de cópulas, foi realizada a simulação de uma mistura com separação quase total entre os *clusters* formados. A simulação correspondeu à mistura

$$C(u, v) = pC_1(u, v; \theta_1) + (1 - p)C_2(u, v; \theta_2),$$

tal que  $C_1$  é uma cópula de Frank com parâmetro de dependência  $\theta_1 = -4$  e  $C_2$  uma cópula de Joe de parâmetro  $\theta_2 = 4$ . As distribuições marginais simuladas foram misturas de duas distribuições Gumbel, como definido na equação (5.3), de parâmetros de locação  $\mu_1 = 100$  e  $\mu_2 = 180$ , parâmetros de escala iguais a  $\sigma_1 = 10$  e  $\sigma_2 = 10$ , e parâmetros de forma  $\alpha_1 = 0$  e  $\alpha_2 = 0$ . Os percentuais das misturas das marginais e das cópulas foram ambos iguais a



$\pi_1 = \pi_2 = 0.5$ . No caso de as distribuições marginais serem compostas por misturas, há a extensão das equações (3.2) e (3.3), impactando na forma das funções discriminantes mostradas na Tabela 4.1, como descrito na equação (4.11) e ilustrado na seção 5.2 deste capítulo. A Distribuição das marginais e o gráfico de dispersão dos dados simulados são mostrados nas Figuras 5.7 e 5.8.

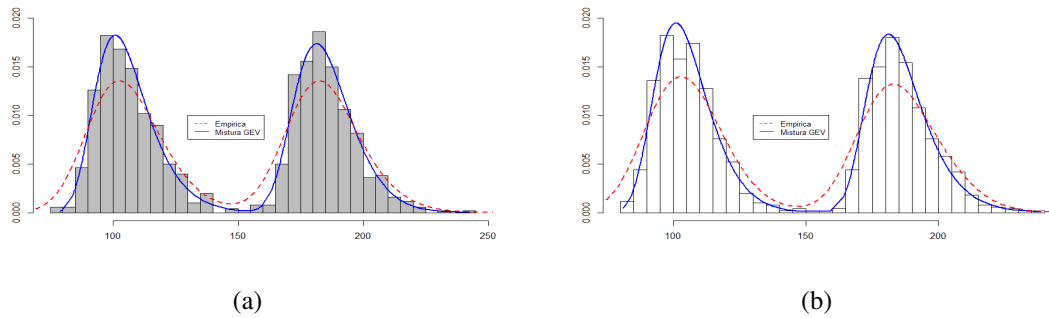


Figura 5.7: Histograma das variáveis X e Y simuladas, as densidades estimadas e suas respectivas densidades empíricas.

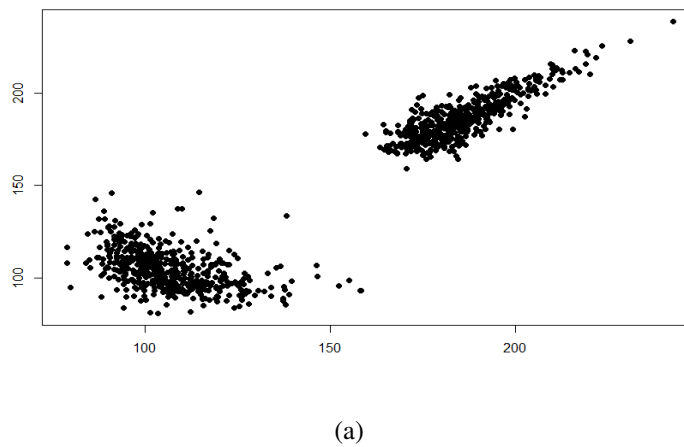


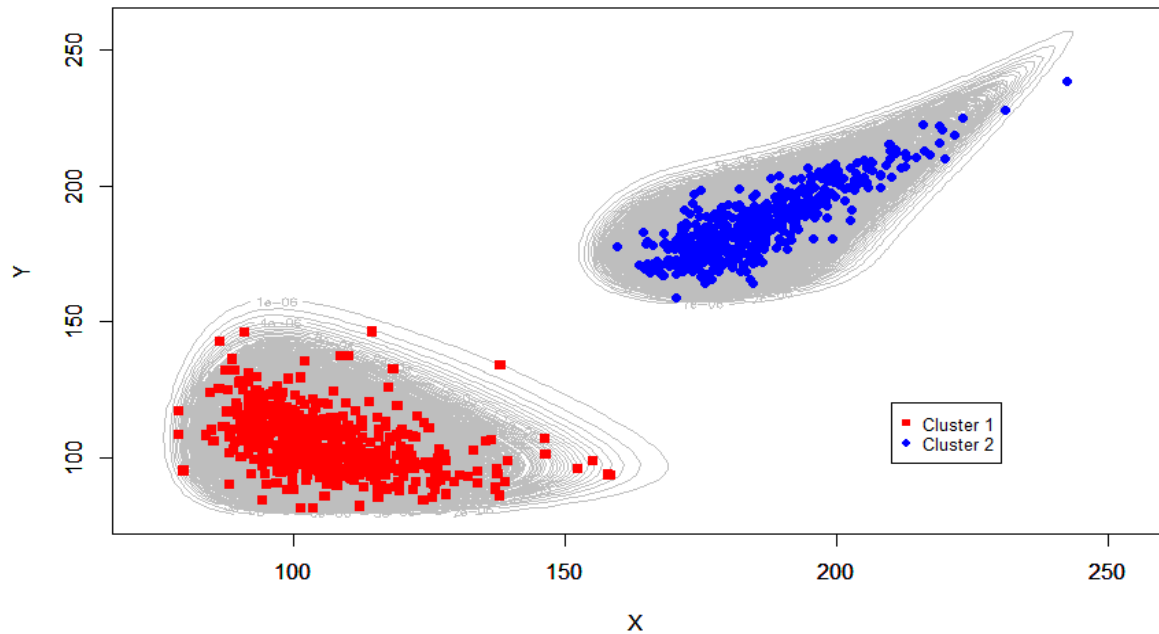
Figura 5.8: Gráfico de dispersão de X e Y.

Os parâmetros das distribuições marginais estimados estão dispostos na Tabela 5.6, sendo os valores estimados para a mistura  $p = 0.5030$ ,  $\theta_1 = -1.4833$  e  $\theta_2 = 2.8625$ .

Tabela 5.6: Parâmetros estimados das distribuições marginais de X e Y, considerando distribuição GEV com  $\mu_i$  parâmetro de locação,  $\sigma_i$  parâmetro de escala e  $\alpha_i$  parâmetro de forma, com  $i = 1, 2$ .

Variável	$\mu_1$	$\sigma_1$	$\alpha_1$	$\pi_1$	$\mu_2$	$\sigma_2$	$\alpha_2$
X	100.498	10.1889	-0.0293	0.5014	180.0859	10.0200	-0.0488
Y	100.2182	9.5323	-0.0800	0.5032	180.7178	9.9804	-0.0427

As curvas de níveis da densidade conjunta estimada do vetor  $(X, Y)$  simulado é apresentada na Figura 5.9.



(a)

Figura 5.9: Curvas de níveis da densidade de  $X$  e  $Y$ , com separação de *clusters* pela função discriminante da mistura.

A função classificou corretamente todas as observações, 503 da componente 1 e 497 da componente 2.

Foi realizada tentativa de estimar os dados simulados via outras misturas de cópulas, porém o algoritmo convergiu somente para a verdadeira mistura simulada e para mistura de duas cópulas de Frank, como mostra a tabela abaixo.

Tabela 5.7: Tabela comparativa de valores da função de log verossimilhança, AIC e BIC para algumas misturas de cópulas estimadas.

Mistura	$\hat{p}$	$\hat{\theta}_1$	$\hat{\theta}_2$	Log Verossimilhança
Frank - Joe	0.5030	-1.4832	2.8625	-0.8259
Frank - Frank	0.5030	-6.1211	-4.3252	-1.0085

Mostrando que as verdadeiras componentes tiveram um resultado de estimação mais adequado.

## 5.2 Aplicação

### 5.2.1 Dados de erupções de Geysers Old Faithful em Yellowstone National Park, Wyoming, EUA.

Com o objetivo de analisar a adequação do modelo de mistura de cópulas, foi feita aplicação em dados de erupções do Geysers Old Faithful em Yellowstone National Park, Wyoming, EUA. a base de dados possui informações do tempo de duração de erupção de Geysers e o tempo até a próxima erupção do geysers. Essa base de dados pode ser encontrada no software R. As Figuras 5.10 e 5.11 mostram os histogramas com a distribuição dessas duas variáveis e o gráfico de dispersão mostrando a relação de dependência entre as duas. Observa-se que as marginais são compostas de misturas, pois são bimodais. Além disso, percebe-se na Figura 5.11 a formação de dois *clusters*.

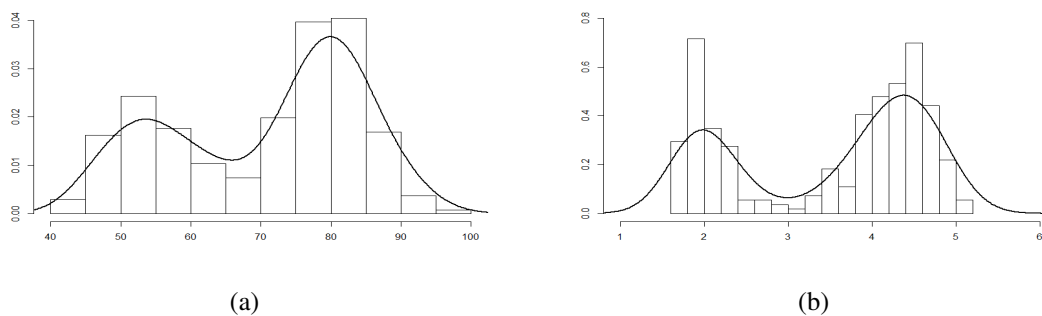


Figura 5.10: Histograma das variáveis, (a): "tempo até próxima erupção" e (b): "tempo de duração da erupção" e suas respectivas densidades empíricas.

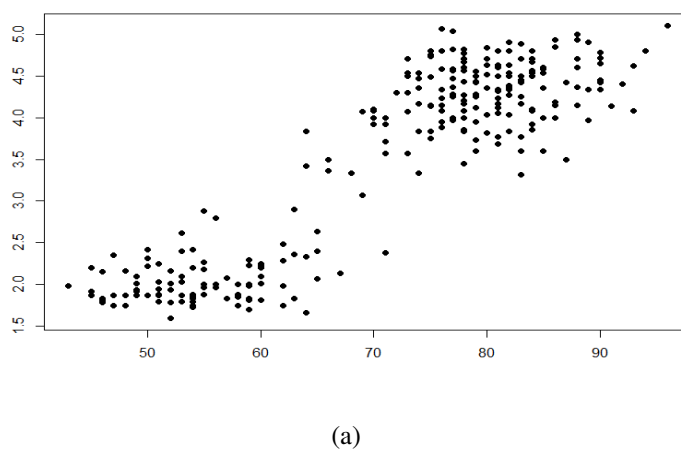


Figura 5.11: Gráfico de dispersão do "tempo até próxima erupção" e "tempo de duração da erupção".

O objetivo é modelar a distribuição conjunta das duas variáveis além de classificar

os pares de observações aos *clusters* observados.

Como explicado no capítulo 3, primeiramente foram estimadas as distribuições marginais dos dados. Estimou-se a mistura de duas distribuições de valor extremo generalizadas (GEV), obtendo os parâmetros mostrados na Tabela 5.8 e cuja função de densidade está dada na equação (5.2).

$$F(x; \mu, \sigma, \alpha) = \begin{cases} e^{-(1+\alpha\frac{x-\mu}{\sigma})^{-\frac{1}{\alpha}}} & \text{se } \alpha \neq 0 \\ e^{-e^{-\frac{x-\mu}{\sigma}}} & \text{se } \alpha = 0 \end{cases} \quad (5.1)$$

$$f(x; \mu, \sigma, \alpha) = \begin{cases} \frac{1}{\sigma} (1 + \alpha\frac{x-\mu}{\sigma})^{-\frac{1}{\alpha}-1} e^{-(1+\alpha\frac{x-\mu}{\sigma})^{-\frac{1}{\alpha}}} & \text{se } \alpha \neq 0 \\ \frac{1}{\sigma} e^{-\frac{x-\mu}{\sigma}} e^{-e^{-\frac{x-\mu}{\sigma}}} & \text{se } \alpha = 0 \end{cases} \quad (5.2)$$

$$g = \pi_1 f(x; \mu_1, \sigma_1, \alpha_1) + \pi_2 f(x; \mu_2, \sigma_2, \alpha_2), \quad (5.3)$$

em que  $\pi_2 = 1 - \pi_1$ . Quando  $\alpha = 0$ , a distribuição GEV torna-se uma distribuição Gumbel.

Tabela 5.8: Parâmetros das distribuições do "tempo até próxima erupção",  $g_1$ , e "tempo de duração da erupção",  $g_2$ ; em que  $g_1$  e  $g_2$  seguem a forma da equação (5.3).

Marginal	$\pi_1$	$\mu_1$	$\mu_2$	$\sigma_1$	$\sigma_2$	$\alpha_1$	$\alpha_2$
$g_1$	0.3997	52.7098	78.5251	6.1483	5.2074	0.0068	-0.2140
$g_2$	0.3796	1.9168	4.2217	0.2437	0.4502	0.5249	-0.4985

A Figura 5.12 apresenta o ajuste das duas misturas de densidades ajustadas aos dados.

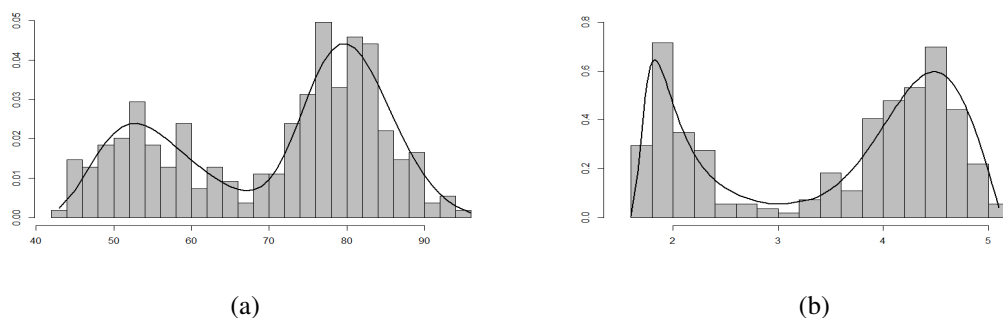


Figura 5.12: Densidade das variáveis, (a): "tempo até próxima erupção" e (b): "tempo de duração da erupção" ajustadas aos dados.

Como foi observado graficamente que as distribuições marginais GEV se ajustavam bem aos dados, então foi testado uma mistura de duas cópulas Gumbel para modelar a distribuição conjunta do tempo até uma próxima erupção e o tempo de duração da erupção do geyser. Então foi ajustada a seguinte mistura de cópulas

$$C(u, v) = pC_1(F_1(x; \mu_1, \sigma_1, \alpha_1), G_1(y; \mu_2, \sigma_2, \alpha_2); \theta_1) + (1 - p)C_2(F_2(x; \mu_1, \sigma_1, \alpha_1), G_2(y; \mu_2, \sigma_2, \alpha_2); \theta_2), \quad (5.4)$$

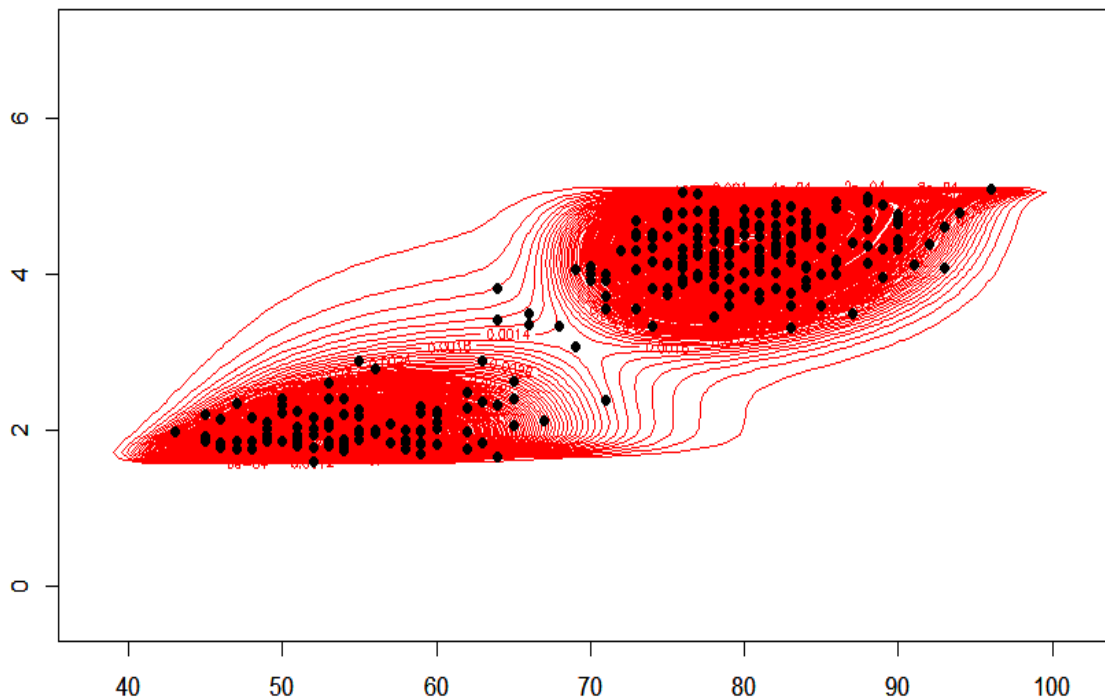
sendo  $C_1$  e  $C_2$  ambas cópulas de Gumbel com densidades marginais dadas por  $F_1(x, 52.7098, 6.1483, 0.0068)$ ,  $F_2(x, 78.5251, 5.2074, -0.2140)$ ,  $G_1(y, 1.9168, 0.2437, 0.5249)$  e  $G_2(y, 4.2217, 4.0.4502, -0.4985)$ , tais que cada uma das distribuições são iguais à função (5.1). De forma que,

$$g_1 = \pi_1 F_1'(x, 52.7098, 6.1483, 0.0068) + \pi_2 F_2'(x, 78.5251, 5.2074, -0.2140)$$

e

$$g_2 = \pi_1 G_1'(y, 1.9168, 0.2437, 0.5249) + \pi_2 G_2'(y, 4.2217, 4.0.4502, -0.4985),$$

sendo  $F_1'$ ,  $F_2'$ ,  $G_1'$  e  $G_2'$  as respectivas derivadas de  $F_1$ ,  $F_2$ ,  $G_1$  e  $G_2$ . Observe que a expressão (5.4) é uma extensão da equação (3.2), esta foi construída sob o pressuposto de que  $F_1 = F_2$  e  $G_1 = G_2$ . As alterações na função discriminante são mostradas na equação (4.11). As estimativas de máxima verossimilhança para os parâmetros foram  $\hat{p} = 0.3824$ ,  $\hat{\theta}_1 = 1.5215$  e  $\hat{\theta}_2 = 1.1901$ . Com base na Figura 5.13, que mostra as curvas de níveis da função densidade da mistura da equação (5.4) e a plotagem dos dados sobre as curvas de níveis, que o ajuste foi adequado. Também observa-se de forma mais clara a formação de dois *clusters*.



(a)

Figura 5.13: Curvas de níveis da densidade conjunta dos dados considerando mistura de duas cópulas Gumbel.

Foram testadas outras misturas, com marginais iguais às listadas na equação (5.4), e também a estimação considerando apenas uma cópula de Gumbel, isto é, sem mistura, e as marginais tais como na equação (5.3). Os resultados dessa comparação se encontram na Tabela 5.9 e na Figura 5.14. Constata-se que a mistura de duas cópulas de Joe obteve melhor ajuste, por possuir log verossimilhança maior e menores valores de AIC e BIC, além de melhor ajuste graficamente.

Tabela 5.9: Tabela comparativa de valores da função de log verossimilhança, AIC e BIC para algumas misturas de cópulas estimadas.

Mistura	$\hat{p}$	$\hat{\theta}_1$	$\hat{\theta}_2$	Log Verossimilhança
Gumbel - Gumbel	0.3824	1.5215	1.1901	1.8164
Frank - Frank	0.3801	2.9351	1.4931	1.7933
Joe - Joe	0.3883	2.0355	1.2280	1.8175
Gumbel (sem mistura)	1.0000	1.9177	Não se aplica	-1215.6170

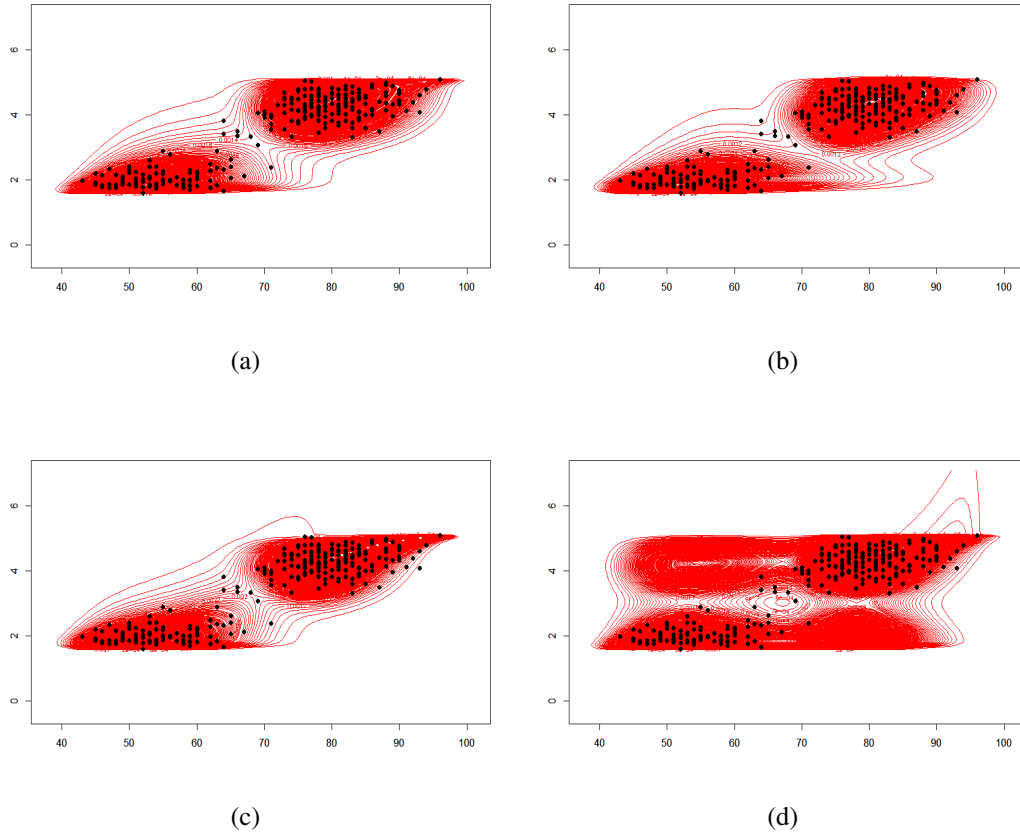


Figura 5.14: Curvas de níveis da densidade conjunta dos dados considerando misturas de cópulas: a) Gumbel-Gumbel; b) Frank-Frank; c) Joe-Joe; e d) Gumbel, sem mistura. Os parâmetros das misturas estão na Tabela 5.9.

Como a mistura de cópulas Joe teve um ajuste melhor nos dados, foi aplicada função discriminante construída com base nesta mistura para separar as observações nos dois *clusters* observados.

Assim, a função discriminante estimada para a mistura de cópulas Joe para os dados em questão é dada abaixo.

$$\alpha = \log \frac{1 - \hat{p}}{\hat{p}} = \frac{0.6117}{0.3883} = 1.5753284;$$

$$\beta_1 = \hat{\theta}_2 - 1 = 1.2280 - 1 = 0.2280;$$

$$\beta_2 = 1 - \hat{\theta}_1 = 1 - 2.0355 = -1.0355;$$

$$z_1 = \log((1 - F_1(x; \hat{\mu}, \hat{\sigma}, \hat{\alpha}))(1 - F_2(y; \hat{\mu}, \hat{\sigma}, \hat{\alpha})));$$

$$z = \log((1 - G_1(x; \hat{\mu}, \hat{\sigma}, \hat{\alpha}))(1 - G_2(y; \hat{\mu}, \hat{\sigma}, \hat{\alpha})));$$

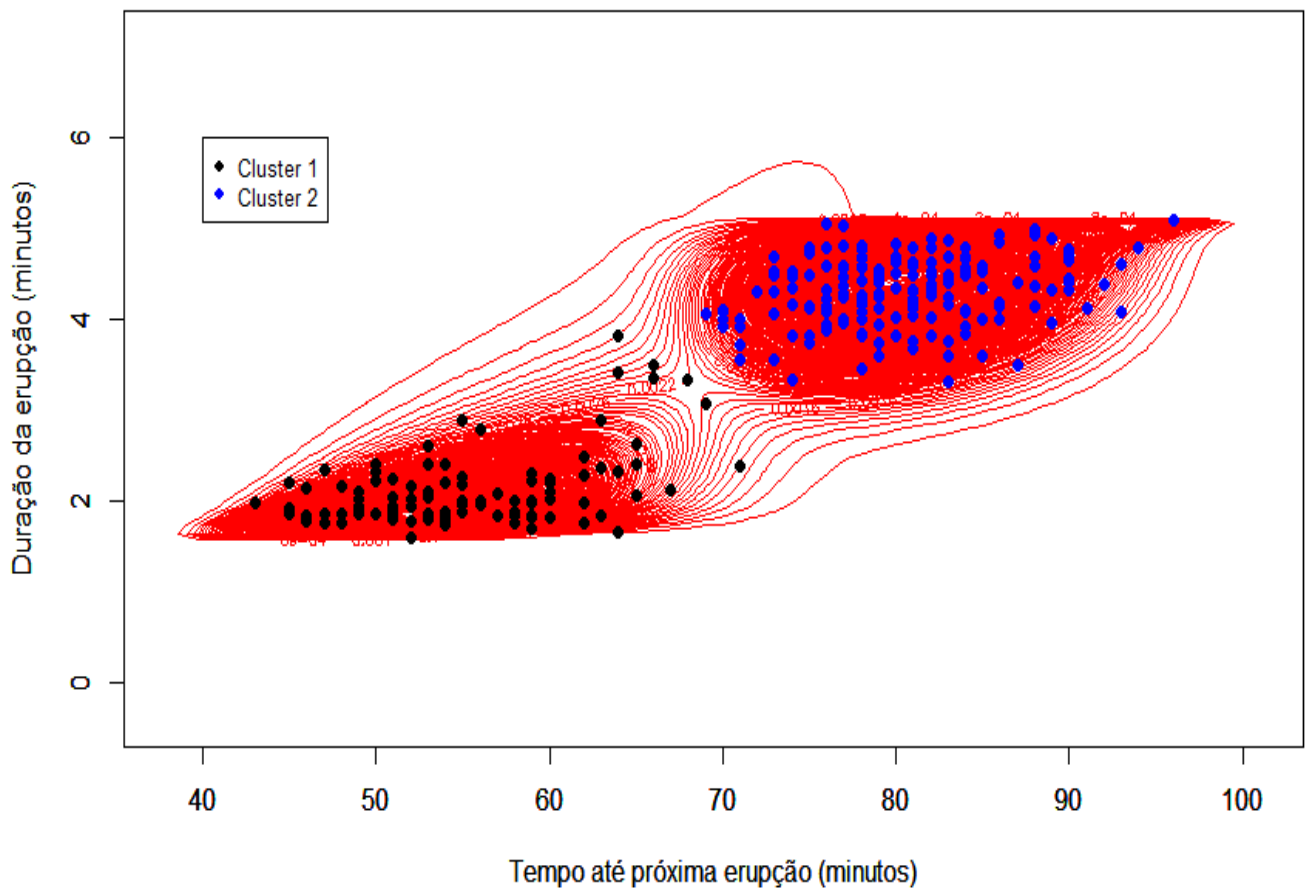
e

$$z_3 = \log(J(x, y; \hat{\phi}_1, \hat{\phi}_2, \hat{\theta}_1, \hat{\theta}_2)),$$

em que  $J(x, y; \hat{\phi}_1, \hat{\phi}_2, \hat{\theta}_1, \hat{\theta}_2)$  tal como na equação (4.10). A função discriminante para os dados é dada por

$$NL(x, y) = \alpha + \beta_1 Z_1 + \beta_2 Z_2 + Z_3.$$

Os parâmetros das funções de distribuições marginais  $F_1, F_2, G_1, G_2$  são os mesmos dados na Tabela 5.8, sendo que suas fórmulas se encontram na equação 5.1. Assim como as densidades marginais, os parâmetros são os mesmos da Tabela 5.8 e fórmulas são das na equação 5.2. A Figura 5.15 mostra o resultado da classificação nos dois *clusters* pelo discriminante calculado. Foram classificadas 103 observações no *cluster 1* (preto) e 169 observações no *cluster 2* (azul).



(a)

Figura 5.15: Curvas de níveis da densidade conjunta plotada contra os dados, com separação de *cluster* pela função discriminante da mistura



# Capítulo 6

## Conclusão

Neste trabalho foram apresentados métodos para modelagem de dados bivariados com formação de dois *clusters*, através da mistura de cópulas. Além disso, foram propostos discriminantes não-lineares de Bayes para mistura de cópulas bivariadas arquimedianas específicas, com uma variedade de simulações para analisar o desempenho das funções discriminantes estimando os erros de classificação por meio da técnica Bootstrap. Os resultados foram satisfatórios, visto que os erros de classificação apresentados pelos estimadores foram pequenos. Ficou demonstrado no trabalho as diversas possibilidades de configurações da estrutura da relação de dependência dos dados que podem ser modeladas por mistura de cópulas.

Com relação à aplicação da metodologia em dados reais, observou-se a boa adequação do uso da modelagem por mistura de cópulas. Sendo demonstrado graficamente, a eficácia da função discriminante em classificar corretamente as observações em cada um dos *clusters*.

Com relação ao processo de estimação, devido à quantidade de parâmetros e à complexidade das funções de log verossimilhança estimadas, constatou-se que a abordagem via algoritmo EM é mais vantajosa, do que a estimação direta por máxima verossimilhança. Além disso, foi observado que algoritmos com base no método de Newton-Raphson têm pouca eficácia na estimação da mistura de cópulas, tendo sido adotados algoritmos mais sofisticados como é o caso do algoritmo PSO.

Por fim, para trabalhos futuros seria interessante estudar formas de se estimar o erro da funções discriminantes propostas quando não se tem conhecimento sobre a qual população os dados pertencem.

# Referências Bibliográficas

- Arakelian, V. e Karlis, D. (2014). Clustering dependencies via mixtures of copulas. *Communications in Statistics-Simulation and Computation*, 43(7):1644–1661. 2, 14
- Böhning, D. (1999). *Computer-assisted analysis of mixtures and applications: meta-analysis, disease mapping and others*, volume 81. CRC press. 1
- Casella, G. e Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA. 23
- Efron, B. (1979). Bootstrap methods: another look at the jackknife *annals of statistics* 7: 1–26. *View Article PubMed/NCBI Google Scholar*. 24
- Efron, B. (1982). *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam. 24
- Efron, B. e Tibshirani, R. (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis. 24
- Embrechts, P., Lindskog, F., e McNeil, A. (2001). Modelling dependence with copulas and applications to risk management. *zurich*. 10
- Huberty, C. (1994). *Applied Discriminant Analysis*. Wiley Series in Probability and Statistics. Wiley. 26
- Joe, H. (2005). Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94(2):401–419. 23
- Karlis, D. e Santourian, A. (2009). Model-based clustering with non-elliptically contoured distributions. *Statistics and Computing*, 19(1):73–83. 1
- Lai, Y., Chen, C. W., e Gerlach, R. (2009). Optimal dynamic hedging via copula-threshold-garch models. *Mathematics and Computers in Simulation*, 79(8):2609–2624. 2
- McLachlan, G. e Peel, D. (2000). *Finite Mixture Models*, (1 ed.). Wiley Series in Probability and Statistics. Wiley-Interscience. 1, 14, 24

Melnykov, V., Maitra, R., et al. (2010). Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116. 1

Nelsen, R. B. (2006). *An introduction to copulas*, (2nd ed.). Springer series in statistics. Springer. 4, 5, 6, 8, 10, 11

Sklar, A. (1973). Random variables, joint distribution functions, and copulas. *Kybernetika*, 9(6):449–460. 4

Team, R. C. (2018). R: A language and environment for statistical computing, r foundation for statistical computing, austria, 2015. 33

# Apêndice A

## Códigos R

Neste apêndice apresenta-se os códigos implementados em linguagem R de computação estatística (R Core Team, 2018) de Simulação e Estimação de Misturas de Cópulas.

```
## Algoritmo PSO para minimizar funções
pso_0=function (AX1, fun, ITER, erro, NPART) {
options(digits = 22)
f=fun
X=AX1
Z=list(rep(0, NPART))
Y=rep(Z, 10000)
j=1
W=c()

#g=X[which.max(X[[j]]),]
criterio.Parada=ifelse(pi<0, T, F)

while(sum(criterio.Parada)!=1 ){

#tratar restrição de espaço paramétrico específico

for(i in 1:NPART){

Y[[j]][i]=ifelse((X[i,1:1]<=0.0 | X[i,1:1]>=1.0)>0.0
, Inf, f(as.vector(X[i,1:1])))

}

if(j==1){
```

```

p=X
g=X[which.min(abs(Y[[j]])),]
V=matrix(0,NPART,1)
Vi=V
gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V
# W[j]=f(g)
}else{
px=matrix(0,NPART,1)
#tratar restri??o de espa?o param?trico espec?fico

AA=((p<=0.0)+(X<=0.0))+((p>=1.0)+(X>=1.0))

for(i in 1:NPART){

if(iffelse(AA[i]<1,abs(f(p[i,]))-abs(f(X[i,])),0)<0){
px[i,]=p[i,]
}else{
px[i,]=X[i,]
}
}

p=px
gk=g
gx=X[which.min(abs(Y[[j]])),]
if(iffelse(is.nan(abs(f(gx))-abs(f(g))),1,
abs(f(gx))-abs(f(g))<0){g=gx}else{g=g}
indicador=iffelse(is.nan(abs(f(gx))-abs(f(g))),1,0)
if(indicador==1){g=g}else{
gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
Vi=V
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X

```

```

X=Xi+V
}
}

W[j]=ifelse(is.nan(abs(f(g))), f(gk), f(g))

maxj=max(W)
minj=min(W)
var.global=ifelse((maxj-minj)==0, 1, maxj-minj)
if(j>ITER){
maxj5=max(W[(j-5):j])
minj5=min(W[(j-5):j])

varj5=maxj5-minj5
}else{
varj5=var.global
}

vard=varj5/var.global

#print(list(X_Otimo=g, F_otimo=W[j], Iteracoes=j))

criterio.Parada=ifelse(is.nan(abs(f(g))), T,
ifelse(j>ITER, ifelse((1/erro)*vard<1, T, F), F))
j=j+1

}
g=ifelse(is.nan(abs(f(g))), gk, g)
saida=list(X_Otimo=g, F_otimo=W[j-1], Iteracoes=j-1)
return(saida)
}

### Procedimentos intermediários a serem usados
### na função que gera Misturas
gera_vet_aleatorio=function(a,b,n){

x1=(b-a)*runif(n)+a

#x2=(b-a)*runif(n)+a

```

```

matrix(c(x1),n,1)
}
gera_vet_aleatorio1=function(a,b,n){

x1=(b-a)*runif(n)+a

#x2=(b-a)*runif(n)+a

matrix(c(x1),n,1)
}
gera_vet_aleatorio2=function(a,b,n){

x1=(b-a)*runif(n)+a

matrix(c(x1),n,1)
}

c_1_u = function(u,v,q,theta){

fv = exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta)
)*(((-log(u))^theta+(-log(v))^theta)^(1/theta-1)
)*((-log(u))^(theta-1))*(1/u)-q
fv
}

c_2_u = function(u,q,theta){

fv = (1-u^(-theta)+(q*u^(theta+1))^(-theta/(theta
+1)))^(-1/theta)
fv
}

c_3_u = function(u,q,theta){

fv = -(1/theta)*log((exp(-theta*u)+q*(exp(-theta)-exp(-theta*u))
)/(exp(-theta*u)-q*(exp(-theta*u)-1)))
fv
}

```

```

}

c_4_u = function(u,v,q,theta){

fv = (((1-u)^theta + (1-v)^theta - ((1-u)^theta)*((1-v)^theta))^(1/theta)
)* (1-(1-v)^theta)-q
fv
}

## Função gerado das misturas para as cópulas Clayton, Gumbel,
##Joe e Frank.
Gera_Mistura = function(n,copula1,copula2,p,theta1,theta2){

if((copula1=="Clayton" | copula1=="Gumbel"
| copula1=="Frank" | copula1=="Joe" ) &
(copula2=="Clayton" | copula2=="Gumbel" |
copula2=="Frank" | copula2=="Joe" )){

copula1 = copula1
copula2 = copula2
p = p
theta1 = theta1
theta2 = theta2
n=n

U1=c()
V1=c()
U2=c()
V2=c()

for( i in 1:n){

p0=runif(1)

if(p0 <= p){

u=runif(1)

```



```

q=runif(1)
theta=theta1

U1[i]=u
if(copula1 == "Gumbel"){

V1[i]=pso_0(gera_vet_aleatorio(0,1,40),function(v) c_1_u(u,v,q,
theta),19,10^(-6),40)$X_Otimo
}else{

if(copula1 == "Joe"){
V1[i]=pso_0(gera_vet_aleatorio(0,1,40),function(v) c_4_u(u,
v,q,theta),19,10^(-6),40)$X_Otimo
}else{

if(copula1 == "Clayton"){
V1[i]=c_2_u(u,q,theta)
}else{
V1[i]=c_3_u(u,q,theta)
}
}
}
}else{

u=runif(1)
q=runif(1)
theta=theta2

U2[i]=u

if(copula2 == "Gumbel"){

V2[i]=pso_0(gera_vet_aleatorio(0,1,40),function(v) c_1_u(u,v,q,
theta),19,10^(-6),40)$X_Otimo
}else{

if(copula2 == "Joe"){
V2[i]=pso_0(gera_vet_aleatorio(0,1,40),function(v) c_4_u(u,v,q,
theta),19,10^(-6),40)$X_Otimo
}else{

```

```

if(copula2 == "Clayton"){
V2[i]=c_2_u(u,q,theta)
}else{
V2[i]=c_3_u(u,q,theta)
}

}

}

}

}

}

( saida=list(m1=cbind(U1,V1),m2=cbind(U2,V2)))
}else{
"Erro, nome da cópula está errado. Tente nomes
nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
}
}

```

```
mist = Gera_Mistura(1000,"Frank","Clayton",0.6,-5,5)
```

```

m1=mist$m1[-which(is.na(mist$m1[,1])),]
m2=mist$m2[-which(is.na(mist$m2[,1])),]
m=rbind(m1,m2)

```

```

length(m1[,1])
length(m2[,2])

```

```
library(extRemes)
```

```
x1=qevd(m1[,1],loc=336.53,scale=67.81,shape=0.0,  
type = "Gumbel")
```

```
y1=qevd(m1[,2],loc=89.78,scale=43.21,shape=0.00,  
type = "Gumbel")
```

```
x2=qevd(m2[,1],loc=336.53,scale=67.81,shape=0.0,  
type = "Gumbel")
```

```
y2=qevd(m2[,2],loc=89.78,scale=43.21,shape=0.00,  
type = "Gumbel")
```

```
x=qevd(m[,1],loc=1.5,scale=2,shape=0.0,  
type = "Gumbel")
```

```
y=qevd(m[,2],loc=-1,scale=1,shape=0.00,  
type = "Gumbel")
```

```
plot(x,y)
```

```
#####  
### Estima mistura de cópulas  
#####
```

```
## Supondo marginais GEV
```

```
modx = fevd(x,type = "GEV",method = "MLE",period.basis = "year")
```

```
round(modx$results$par,2)
```

```
plot(modx, "qq",main="")
```

```
#Acumudala
```

```
FModEstx=function(t) {
```

```

a=as.numeric(modx$results$par[3])
s=as.numeric(modx$results$par[2])
u=as.numeric(modx$results$par[1])

exp(-(1+(a/s)*(t-u))^-1/a)

}

densfx=function(t){
a=as.numeric(modx$results$par[3])
s=as.numeric(modx$results$par[2])
u=as.numeric(modx$results$par[1])
devd(t, loc = u, scale = s, shape = a,
type = "GEV")

}

mody = fevd(y,type = "GEV",method = "MLE",period.basis = "year")

round(mody$results$par,2)
plot(mody, "qq",main="")

#Acumudala
FModEsty=function(t){
a=as.numeric(mody$results$par[3])
s=as.numeric(mody$results$par[2])
u=as.numeric(mody$results$par[1])

exp(-(1+(a/s)*(t-u))^-1/a)

}

densfy=function(t){
a=as.numeric(mody$results$par[3])
s=as.numeric(mody$results$par[2])
u=as.numeric(mody$results$par[1])
devd(t, loc = u, scale = s, shape = a,
type = "GEV")

```

```
}
```

```
dgumbel=function(x,y,theta,Fx,Gy,fx,gy) {  
u=Fx(x)  
v=Gy(y)  
fx(x)*gy(y)*(1/(u*v))*(log(u)*log(v))^(theta-1)*exp(-  
(-log(u))^theta+(-log(v))^theta)^(1/theta))*((theta-  
1)*((-log(u))^theta+(-log(v))^theta)^(1/theta-2)+(  
(-log(u))^theta+(-log(v))^theta)^(2/theta-2))  
}
```

```
dclayton=function(x,y,theta,Fx,Gy,fx,gy) {  
u=Fx(x)  
v=Gy(y)  
fx(x)*gy(y)*u^(-theta+1)*v^(-theta+1)*(theta+1  
)*ifelse((u^(-theta)+v^(-theta)-1)<=0.0,0.00,(u^(-theta)  
+v^(-theta)-1)^(-(1/theta+2)))  
}
```

```
dFrank=function(x,y,theta,Fx,Gy,fx,gy) {  
u=Fx(x)  
v=Gy(y)  
fx(x)*gy(y)*(-theta)*exp(-theta*u)*exp(-theta*v)*(  
(exp(-theta)-1)/(exp(-theta)-1+  
(exp(-theta*u)-1)*(exp(-theta*v)-1))^2)  
}
```

```
dJoe=function(x,y,theta,Fx,Gy,fx,gy) {  
u=Fx(x)  
v=Gy(y)
```

```
H=((1-u)^theta + (1-v)^theta - ((1-u)^theta)*(  
(1-v)^theta))^(1/theta-1)  
H1=(theta-1)*((1-u)^theta + (1-v  
)^theta - ((1-u)^theta)*((1-v)^theta)  
)^(1/theta-2)*((1-u)^(theta-1)-(1-u  
)^(theta-1))*(1-v)^theta  
G=(1-v)^(theta-1)-(1-v)^(theta-1)*(1-u)^theta  
G1=theta*(1-v)^(theta-1)*(1-u)^(theta-1)
```

```
fx(x)*gy(y)*(H1*G+H*G1)
}
```

```
dJoe(x[911],y[911],3,Fx,Gy,fx,gy)
x=x
y=y
fx=densfx
gy=densfy
Fx=FModEstx
Gy=FModEsty
```

```
Ddclayton=function(theta){
u=Fx(x)
v=Gy(y)
h1=(u*v)^(-(theta+1))*(theta+1)
h11=(u*v)^(-(theta+1))*(1-(theta+1)*log(u*v))
h2=ifelse((u^(-theta)+v^(-theta)-1)<=0.0,0.00,
(u^(-theta)+v^(-theta)-1)^(-(1/theta+2)))
h21=ifelse((u^(-theta)+v^(-theta)-1)<0.0,0.00,
h2*((1/theta^2)*log(u^(-theta)+v^(-theta)-1)+
(1/theta+2)*(1/(u^(-theta)+v^(-theta)-1))*(u^(-theta)
)*log(u)+v^(-theta)*log(v)))
fx(x)*gy(y)*(h11*h2+h1*h21)
}
```

```
Ddgumbel=function(theta){
u=Fx(x)
v=Gy(y)
```

```
B_0=(-log(u))^theta+(-log(v))^theta
B_0L=log((-log(u))*(-log(u))^theta+log((-log(v))
)*(-log(v))^theta)
J_0=B_0^(1/theta)
J_0L=-1/theta*J_0*((1/theta)*log(B_0)-(B_0L/B_0))
```

```

H1_0=B_0^(2/theta-2)
H1_0L=H1_0*(-2*(1/theta^2)*log(B_0)+(2/theta-2)*(B_0L/B_0))
H2_0=(theta-1)*B_0^(1/theta-2)
H2_0L=B_0^(1/theta-2)+(theta-1)*B_0^(1/theta-2
)*(-(1/theta^2)*log(B_0)+(1/theta-2)*(B_0L/B_0))

F_0=(log(u)*log(v))^(theta-1)
F_0L=F_0*log(log(u)*log(v))
G_0=exp(-J_0)
G_0L=G_0*(-J_0L)
H_0=H1_0+H2_0
H_0L=H1_0L+H2_0L

fx(x)*gy(y)*(1/(u*v))*(F_0L*G_0+F_0*G_0L)*H_0+(F_0*G_0)*H_0L)
}

DdFrank=function(theta){
u=Fx(x)
v=Gy(y)
f=-theta*exp(-theta*u)
f2=exp(-theta*v)
f1=exp(-theta*u)*(theta*u-1)
f21=-v*exp(-theta*v)
g1=f*f2
g11=f1*f2+f*f21
g2=exp(-theta)-1
g21=-exp(-theta)
h1=g1*g2
h11=g11*g2+g1*g21
gx1=exp(-theta)-1
gx11=-exp(-theta)
gx2=exp(-theta*u)-1
gx21=-u*exp(-theta*u)
gx3=exp(-theta*v)-1
gx31=-v*exp(-theta*v)
gx=gx1+gx2*gx3
gx1=gx11+gx21*gx3+gx2*gx31
h2=gx^2
h21=2*gx*gx1
fx(x)*gy(y)*(h11*h2-h1*h21)/h2^2

```

```
}
```

```
DdJoe=function(theta){
```

```
u=Fx(x)
```

```
v=Gy(y)
```

```
jx=1-u
```

```
jy=1-v
```

```
j1=jx^theta
```

```
j11=j1*log(jx)
```

```
j2=jy^theta
```

```
j21=j2*log(jy)
```

```
j1x=jx^(theta-1)
```

```
j1x1=j1x*log(jx)
```

```
j2y=jy^(theta-1)
```

```
j2y1=j2y*log(jy)
```

```
g=j1+j2-j1*j2
```

```
g1=j11+j21-(j11*j2+j1*j21)
```

```
h1=(theta-1)*g^(1/theta-2)
```

```
h11=g^(1/theta-2)+(theta-1)*(g^(1/theta-2)  
)*(-(1/theta^2)*log(g)+(1/theta-2)*(g1/g))
```

```
h2=j2y-j2y*j1
```

```
h21=j2y1-(j2y1*j1+j2y*j11)
```

```
H1=h1*h2
```

```
H1L=h11*h2+h1*h21
```

```
H2=j1x-j1x*j2
```

```
H2L=j1x1-(j1x1*j2+j1x*j21)
```

```
H3=g^(1/theta-1)
```



```
H3L=(g^(1/theta-1))*(-(1/theta^2)*log(g)+(1/theta-1)*(g1/g))
```

```
H4=theta*j1x*j2y
```

```
H4L=j1x*j2y+theta*j1x1*j2y+theta*j1x*j2y1
```

```
fx(x)*gy(y)*(H1L*H2+H1*H2L+H3L*H4+H3*H4L)
}
```

```
pso_min=function(AX, funt, ITER, erro, NPART, copula1, copula2) {
```

```
options(digits = 22)
```

```
f=funt
```

```
X=AX
```

```
Z=list(rep(0, NPART))
```

```
Y=rep(Z, 10000)
```

```
j=1
```

```
W=c()
```

```
#g=X[which.min(Y[[j]]),]
```

```
criterio.Parada=ifelse(pi<0, T, F)
```

```
while(sum(criterio.Parada)!=1 ) {
```

```
for(i in 1:NPART) {
```

```
if(copula1=="Frank") {
```

```
Par_Rest_1=X[i,1]==0.0
```

```
}else{
```

```
if(copula1=="Clayton") {
```

```
Par_Rest_1=X[i,1]<=0.0
```

```
}else{
```

```
Par_Rest_1=X[i,1]<1.0
```

```
}
```

```
}
```

```
#####3
```

```
if(copula2=="Frank") {
```

```

Par_Rest_2=X[i,1]==0.0
}else{
if(copula2=="Clayton"){
Par_Rest_2=X[i,1]<=0.0
}else{
Par_Rest_2=X[i,1]<1.0
}
}

Y[[j]][i]=ifelse((Par_Rest_1+Par_Rest_2)>0, Inf
,f(as.vector(X[i,1:2])))

}

if(j==1){
p=X
g=X[which.min(abs(Y[[j]])),]
V=matrix(0,NPART,2)
Vi=V
gb=matrix(g,NPART,2)
for(k in 1:NPART){
gb[k,1:2]=g
}
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V
# W[j]=f(g)
}else{
px=matrix(0,NPART,2)

#####
if(copula1=="Frank"){
Par_Rest_1=X[,1]==0.0
}else{
if(copula1=="Clayton"){
Par_Rest_1=X[,1]<0.0
}else{
Par_Rest_1=X[,1]<1.0
}
}
}

```

```
}  
}
```

```
#####  
if(copula2=="Frank"){  
Par_Rest_2=X[,1]==0.0  
}else{  
if(copula2=="Clayton"){  
Par_Rest_2=X[,1]<0.0  
}else{  
Par_Rest_2=X[,1]<1.0  
}  
}
```

```
#####  
#####  
if(copula1=="Frank"){  
Par_Rest_1x=p[,1]==0.0  
}else{  
if(copula1=="Clayton"){  
Par_Rest_1x=p[,1]<0.0  
}else{  
Par_Rest_1x=p[,1]<1.0  
}  
}
```

```
#####  
  
if(copula2=="Frank"){  
Par_Rest_2x=p[,1]==0.0  
}else{  
if(copula2=="Clayton"){  
Par_Rest_2x=p[,1]<0.0  
}else{  
Par_Rest_2x=p[,1]<1.0  
}  
}  
}
```

```
#####
```

```
AA1=Par_Rest_1+Par_Rest_2
```

```

AA2=Par_Rest_1x+Par_Rest_2x
AA=AA1+AA2

for(i in 1:NPART){
if(iffelse(AA[i]<1,abs(f(p[i,]))-abs(f(X[i,])),1)<0){
px[i,]=p[i,]
}else{
px[i,]=X[i,]
}
}
p=px
gk=g
gx=X[which.min(abs(Y[[j]])),]
if(abs(f(gx))-abs(f(g))<0){g=gx}else{g=g}

gb=matrix(g,NPART,2)
for(k in 1:NPART){
gb[k,1:2]=g
}
Vi=V
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V

}
W[j]=f(g)

maxj=max(W)
minj=min(W)
var.global=iffelse((maxj-minj)==0,1,maxj-minj)
if(j>ITER){
maxj5=max(W[(j-5):j])
minj5=min(W[(j-5):j])

varj5=maxj5-minj5
}else{
varj5=var.global
}

```

```

vard=varj5/var.global
#print(list(X_Otimo=g,F_otimo=W[j],Iteracoes=j))
criterio.Parada=ifelse(j>ITER,ifelse((1/erro)*vard<1,T,F),F)
j=j+1

}
saida=list(X_Otimo=g,F_otimo=W[j-1],Iteracoes=j-1)
return(saida)
}
psol_min=function(AX1,fun,ITER,erro,NPART,copula){
options(digits = 22)
f=fun
X=AX1
Z=list(rep(0,NPART))
Y=rep(Z,10000)
j=1
W=c()

#g=X[which.max(X[[j]]),]
criterio.Parada=ifelse(pi<0,T,F)

while(sum(criterio.Parada)!=1 ){

#tratar restrição de espaço paramétrico específico

for(i in 1:NPART){

if(copula=="Frank"){
Par_Rest=X[i,1]==0.0
}else{
if(copula=="Clayton"){
Par_Rest=X[i,1]<=0.0
}else{
Par_Rest=X[i,1]<1.0
}
}

Y[[j]][i]=ifelse(Par_Rest,-Inf,f(as.vector(X[i,1:1])))

```

```

}

if(j==1){
p=X
g=X[which.min(abs(Y[[j]])),]
V=matrix(0,NPART,1)
Vi=V
gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V
# W[j]=f(g)
}else{
px=matrix(0,NPART,1)

#####
if(copula=="Frank"){
Par_Rest=X[,1]==0.0
}else{
if(copula=="Clayton"){
Par_Rest=X[,1]<0.0
}else{
Par_Rest=X[,1]<1.0
}
}

#####
#####
if(copula=="Frank"){
Par_Restx=p[,1]==0.0
}else{
if(copula=="Clayton"){
Par_Restx=p[,1]<0.0
}else{
Par_Restx=p[,1]<1.0
}
}
}

```

```

}
}

AA=Par_Rest+Par_Restx

for(i in 1:NPART){

if(iffelse(AA[i]<1,abs(f(p[i,]))-abs(f(X[i,])),0)<0){
px[i,]=p[i,]
}else{
px[i,]=X[i,]
}
}
p=px
gk=g
gx=X[which.min(abs(Y[[j]])),]
if(abs(f(gx))-abs(f(g))<0){g=gx}else{g=g}

gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
Vi=V
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V

}
W[j]=f(g)

maxj=max(W)
minj=min(W)
var.global=iffelse((maxj-minj)==0,1,maxj-minj)
if(j>ITER){
maxj5=max(W[(j-5):j])
minj5=min(W[(j-5):j])

varj5=maxj5-minj5
}else{

```

```

varj5=var.global
}

vard=varj5/var.global

print(list(X_Otimo=g,F_otimo=W[j],Iteracoes=j))
criterio.Parada=ifelse(j>ITER,ifelse((1/erro)*vard<1,T,F),F)
j=j+1

}
saida=list(X_Otimo=g,F_otimo=W[j-1],Iteracoes=j-1)
return(saida)
}

pso_max=function(AX, funt, ITER, erro, NPART, copula1, copula2) {
options(digits = 22)
f=funt
X=AX
Z=list(rep(0, NPART))
Y=rep(Z, 10000)
j=1
W=c()

#g=X[which.min(Y[[j]]),]
criterio.Parada=ifelse(pi<0,T,F)

while(sum(criterio.Parada)!=1 ){

for(i in 1:NPART){

if(copula1=="Frank") {
Par_Rest_1=X[i,1]==0.0
}else{
if(copula1=="Clayton") {
Par_Rest_1=X[i,1]<=0.0
}else{
Par_Rest_1=X[i,1]<1.0

```



```

}
}

if(copula2=="Frank"){
Par_Rest_2=X[i,1]==0.0
}else{
if(copula2=="Clayton"){
Par_Rest_2=X[i,1]<=0.0
}else{
Par_Rest_2=X[i,1]<1.0
}
}

Y[[j]][i]=ifelse((Par_Rest_1+Par_Rest_2)>0,-Inf,
f(as.vector(X[i,1:2])))

}

if(j==1){
p=X
g=X[which.max(Y[[j]]),]
V=matrix(0,NPART,2)
Vi=V
gb=matrix(g,NPART,2)
for(k in 1:NPART){
gb[k,1:2]=g
}
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V
# W[j]=f(g)
}else{
px=matrix(0,NPART,2)

#####

if(copula1=="Frank"){

```

```

Par_Rest_1=X[,1]==0.0
}else{
if(copula1=="Clayton"){
Par_Rest_1=X[,1]< -1.0
}else{
Par_Rest_1=X[,1]<1.0
}
}

```

```
#####
```

```

if(copula2=="Frank"){
Par_Rest_2=X[,1]==0.0
}else{
if(copula2=="Clayton"){
Par_Rest_2=X[,1]< -1.0
}else{
Par_Rest_2=X[,1]<1.0
}
}

```

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

```

if(copula1=="Frank"){
Par_Rest_1x=p[,1]==0.0
}else{
if(copula1=="Clayton"){
Par_Rest_1x=p[,1]< -1.0
}else{
Par_Rest_1x=p[,1]<1.0
}
}

```

```
#####
```

```

if(copula2=="Frank"){
Par_Rest_2x=p[,1]==0.0

```

```

}else{
if(copula2=="Clayton"){
Par_Rest_2x=p[,1]<=0.0
}else{
Par_Rest_2x=p[,1]<1.0
}
}

AA1=Par_Rest_1+Par_Rest_2
AA2=Par_Rest_1x+Par_Rest_2x
AA=AA1+AA2

for(i in 1:NPART){
if(iffelse(AA[i]<1,f(p[i,])-f(X[i,]),1)>0){
px[i,]=p[i,]
}else{
px[i,]=X[i,]
}
}
p=px
gk=g
gx=X[which.max(Y[[j]]),]
if(f(gx)-f(g)>0){g=gx}else{g=g}

gb=matrix(g,NPART,2)
for(k in 1:NPART){
gb[k,1:2]=g
}
Vi=V
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V

}
W[j]=f(g)

```

```

maxj=max(W)
minj=min(W)
var.global=ifelse((maxj-minj)==0,1,maxj-minj)
if(j>ITER){
maxj5=max(W[(j-5):j])
minj5=min(W[(j-5):j])

varj5=maxj5-minj5
}else{
varj5=var.global
}

vard=varj5/var.global
#print(list(X_Otimo=g,F_otimo=W[j],Iteracoes=j))
criterio.Parada=ifelse(j>ITER,ifelse((1/erro)*vard<1,T,F),F)
j=j+1

}
saida=list(X_Otimo=g,F_otimo=W[j-1],Iteracoes=j-1)
return(saida)
}
psol_max=function(AX, fun, ITER, erro, NPART, copula) {
options(digits = 22)
f=fun
X=AX
Z=list(rep(0,NPART))
Y=rep(Z,10000)
j=1
W=c()

#g=X[which.max(X[[j]]),]
criterio.Parada=ifelse(pi<0,T,F)

while(sum(criterio.Parada)!=1 ){

for(i in 1:NPART){

```

```

if(copula=="Frank"){
Par_Rest=X[i,1]==0.0
}else{
if(copula=="Clayton"){
Par_Rest=X[i,1]< -1.0
}else{
Par_Rest=X[i,1]<1.0
}
}

Y[[j]][i]=ifelse(Par_Rest,-Inf,f(as.vector(X[i,1:1])))

}

if(j==1){
p=X
g=X[which.max(Y[[j]])],]
V=matrix(0,NPART,1)
Vi=V
gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
V=0.73*(0.3*Vi+runif(1)*2.05*(p-X)+runif(1)*2.05*(gb-X))
Xi=X
X=Xi+V
# W[j]=f(g)
}else{
px=matrix(0,NPART,1)

#####
if(copula=="Frank"){
Par_Rest=X[,1]==0.0
}else{
if(copula=="Clayton"){
Par_Rest=X[,1]< -1.0
}else{
Par_Rest=X[,1]<1.0
}
}
}

```

```

}
}

#####
#####
if(copula=="Frank"){
Par_Restx=p[,1]==0.0
}else{
if(copula=="Clayton"){
Par_Restx=p[,1]< -1.0
}else{
Par_Restx=p[,1]<1.0
}
}

AA=Par_Rest+Par_Restx

for(i in 1:NPART){

if(iffelse(AA[i]<1,f(p[i,])-f(X[i,]),0)>0){
px[i,]=p[i,]
}else{
px[i,]=X[i,]
}
}
p=px
gk=g
gx=X[which.max(Y[[j]]),]
if(f(gx)-f(g)>0){g=gx}else{g=g}

gb=matrix(g,NPART,1)
for(k in 1:NPART){
gb[k,1:1]=g
}
Vi=V
V=0.73*(0.3*Vi+runiform(1)*2.05*(p-X)+runiform(1)*2.05*(gb-X))
Xi=X
X=Xi+V

```

```

}
W[j]=f(g)

maxj=max(W)
minj=min(W)
var.global=ifelse((maxj-minj)==0,1,maxj-minj)
if(j>ITER){
maxj5=max(W[(j-5):j])
minj5=min(W[(j-5):j])

varj5=maxj5-minj5
}else{
varj5=var.global
}

vard=varj5/var.global

#print(list(X_Otimo=g,F_otimo=W[j],Iteracoes=j))
criterio.Parada=ifelse(j>ITER,ifelse((1/erro)*vard<1,T,F),F)
j=j+1

}
saida=list(X_Otimo=g,F_otimo=W[j-1],Iteracoes=j-1)
return(saida)
}

gera_vet_aleatoriol=function(a,b,n){

x1=(b-a)*runif(n)+a

#x2=(b-a)*runif(n)+a

matrix(c(x1),n,1)
}

```

```

EM=function (Erro, Npart, p0, theta1, theta2, copula1, copula2) {

copula1=copula1
copula2=copula2
p0=p0
theta1_0=theta1
theta2_0=theta2
tol = Erro
NPART = Npart
cond=0
j=1
L=c ()
L[0]=0

pk=c ()
pj=c ()
pj[0]=0

if (1==1) {

if (copula1=="Gumbel") {
cop1=dgumbel
Dcop1=Ddgumbel
A=1
A2=10
} else {
if (copula1=="Clayton") {
cop1=dclayton
Dcop1=Ddclayton
A=0
A2=10
} else {
if (copula1=="Frank") {
cop1=dFrank
Dcop1=DdFrank
A=-10
A2=10
} else {

```



```

if(copula1=="Joe"){
cop1=dJoe
Dcop1=DdJoe
A=1
A2=10
}else{
"Erro, nome da cópula está errado. Tente nomes
nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
}
}
}
}

```

```

if(copula2=="Gumbel"){
cop2=dgumbel
Dcop2=Ddgumbel
B=1
B2=10
}else{
if(copula2=="Clayton"){
cop2=dclayton
Dcop2=Ddclayton
B=0
B2=10
}else{
if(copula2=="Frank"){
cop2=dFrank
Dcop2=DdFrank
B=-10
B2=10
}else{
if(copula2=="Joe"){
cop2=dJoe
Dcop2=DdJoe
B=1
B2=10
}else{
"Erro, nome da cópula está errado. Tente nomes

```

```

    nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
  }
}
}
}

}

while (cond < 1) {

# E-STEP

(w1=p0*cop1(x,y,theta1_0,Fx,Gy,fx,gy) / (p0*cop1(x,y
,theta1_0,Fx,Gy,fx,gy) + (1-p0)*cop2(x,y,
theta2_0,Fx,Gy,fx,gy)))

# M-Estp

f1=function(a) {

fx1=log(cop1(x,y,a,Fx,Gy,fx,gy))
D1=sum(w1*fx1)
D1
}
f2=function(b) {

fx2=log(cop2(x,y,b,Fx,Gy,fx,gy))
D2=sum((1-w1)*fx2)
D2
}

#f=function(vet) {
#a=vet[1]
#b=vet[2]
# fx1=Dcop1(a) / (p0*cop1(x,y,a,Fx,Gy,fx,gy)

```

```

+(1-p0)*cop2(x,y,b,Fx,Gy,fx,gy)
#fx2=Dcop2(b)/(p0*cop1(x,y,a,Fx,Gy,fx,gy)
+(1-p0)*cop2(x,y,b,Fx,Gy,fx,gy)
#D1=sum(w1*fx1)
#D2=sum((1-w1)*fx2)
#D1+D2
#}

AX1 = gera_vet_aleatorio1(A,A2,NPART)
AX2 = gera_vet_aleatorio1(B,B2,NPART)
#(AX = rbind(AX,c(theta1_0,theta2_0)))
(AX1 = rbind(AX1,theta1_0))
(AX2 = rbind(AX2,theta2_0))

NPART=NPART+1

#(psx=pso_min(AX,f,5,10^(-4),NPART,copula1,copula2))

(psx1=psol_max(AX1,f1,9,10^(-4),NPART,copula1))
(psx2=psol_max(AX2,f2,9,10^(-4),NPART,copula2))

pk[j]=mean(w1)
p=mean(w1)

theta1 = theta1_0
theta2 = theta2_0
theta1_0 = psx1$X_Otimo
theta2_0 = psx2$X_Otimo

Lx=function(vet,px){
a=vet[1]
b=vet[2]
p=px
sum(p*cop1(x,y,a,Fx,Gy,fx,gy)+(1-p)*cop2(x,y,b,Fx,Gy,fx,gy))
}

```

```

L[j]=Lx(c(theta1_0,theta2_0),p)

cond = ifelse(j>5,ifelse(abs(L[j]-L[j-1]) < tol,1,0),0)
pj[j]=mean(pk)

cond2=ifelse((j>10),ifelse(abs(pj[j]-pj[j-5])<0.0001,1,0),0)
if(sum(cond2)>0){
cond=1
}

print(list(theta=c(theta1_0,theta2_0),
L_otimo=L[j],p=p,Iteracoes=j))
j=j+1
p0=p
}

SAIDA=list(phi_Otimo=c(p=p,theta1=theta1_0,
theta2=theta2_0),L_otimo=L[j-1],Iteracoes=j-1,
pk=pk,mediap=mean(pk),varp=var(pk))
SAIDA
}

cor(x,y, method = "kendall")

hist(x,col=2)
hist(y,col=4)

(xamostra=EM(10^(-4),40,0.6,-4,1.5,"Frank","Gumbel"))

phi=c(phi1=round(modx),phi2=round(mody))
parametros_amostra=
mx=data.frame(x,y)
row.names(mx)=NULL

```

```

FINAL=list()

## Bootstrap
for (k in 1:100) {
mb=mx[sample(1:1000,1000,replace = T),]
x=mb$x
y=mb$y

modx = fevd(x,type = "GEV",method = "MLE",period.basis = "year")

round(modx$results$par,2)
plot(modx, "qq",main="")

#Acumudala
FModEstx=function(t) {
a=as.numeric(modx$results$par[3])
s=as.numeric(modx$results$par[2])
u=as.numeric(modx$results$par[1])

exp(-(1+(a/s)*(t-u))^-1/a))

}

densfx=function(t) {
a=as.numeric(modx$results$par[3])
s=as.numeric(modx$results$par[2])
u=as.numeric(modx$results$par[1])
devd(t, loc = u, scale = s, shape = a,
type = "GEV")

}

mody = fevd(y,type = "GEV",method = "MLE",period.basis = "year")

round(mody$results$par,2)
plot(mody, "qq",main="")

#Acumudala
FModEsty=function(t) {

```

```

a=as.numeric(mody$results$par[3])
s=as.numeric(mody$results$par[2])
u=as.numeric(mody$results$par[1])

exp(-(1+(a/s)*(t-u))^-1/a)

}

densfy=function(t){
a=as.numeric(mody$results$par[3])
s=as.numeric(mody$results$par[2])
u=as.numeric(mody$results$par[1])
devd(t, loc = u, scale = s, shape = a,
type = "GEV")
}

dgumbel=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*(1/(u*v))*(log(u)*log(v))^(theta-1)
)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta)
)*(theta-1)*((-log(u))^theta+(-log(v))^theta)
)^(1/theta-2)+((-log(u))^theta+(-log(v))^theta)^(2/theta-2))
}

dclayton=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*u^(-theta+1)*v^(-theta+1)*(theta+
1)*ifelse((u^(-theta)+v^(-theta)-1)<=0.0,0.00,(u^(-theta)
+v^(-theta)-1)^(-(1/theta+2)))
}

dFrank=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*(-theta)*exp(-theta*u)*exp(-theta*v

```

```

) * ((exp(-theta)-1) / (exp(-theta)-1+
(exp(-theta*u)-1) * (exp(-theta*v)-1)) ^2)
}

dJoe=function(x,y,theta,Fx,Gy,fx,gy) {
u=Fx(x)
v=Gy(y)

H=((1-u)^theta + (1-v)^theta - ((1-u)^theta
) * ((1-v)^theta))^(1/theta-1)
H1=(theta-1) * ((1-u)^theta + (1-v
)^theta - ((1-u)^theta) * ((1-v
)^theta))^(1/theta-2) * ((1-u)^(theta-
1)-(1-u)^(theta-1) * (1-v)^theta)
G=(1-v)^(theta-1) - (1-v)^(theta-1) * (1-u)^theta
G1=theta * (1-v)^(theta-1) * (1-u)^(theta-1)
fx(x) * gy(y) * (H1 * G + H * G1)
}

```

```

x=x
y=y
fx=densfx
gy=densfy
Fx=FModEstx
Gy=FModEsty

```

```

Ddclayton=function(theta) {
u=Fx(x)
v=Gy(y)
h1=(u*v)^(-(theta+1)) * (theta+1)
h11=(u*v)^(-(theta+1)) * (1-(theta+1) * log(u*v))
h2=ifelse((u^(-theta)+v^(-theta)-1)<=0.0,0.00,
(u^(-theta)+v^(-theta)-1)^(-(1/theta+2)))
h21=ifelse((u^(-theta)+v^(-theta)-1)<0.0,0.00,
h2 * ((1/theta^2) * log(u^(-theta)+v^(-theta)-1)
+ (1/theta+2) * (1/(u^(-theta)+v^(-theta)-1)) * (
u^(-theta) * log(u)+v^(-theta) * log(v))))
fx(x) * gy(y) * (h11 * h2 + h1 * h21)
}

```

```
}
```

```
Ddgumbel=function(theta) {
```

```
u=Fx(x)
```

```
v=Gy(y)
```

```
B_0=(-log(u))^theta+(-log(v))^theta
```

```
B_0L=log((-log(u))*(-log(u))^theta+log(
```

```
(-log(v)))*(-log(v))^theta
```

```
J_0=B_0^(1/theta)
```

```
J_0L=- (1/theta)*J_0*((1/theta)*log(B_0)-(B_0L/B_0))
```

```
H1_0=B_0^(2/theta-2)
```

```
H1_0L=H1_0*(-2*(1/theta^2)*log(B_0)+(2/theta-2)*(B_0L/B_0))
```

```
H2_0=(theta-1)*B_0^(1/theta-2)
```

```
H2_0L=B_0^(1/theta-2)+(theta-1)*B_0^(1/theta-2)* (
```

```
-(1/theta^2)*log(B_0)+(1/theta-2)*(B_0L/B_0))
```

```
F_0=(log(u)*log(v))^(theta-1)
```

```
F_0L=F_0*log(log(u)*log(v))
```

```
G_0=exp(-J_0)
```

```
G_0L=G_0*(-J_0L)
```

```
H_0=H1_0+H2_0
```

```
H_0L=H1_0L+H2_0L
```

```
fx(x)*gy(y)*(1/(u*v))*((F_0L*G_0+F_0*G_0L)*H_0+(F_0*G_0)*H_0L)
```

```
}
```

```
DdFrank=function(theta) {
```

```
u=Fx(x)
```

```
v=Gy(y)
```

```
f=-theta*exp(-theta*u)
```

```
f2=exp(-theta*v)
```

```
f1=exp(-theta*u)*(theta*u-1)
```

```
f21=-v*exp(-theta*v)
```

```
g1=f*f2
```

```
g11=f1*f2+f*f21
```

```
g2=exp(-theta)-1
```

```
g21=-exp(-theta)
```



```

h1=g1*g2
h11=g11*g2+g1*g21
gx1=exp(-theta)-1
gx11=-exp(-theta)
gx2=exp(-theta*u)-1
gx21=-u*exp(-theta*u)
gx3=exp(-theta*v)-1
gx31=-v*exp(-theta*v)
gx=gx1+gx2*gx3
gx1=gx11+gx21*gx3+gx2*gx31
h2=gx^2
h21=2*gx*gx1
fx(x)*gy(y)*(h11*h2-h1*h21)/h2^2
}

```

```

DdJoe=function(theta) {
u=Fx(x)
v=Gy(y)

jx=1-u
jy=1-v

j1=jx^theta
j11=j1*log(jx)

j2=jy^theta
j21=j2*log(jy)

j1x=jx^(theta-1)
j1x1=j1x*log(jx)

j2y=jy^(theta-1)
j2y1=j2y*log(jy)

g=j1+j2-j1*j2
g1=j11+j21-(j11*j2+j1*j21)

h1=(theta-1)*g^(1/theta-2)

```

```

h11=g^(1/theta-2)+(theta-1)*(g^(1/theta-2))*(
-(1/theta^2)*log(g)+(1/theta-2)*(g1/g))

h2=j2y-j2y*j1
h21=j2y1-(j2y1*j1+j2y*j11)

H1=h1*h2
H1L=h11*h2+h1*h21

H2=j1x-j1x*j2
H2L=j1x1-(j1x1*j2+j1x*j21)

H3=g^(1/theta-1)
H3L=(g^(1/theta-1))*(-(1/theta^2)*log(g)+(1/theta-1)*(g1/g))

H4=theta*j1x*j2y
H4L=j1x*j2y+theta*j1x1*j2y+theta*j1x*j2y1

fx(x)*gy(y)*(H1L*H2+H1*H2L+H3L*H4+H3*H4L)
}

EM=function(Erro,Npart,p0,theta1,theta2,copula1,copula2){

copula1=copula1
copula2=copula2
p0=p0
theta1_0=theta1
theta2_0=theta2
tol = Erro
NPART = Npart
cond=0
j=1
L=c()
L[0]=0

pk=c()
pj=c()
pj[0]=0

```

```

if(1==1) {

if(copula1=="Gumbel") {
cop1=dgumbel
Dcop1=Ddgumbel
A=1
A2=10
}else{
if(copula1=="Clayton") {
cop1=dclayton
Dcop1=Ddclayton
A=0
A2=10
}else{
if(copula1=="Frank") {
cop1=dFrank
Dcop1=DdFrank
A=-10
A2=10
}else{
if(copula1=="Joe") {
cop1=dJoe
Dcop1=DdJoe
A=1
A2=10
}else{
"Erro, nome da cópula está errado. Tente nomes
nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
}
}
}
}

if(copula2=="Gumbel") {
cop2=dgumbel
Dcop2=Ddgumbel
B=1
B2=10
}

```

```

}else{
if(copula2=="Clayton"){
cop2=dclayton
Dcop2=Ddclayton
B=0
B2=10
}else{
if(copula2=="Frank"){
cop2=dFrank
Dcop2=DdFrank
B=-10
B2=10
}else{
if(copula2=="Joe"){
cop2=dJoe
Dcop2=DdJoe
B=1
B2=10
}else{
"Erro, nome da cópula está errado.
Tente nomes nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
}
}
}
}

}

while (cond < 1) {

# E-STEP

(w1=p0*cop1(x,y,theta1_0,Fx,Gy,fx,gy)/(p0*cop1(x,y,
theta1_0,Fx,Gy,fx,gy)+(1-p0)*cop2(x,y,theta2_0,Fx,Gy,fx,gy)))

```

```

# M-Estp

f1=function(a) {

fx1=log(cop1(x,y,a,Fx,Gy,fx,gy))
D1=sum(w1*fx1)
D1
}

f2=function(b) {

fx2=log(cop2(x,y,b,Fx,Gy,fx,gy))
D2=sum((1-w1)*fx2)
D2
}

AX1 = gera_vet_aleatorio1(A,A2,NPART)
AX2 = gera_vet_aleatorio1(B,B2,NPART)
#(AX = rbind(AX,c(theta1_0,theta2_0)))
(AX1 = rbind(AX1,theta1_0))
(AX2 = rbind(AX2,theta2_0))

NPART=NPART+1

#(psx=pso_min(AX,f,5,10^(-4),NPART,copula1,copula2))

(psx1=pso1_max(AX1,f1,9,10^(-4),NPART,copula1))
(psx2=pso1_max(AX2,f2,9,10^(-4),NPART,copula2))

pk[j]=mean(w1)
p=mean(w1)

theta1 = theta1_0
theta2 = theta2_0
theta1_0 = psx1$X_Otimo
theta2_0 = psx2$X_Otimo

Lx=function(vet,px) {

```

```

a=vet [1]
b=vet [2]
p=px
sum(p*cop1(x,y,a,Fx,Gy,fx,gy)+(1-p)*cop2(x,y,b,Fx,Gy,fx,gy))
}

L[j]=Lx(c(theta1_0,theta2_0),p)

cond = ifelse(j>5,ifelse(abs(L[j]-L[j-1]) < tol,1,0),0)
pj[j]=mean(pk)

cond2=ifelse((j>10),ifelse(abs(pj[j]-pj[j-5])<0.0001,1,0),0)
if(sum(cond2)>0){
cond=1
}

# print(list(theta=c(theta1_0,theta2_0),L_otimo=L[j],p=p,Iteracoes=j))
j=j+1
p0=p
}

SAIDA=list(phi_Otimo=c(p=p,theta1=theta1_0,
theta2=theta2_0),L_otimo=L[j-1],Iteracoes=j-1,
pk=pk,mediap=mean(pk),varp=var(pk))
SAIDA
}
emap=EM(10^(-4),40,0.5,-4.0,0.5,"Frank","Gumbel")
(phi1=as.numeric(round(modx$results$par,2)))
(phi2=as.numeric(round(mody$results$par,2)))
(thetacop_p=as.numeric(round(emap$phi_Otimo,2)))

vet_fim=c(phi1,phi2,thetacop_p)
FINAL[[k]]=vet_fim
print(k)
print(FINAL[k])
}

#save.image(file = "mistura.RData")

```

```

FINAL[1]

PHI=matrix(FINAL[[1]],1,9)
for( j in 2:100){
PHI=rbind(PHI,FINAL[[j]])
}

colMeans(PHI)

varPHI=c()
for( i in 1:9){
varPHI[i]=var(PHI[,i])
}

sqrt(varPHI)

save.image(file = "Simulacaoclayton.RData")

#####
##### Plota curva de nível
#####

library(lattice)

dgumbel=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*(1/(u*v))*(log(u)*log(v))^(theta-1)
)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta)
)*(theta-1)*((-log(u))^theta+(-log(v))^theta)
)^(1/theta-2)+((-log(u))^theta+(-log(v))^theta)^(2/theta-2)
}

dclayton=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*u^(-(theta+1))*v^(-(theta+1))*(theta+1)

```

```

) *ifelse((u^(-theta)+v^(-theta)-1)<=0.0,0.00,(
u^(-theta)+v^(-theta)-1)^(-(1/theta+2)))
}

dFrank=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)
fx(x)*gy(y)*(-theta)*exp(-theta*u)*exp(-theta*v)
)*((exp(-theta)-1)/(exp(-theta)-1+
(exp(-theta*u)-1)*(exp(-theta*v)-1))^2)
}

dJoe=function(x,y,theta,Fx,Gy,fx,gy){
u=Fx(x)
v=Gy(y)

H=((1-u)^theta+(1-v)^theta-((1-u)^theta
)*(1-v)^theta)^(1/theta-1)
H1=(theta-1)*((1-u)^theta+(1-v)
)^theta-((1-u)^theta)*(1-v)^theta
)^(1/theta-2)*((1-u)^(theta
-1)-(1-u)^(theta-1)*(1-v)^theta)
G=(1-v)^(theta-1)-(1-v)^(theta-1)*(1-u)^theta
G1=theta*(1-v)^(theta-1)*(1-u)^(theta-1)
fx(x)*gy(y)*(H1*G+H*G1)
}

parametros_amostra
vet=as.numeric(parametros_amostra)

p=v[5]
theta1=v[6]
theta2=v[7]
u1=v[1]

```



```

u2=vet [3]

s1=vet [2]
s2=vet [4]

f1x = function(x) (1/s1)*exp(-(x-u1)/s1)*exp(-exp(-(x-u1)/s1))
g1y = function(y) (1/s2)*exp(-(y-u2)/s2)*exp(-exp(-(y-u2)/s2))

f2x = function(x) (1/s1)*exp(-(x-u1)/s1)*exp(-exp(-(x-u1)/s1))
g2y = function(y) (1/s2)*exp(-(y-u2)/s2)*exp(-exp(-(y-u2)/s2))

F1x = function(x) exp(-exp(-(x-u1)/s1))
G1y = function(y) exp(-exp(-(y-u2)/s2))
F2x = function(x) exp(-exp(-(x-u1)/s1))
G2y = function(y) exp(-exp(-(y-u2)/s2))

misturax=function(p,q1,q2,copula1,copula2,theta1,theta2)
{
if(copula1=="Gumbel"){
cop1=dgumbel

}else{
if(copula1=="Clayton"){
cop1=dclayton

}else{
if(copula1=="Frank"){
cop1=dFrank

}else{
if(copula1=="Joe"){
cop1=dJoe

}else{
"Erro, nome da cópula está errado. Tente nomes
nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."
}
}
}
}
}

```

```
}  
}  
}
```

```
if (copula2=="Gumbel") {  
cop2=dgumbel
```

```
}else{  
if (copula2=="Clayton") {  
cop2=dclayton
```

```
}else{  
if (copula2=="Frank") {  
cop2=dFrank
```

```
}else{  
if (copula2=="Joe") {  
cop2=dJoe
```

```
}else{  
"Erro, nome da cópula está errado. Tente nomes  
nesta lista : c("Clayton", "Gumbel", "Frank", "Joe")."  
}  
}  
}  
}
```

```
Copula1 = cop1(q1,q2,theta1,F1x,G1y,f1x,g1y)  
Copula2 = cop2(q1,q2,theta2,F2x,G2y,f2x,g2y)
```

```
Mix=p*Copula1+(1-p)*Copula2  
return (Mix)  
}
```

```
x=mx$x
```

```

y=mx$y

n=100
xMin=min(x)-1
xMax=max(x)+4
yMin=min(y)-1
yMax=max(y)+4
x1=seq(xMin,xMax,length=n)
y1=seq(yMin,yMax,length=n)
z=matrix(nrow=n,ncol=n)

#Numero d fotos

zpx=list()
for(k in 1:(length(p))){

for(i in 1:n)
{
for(j in 1:n)
{
z[i,j]=misturax(p,x1[i],y1[j],"Frank","Gumbel",theta1 ,theta2)
}
}
#Gerar grafico de contornos
zpx[[k]]=z
# contour(x=x1,y=y1,z,nlevels=10)
# print(k)
}

zpx1=list()
for(k in 1:(length(p))){

for(i in 1:n)
{
for(j in 1:n)
{
z[i,j]=misturax(p,x1[i],y1[j],"Frank","Gumbel",theta1 ,theta2)
}
}
}

```

```

#Gerar grafico de contornos
zpx1[[k]]=z
# contour(x=x1,y=y1,z,nlevels=10)
# print(k)
}

ind=rep(c(1,2),c(301,699))
M=matrix(c(ind,x,y),nrow = 1000, ncol = 3)

contour(x=x1,y=y1,zpx[[1]],nlevels=272,col=2, lwd=1,axes = TRUE,
xlim = range(x1, finite = F),
ylim = range(y1, finite = F))

par(new=T)
plot(M[ind==1,2],M[ind==1,3], col=1,
xlim = range(x1),ylim = range(y1), pch=16,
xlab = "Tempo até próxima erupção (minutos)",
ylab="Duração da erupção (minutos)")

par(new=T)
plot(M[ind==2,2],M[ind==2,3], col=4,
xlim = range(x1),ylim = range(y1), pch=16,
xlab = "Tempo até próxima erupção (minutos)",
ylab="Duração da erupção (minutos)")
par(new=T)
plot(x,y, pch=16, col="green", xlim = range(x1),ylim = range(y1))

legend(40,6.0,legend = c("Cluster 1","Cluster 2"),
col = c(1,4),cex = 0.8,pch = c(16,16))

```