UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

SOME NEW FAMILIES OF CONTINUOUS DISTRIBUTIONS

PEDRO RAFAEL DINIZ MARINHO

Tese de Doutorado

Recife
2016

PEDRO RAFAEL DINIZ MARINHO

SOME NEW FAMILIES OF CONTINUOS DISTRIBUTIONS

Doctoral thesis submitted to the Graduate Program in Statistics, Department of Statistics, Federal University of Pernambuco as a partial requirement for obtaining a Ph.D in Statistics.

Advisor: Professor Dr. Gauss Moutinho Cordeiro

Recife
2016

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M338s Marinho, Pedro Rafael Diniz
Some new families of continuous distributions / Pedro Rafael Diniz Marinho. – 2016.
113 f.: il., fig., tab.

Orientador: Gauss Moutinho Cordeiro.
Tese (Doutorado) – Universidade Federal de Pernambuco. CCEN. Estatística, Recife, 2016.

Inclui referências a apêndices.

1. Probabilidades. 2. Distribuição de probabilidades. I. Cordeiro, Gauss Moutinho (orientador). II. Título.

519.2 CDD (23. ed.) UFPE- MEI 2016-086

PEDRO RAFAEL DINIZ MARINHO

SOME NEW FAMILIES OF CONTINUOS DISTRIBUTIONS

Tese apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Estatística.

Aprovada em: 27 de junho de 2016.

BANCA EXAMINADORA

Prof. Dr. Gauss Moutinho Cordeiro
UFPE

Prof. Dr. Abraão David Costa do Nascimento
UFPE

Prof[a]. Dr. Audrey Helen Mariz de Aquino Cysneiros
UFPE

Prof. Dr. Cláudio Tadeu Cristino
UFRPE

Prof[a]. Dr. Renata Maria Cardoso Rodrigues de Souza
UFPE/CIn

Resumo

A área de análise de sobrevivência é importante na Estatística e é comumente aplicada às ciências biológicas, engenharias, ciências sociais, entre outras. Tipicamente, o tempo de vida ou falha pode ter diferentes interpretações dependendo da área de aplicação. Por exemplo, o tempo de vida pode significar a própria vida de uma pessoa, o tempo de funcionamento de um equipamento até sua falha, o tempo de sobrevivência de um paciente com uma doença grave desde o diagnóstico, a duração de um evento social como um casamento, entre outros significados. O tempo de vida é uma variável aleatória não negativa, que pode ter a função de risco na forma constante, monótona crescente, monótona decrescente ou não monótona (por exemplo, em forma de U).

Nas últimas décadas, várias famílias de modelos probabilísticos têm sido propostas. Esses modelos podem ser construídos com base em alguma transformação de uma distribuição padrão, geralmente já conhecida na literatura. Uma dada combinação linear ou mistura de modelos G normalmente define uma classe de modelos probabilísticos tendo G como caso especial.

Esta tese é composta de capítulos independentes. O primeiro e último são curtos capítulos que incluem a introdução e as conclusões do estudo desenvolvido. Duas famílias de distribuições, denominadas de classe "exponentiated logarithmic generated" (ELG) e a classe "geometric Nadarajah-Haghighi" (NHG) são estudadas. A última é uma composição das distribuições de Nadarajah-Haghighi e geométrica. Além disso, desenvolvemos uma biblioteca estatística para a linguagem de programação `R` chamada `AdequacyModel`. Esta é uma melhoria do pacote que foi disponibilizado no CRAN (Comprehensive `R` Archive Network) e está atualmente na versão 2.0.0. As duas principais funções da biblioteca são as funções `goodness.fit` e `pso`. A primeira função permite obter as estimativas de máxima verossimilhança (EMVs) dos parâmetros de um modelo e algumas medidas de bondade de ajuste dos modelos probabilísticos ajustados. É possível escolher o método de otimização para maximizar a função de log-verossimilhança. A segunda função apresenta o método meta-heurístico de busca global conhecido como Particle Swarm Optimization (PSO) proposto por Eberhart e Kennedy (1995). Algumas metodologias podem ser utilizadas para obtenção das EMVs necessárias para o cálculo de algumas medidas de adequação dos modelos probablísticos ajustados.

Palavras chaves: `AdequacyModel`. distribuição. mistura linear. PSO.

Abstract

The area of survival analysis is important in Statistics and it is commonly applied in biological sciences, engineering, social sciences, among others. Typically, the time of life or failure can have different interpretations depending on the area of application. For example, the lifetime may mean the life itself of a person, the operating time of equipment until its failure, the time of survival of a patient with a severe disease from the diagnosis, the duration of a social event as a marriage, among other meanings. The time of life or survival time is a positive continuous random variable, which can have constant, monotonic increasing, monotonic decreasing or non-monotonic (for example, in the form of a U) hazard function.

In the last decades, several families of probabilistic models have been proposed. These models can be constructed based on some transformation of a parent distribution, commonly already known in the literature. A given linear combination or mixture of G models usually defines a class of probabilistic models having G as a special case.

This thesis is composed of independent chapters. The first and last chapters are short chapters that include the introduction and conclusions of the study developed. Two families of distributions, namely the exponentiated logarithmic generated (ELG) class and the geometric Nadarajah-Haghighi (NHG) class are studied. The last one is a composition of the Nadarajah-Haghighi and geometric distributions. Further, we develop a statistical library for the `R` programming language called the `AdequacyModel`. This is an improvement of the package that was available on CRAN (Comprehensive `R` Archive Network) and it is currently in version 2.0.0. The two main functions of the library are the `goodness.fit` and `pso` functions. The first function allows to obtain the maximum likelihood estimates (MLEs) of the model parameters and some goodness-of-fit of the fitted probabilistic models. It is possible to choose the method of optimization for maximizing the log-likelihood function. The second function presents the method meta-heuristics global search known as particle swarm optimization (PSO) proposed by Eberhart and Kennedy (1995). Such methodology can be used for obtaining the MLEs necessary for the calculation of some measures of adequacy of the probabilistic models.

Keywords: `AdequacyModel` package. distribution. linear mixture. PSO.

List of Figures

2.1 The NHG density function for some parameter values; $\lambda = 1$. 17
2.2 The NHG hrf for some parameter values; $\lambda = 1$. 20
2.3 Points of minimum obtained by `optim` and `pso` functions the using methods BFGS and PSO. 24
2.4 Errors evaluated by bootstrap the maximum likelihood using the PSO method with 500 bootstrap replicates. 30
2.5 The TTT plot for the times between failures (thousands of hours) of secondary reactor pumps. 32
2.6 Gaussian kernel density estimation for the data of the time between failures (thousands of hours) of secondary reactor pumps. 32
2.7 Estimated for densities fitted to the times between failures (thousands of hours) of secondary reactor pumps. 34

3.1 Sums of the constants of the linear combination in equation (3.10). 47
3.2 The ELG-normal pdf for $\mu = 1$, $\sigma = 1$ and some values of a, b and c. 48
3.3 The ELG-Weibull pdf for $\alpha = 0.5$, $\beta = 2.0$ and some values of a, b and c. 49
3.4 The ELG-Weibull hrf for $\alpha = 0.5$, $\beta = 2.0$ and some values of a, b and c. 49
3.5 The ELG-gamma pdf for $\alpha = 1.0$, $\beta = 2.0$ and some values of a, b and c. 50
3.6 The ELG-Gamma hrf for the parameters $\alpha = 1.0$, $\beta = 2.0$ and some values of a, b and c. 50
3.7 The ELGLL pdf for $\alpha = 2.0$, $\beta = 2.0$ and some values of a, b and c. 51
3.8 The ELGLL hrf for $\alpha = 2.0$, $\beta = 2.0$ and some values of a, b and c. 51
3.9 Errors evaluated by bootstrap and the MLEs obtained by the PSO method with 500 bootstrap replicates. 57
3.10 The TTT plots for the: (a) glass fibres and (b) percentage of body fat. 59
3.11 The Gaussian kernel density estimation for: (a) the glass fibres and (b) percentage of body fat. 59
3.12 Estimates of the density functions for the: (a) glass fibres (b) percentages of body fat. 61
3.13 Estimates of the distribution functions and empirical distribution for the: (a) glass fibre and (b) percentages of body fat. 62
3.14 Kaplan-Meier (K-M) estimates compared with the ELG-W survival estimates for the: (a) glass fibres and (b) percentages of body fat. 62

4.1 The TIHL-N densities for (a) $\mu = 2.5$ and $\sigma = 3.4$ and for (b) $\mu = 2.5$. 70
4.2 The TIHL-Ga densities for (a) $\beta_2 = 1.1$ and for (b) $\beta_1 = 0.5$. 71
4.3 The hrf of the TIHL-Ga model. 71
4.4 The TIHL-Fr densities for (a) $\alpha = 1.5$ and for (b) $\beta = 0.5$. 72
4.5 The hrf of the TIHL-Fr model. 73
4.6 The TTT plot for: (a) the glass fibres data (b) and percentage of body fat. 78

4.7 Gaussian kernel density estimation for: (a) the glass fibres data and (b) for percentage of body fat. . . . 78
4.8 Estimates of the density functions for the: (a) glass fibres data and (b) percentage of body fat. . . . 81
4.9 Estimates of the distribution function and empirical distribution for the: (a) glass fibre data and (b) percentage of body fat data. . . . 81
4.10 Kaplan-Meier estimates (K-M estimates) compared with the TIHL-W survival estimates for the: (a) glass fibre data and (b) percentage of body fat data. The confidence intervals are considered 95%. . . . 82

5.1 Function $f(\theta) = 6 + \theta^2 \sin(14\theta)$ with global minimum estimates. . . . 93
5.2 Easom function at two different angles. . . . 95
5.3 Curves of levels of the Easom function. . . . 97
5.4 Cross-in-tray function at two different angles. . . . 98
5.5 Curves of levels of Cross-in-tray function. . . . 99
5.6 Hölder function at two different angles. . . . 100
5.7 Curves of levels of Hölder function. . . . 101
5.8 Fitted Exp-Weibull density to stress data (until fracture) of carbon fibers in Gba. . . . 106
5.9 TTT-plot for carbon data. . . . 108
5.10 Numbers of downloads from `AdequacyModel` package by RStudio repository. . . . 109
5.11 Numbers of downloads from `AdequacyModel` package by countries by RStudio repository. 109

List of Tables

2.1 Times of executions of Monte Carlo simulations to evaluate the interval estimates obtained by single and double percentile bootstrap. 29
2.2 Coverage and range of interval estimates by simple bootstrap, double and nominal level of 95%. 29
2.3 Times between failures (thousands of hours) of secondary reactor pumps. 30
2.4 Descriptive statistics. 31
2.5 The MLEs of the parameters for the NHG, ENH, NH and EXP models through of the PSO method (standard errors bootstrap in parentheses). 34
2.6 Interval estimates by double percentile bootstrap for the model parameters at a confidence level of 95%. 34
2.7 Goodness-of-fit statistics. 35

3.1 Percentage of coverage and average amplitude of interval estimates for the parameters added by the generator by bootstrap percentile at a nominal level of 95%. 57
3.2 Descriptive statistics. 58
3.3 MLEs for the glass fibres data denoted by (I) and percentage of body fat data denoted by (II). 60
3.4 Goodness-of-fit statistics for the glass fibres data denoted by (I) and percentages of body fat data denoted by (II). 61

4.1 Descriptive statistics. 77
4.2 MLEs for the of glass fiber data (standard errors in parentheses). 80
4.3 MLEs for the percentage of body fat data (standard errors in parentheses). 80
4.4 Goodness-of-fit statistics for the fits to the glass fibre data denoted by (I) and percentage of body fat data denoted by (II). 82

5.1 Useful quantities for some PS distributions. 89

Contents

1 Introduction **11**

References **12**

2 A new class of lifetime models and the evaluation of the confidence intervals by double percentile bootstrap **14**
2.1 Introduction . . . 14
2.2 Construction of the NHG distribution . . . 16
2.3 Maximum likelihood estimation . . . 21
2.4 Optimization algorithm . . . 21
2.5 Bootstrap confidence intervals . . . 24
2.5.1 Bootstrap percentile interval . . . 25
2.5.2 Interval Double Bootstrap Percentile . . . 26
2.6 Simulations and hardware used . . . 27
2.7 Application . . . 30
2.8 Concluding remarks . . . 35

Appendix A Code in R language for the PSO method **36**

References **39**

3 The exponentiated logarithmic generated family of distributions **43**
3.1 Introduction . . . 43
3.2 New Generator . . . 44
3.3 Special distributions . . . 47
3.3.1 ELG-Normal . . . 47
3.3.2 ELG-Weibull . . . 48
3.3.3 ELG-Gamma . . . 49
3.3.4 ELG-Log-logistic (ELGLL) . . . 50
3.4 Mathematical properties . . . 52
3.4.1 Moments . . . 52
3.4.2 Incomplete moments . . . 52
3.5 Maximum Likelihood Estimation . . . 53
3.6 Bootstrap confidence intervals . . . 53
3.6.1 Bootstrap percentile interval . . . 53
3.7 Optimization algorithm . . . 54
3.7.1 Swarm intelligence and particle swarm optimization . . . 54
3.7.2 Hardware Used . . . 55

3.7.3 Simulation study . . . 56
3.8 Applications . . . 58
3.9 Conclusions . . . 62

Appendix B Script in Julia language **64**

References **65**

4 The Type I Half-Logistic Family of Distributions **68**
4.1 Introduction . . . 68
4.2 Special TIHL distributions . . . 69
4.2.1 The type I half-logistic normal (TIHL-N) model . . . 69
4.2.2 The type I half-logistic gamma (TIHL-Ga) model . . . 70
4.2.3 The type I half-logistic Fréchet (TIHL-Fr) model . . . 72
4.3 Asymptotes and Shapes . . . 73
4.4 Useful expansions . . . 74
4.5 Generating function . . . 75
4.6 Estimation . . . 76
4.7 Applications . . . 76
4.8 Concluding Remarks . . . 82

References **83**

5 AdequacyModel: An R Package for modeling probability distributions and general optimization **86**
5.1 Introduction . . . 86
5.1.1 A short review on recent families of distributions . . . 87
5.1.2 Heuristic algorithms for NP-complete problems . . . 89
5.2 Conceptual design of the framework . . . 91
5.3 Fitting a distribution . . . 101
5.4 Concluding remarks and the package current usage . . . 108

References **109**

CHAPTER 1

Introduction

The area of survival analysis is important in Statistics and it is commonly applied in biological sciences, engineering, social sciences, among others. Typically, the time of life or failure can have different interpretations depending on the area of application. According to Lai (2013), it may represent:

- the lifetime may mean the life itself of a person;
- the operating time of equipment until its failure;
- the time of survival of a patient with a severe disease from the diagnosis;
- the duration of a social event as a marriage.

The time of life or survival time is a positive continuous random variable, which can have constant, monotonic increasing, monotonic decreasing or non-monotonic (for example, in the form of a U) hazard function. The form of the failure rate function defines a possible probability model for the survival time. For example, when the function failure rate is monotonous, probability distributions traditionally associated with the time of survival are the exponential, Weibull and gamma. In situations in which the function failure rate is unimodal, it is common to the use of the log-logistic or log-normal distributions.

Although no model cited above displays failure rate function in form of U, some models of this type are useful in the survival analysis. Such models can be constructed based on a distribution already known in the literature by means of compositions or mixtures of probability distributions. These compositions, in general, are obtained by a baseline distribution G in a generator of probability distributions. For example, in the last decades new distributions have been defined as extensions of the Weibull distribution. An important characteristic of these new distributions is that several of them show hazard rate function in the form of U, such as the exponentiated Weibull (EW) distribution [see Mudholkar and Srivastava (1993) and Mudholkar et al. (1995)]. This model also presents unimodal hazard rate function.

New distributions are often generated from a modification of a baseline random variable X by (i) linear transformation, (ii) power transformation (e.g. the Weibull is obtained from the exponential), (iii) non-linear transformation (e.g. the lognormal from the normal), (iv) log transformation (e.g. the log Weibull, also known as the type 1 extreme value distribution), and (v) inverse transformation (e.g. the inverse Weibull and inverse gamma models). In what follows, we present two simple transformations for generating new models.

Power transformation

Consider $G(x)$ be the original cumulative distribution function (cdf) and $F(x)$ be the cdf of a new ageing distribution derived from $Y \sim G$ by exponentiating as follows:

- $F(x) = G(x)^{\alpha}$: From such power transformation, we obtain exponentiated Weibull by Mudholkar and Srivastava (1993), the generalized modified Weibull proposed by Carrasco (2008) and the exponentiated Erlang by Lai (2010).
- $F(x) = 1 - \{1 - G(x)\}^{\beta}$: The Lomax model is obtained from the Pareto distribution in this way.

Mixture of distributions

New models are often deduced from mixtures of two or more distributions. Let π be the mixing proportion of two cdfs $F_1(x)$ and $F_2(x)$. The cdf $F(x)$ resulting from the mixture between the two cdfs is given by

$$F(x) = \pi F_1(x) + (1 - \pi)F_2(x).$$

This thesis is composed of independent chapters. Two families of distributions, namely the exponentiated logarithmic generated (ELG) class and the geometric Nadarajah-Haghighi (NHG) class are investigated. The last one is a composition of the Nadarajah-Haghighi and geometric distributions. Further, we develop a statistical library for the `R` programming language called the `AdequacyModel`. This is an improvement of the package that was available on CRAN (Comprehensive `R` Archive Network) and it is currently in version 2.0.0. The two main functions of the library are the `goodness.fit` and `pso` functions. The first function allows to obtain the maximum likelihood estimates (MLEs) of the model parameters and some goodness-of-fit of the fitted probabilistic models. It is possible to choose the method of optimization for maximizing the log-likelihood function. The second function presents the method meta-heuristics global search known as particle swarm optimization (PSO) proposed by Eberhart and Kennedy (1995). Such methodology can be used for obtaining the MLEs necessary for the calculation of some measures of adequacy of the probabilistic models.

This thesis is composed of independent chapters. In Chapter 2 is presented the distribution Nadarajah-Haghighi-geometric (NHG), in which they are discussed various mathematical properties. Studies of the evaluation of intervals estimates by single and double bootstrap percentile (two levels of bootstrap) for the parameters that index the NHG distribution is performed and discussed. Chapter 3 presents a discussion about the Exponentiated Logarithm Generated (ELG) family of distributions. In this chapter are discussed various mathematical properties of the distributions belonging to the ELG family of distributions. In Chapter 4 is presented the generator Type I Half-Logistic (TIHL) family of distributions. Mathematical properties of moments, order statistics and likelihood maximum are constructed and discussed. In Chapter 5 is proposed the library `AdequacyModel` version 2.0.0 for the `R` language. The library `AdequacyModel` is intended for the calculation of statistics of adequacy of adjustment of probabilistic models and optimization of overall purpose. `AdequacyModel` is distributed over the license terms General Public License (GPL$\geq$2.0) in the official repositories of `R` language.

References

[1] Nadarajah, Saralees; Haghighi, Firoozeh. An extension of the exponential distribution. Statistics, v. 45, n. 6, p. 543-558, 2011.

[2] Eberhart, Russ C., and James Kennedy. A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science. v. 1, 1995.

[3] Carrasco, Jalmar MF, Edwin MM Ortega, and Gauss M. Cordeiro. A generalized modified Weibull distribution for lifetime modeling. Computational Statistics and Data Analysis, v. 53, p. 450-462, 2008.

[4] Lai, C. D. Generalized Erlang and mortality levelling off distributions. Mathematical and Computer Modelling, v. 51, p. 1268-1276, 2010.

[5] Mudholkar, Govind S., and Deo Kumar Srivastava. Exponentiated Weibull family for analyzing bathtub failure-rate data. IEEE Transactions on Reliability, v. 42, p. 299-302, 1993.

[6] Lai, C. D. Constructions and applications of lifetime distributions. Applied Stochastic Models in Business and Industry, v. 29, p. 127-140, 2013.

[7] Mudholkar, Govind S., Srivastava, Deo Kumar, and Freimer, Marshall. The exponentiated Weibull family: a reanalysis of the bus-motor-failure data. Technometrics, v. 37, p. 436-445, 1995.

CHAPTER 2

A new class of lifetime models and the evaluation of the confidence intervals by double percentile bootstrap

Abstract: In this chapter, we introduce a new three-parameter distribution by compounding the Nadarajah-Haghighi and geometric distributions, which can be interpreted as a truncated Marshall-Olkin extended Weibull. The compounding procedure is based on the work by Marshall and Olkin (1997), [Marshall, A.W., Olkin, I. (1997). A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. *Biometrika*, v. 84, 641–652]. We prove that the new distribution can be obtained as a compound model with mixing exponential distribution. It can have decreasing, increasing, upside-down bathtub, bathtub-shaped, constant and decreasing-increasing-decreasing failure rate functions depending on the values of its parameters. Some mathematical properties of the new distribution are studied including moments and quantile function. The maximum likelihood estimation procedure is discussed and an EM algorithm is given for estimating the model parameters. We obtain the observed information matrix and discuss inference issues. The flexibility of the new model is illustrated with an application to a real data set.

Keywords: Exponential distribution. failure rate function. geometric distribution. Maximum likelihood estimation. Nadarajah-Haghighi distribution.

2.1 Introduction

Nadarajah and Haghighi (2011) introduced and studied the mathematical properties of an extension of the exponential distribution that allows for increasing, decreasing and constant hazard rate functions (hrfs). The model is referred to as the Nadarajah-Haghighi (NH) distribution. Its cumulative function is given by

$$G(t;\alpha,\lambda) = 1 - \exp\{1 - (1+\lambda\, t)^{\alpha}\}, \quad t > 0, \tag{2.1}$$

where $\lambda > 0$ is the scale parameter and $\alpha > 0$ is the shape parameter. If T follows the Nadarajah-Haghighi distribution, we shall denote $T \sim \mathrm{NH}(\alpha,\lambda)$. The corresponding density function is

$$g(t;\alpha,\lambda) = \alpha\,\lambda\,(1+\lambda\, t)^{\alpha-1}\exp\{1 - (1+\lambda\, t)^{\alpha}\}, \quad t > 0. \tag{2.2}$$

The NH distribution presents several advantages if compared with some well-known generalizations of the Exponential model, such as the Gamma, Weibull and Exponentiated Exponential (EE)

distributions. For example, unlike these distributions, the NH distribution allows for an increasing hrf when its corresponding density is monotonically decreasing. The second advantage is the ability to model data that have their mode fixed at zero. Other advantage is based on a mathematical relationship with the Weibull distribution, in which the NH model can be interpreted as a truncated Weibull distribution. These three facts combined may attract more complex applications in the literature of lifetime distributions.

The NH distribution is not the only extension of the exponential distribution. In addition to the gamma and Weibull distributions, many other generalizations have been proposed over the years. One of them is

$$G(t;\rho,\lambda,\alpha) = (1-\rho\,\mathrm{e}^{-\lambda t})^{\alpha}, \quad t > \lambda^{-1}\log\rho, \tag{2.3}$$

where $\rho > 0$, $\lambda > 0$ and $\alpha > 0$, which was discussed by Gompertz (1825) to compare known human mortality tables and to represent mortality growth. Note that the EE distribution, discussed in Gupta et al. (1998), is actually a particular case of equation (2.3) when $\rho = 1$. Nadarajah and Kotz (2006) and Barreto-Souza et al. (2010) proposed two important extensions of the exponential model, which are the beta exponential (BE) and beta generalized exponential (BGE) distributions, respectively. The BE and BGE distributions are special cases of the beta family of distributions, which was introduced by Eugene et al. (2002). The beta family still contains other generalizations of the exponential model such as the beta Weibull (BW) [Lee et al. (2007)] and beta modified Weibull (BMW) [Silva et al. (2010)] distributions. The Kumaraswamy class, introduced by Cordeiro and Castro (2011), also contains several models that extend the exponential distribution, such as the Kumaraswamy Weibull (KwW) [Cordeiro et al. (2010)] and Kumaraswamy generalized Rayleigh (KwGR) [Gomes et al. (2014)] distributions. References about other generalizations of the exponential model are widespread and the reader can to see those listed in the above papers.

In this chapter, we introduce a new continuous distribution, which is an extension of the NH model, by compounding the NH and geometric distributions. The new model is therefore another extension of the exponential distribution and is referred to as the Nadarajah-Haghighi geometric (NHG) distribution. The proposed distribution is more flexible for modeling lifetime data, namely in reliability, in terms of its failure rate shapes, which are constant, decreasing, increasing, upside-down bathtub and bathtub shaped. The compounding procedure follows the pioneering work by Marshall and Olkin (1997). In the same way, several classes of distributions were proposed by compounding some useful lifetime and power series (PS) distributions in the last few years. Chahkandi and Ganjali (2009), introduced the exponential power series (EPS) class of distributions, which contains as special cases the exponential Poisson (EP), exponential geometric (EG) and exponential logarithmic (EL) distributions. Morais et al. (2011) defined the Weibull power series (WPS) class, which includes as sub-models the EPS distributions. The WPS distributions can have increasing, decreasing and upside down bathtub hrfs. The generalized exponential power series (GEPS) distributions were proposed by Mahmoudi and Akbar (2012) following the same approach of Morais and Barreto-Souza (2011). Silva et al. (2013) studied the extended Weibull power series (EWPS) family, which includes as special models the EPS and WPS families. Bourguignon et al. (2014) extended the Birnbaum-Saunders distribution through the class of Birnbaum-Saunders power series (BSPS) distributions. In a very recent paper, Silva and Cordeiro (2015) introduced the Burr XII power series (BXIIPS) family of distributions.

2.2 Construction of the NHG distribution

Let $T_1, \ldots, T_N$ be independent and identically distributed (iid) NH random variables with cdf (2.1) and pdf (2.2). We assume that N has a zero-truncated geometric distribution independent of the T's with probability mass function (pmf) given by

$$p_n = P(N = n) = (1 - p)\, p^{n-1},\ n = 1, 2, \ldots, \tag{2.4}$$

where $p \in (0, 1)$. Let $X = \min(T_1, \ldots, T_N)$. Then, the conditional random variable $(X|N = n)$ has cumulative distribution function (cdf)

$$P(X \leq x, N = n) = (1 - p)\, p^{n-1}(1 - \{\exp[1 - (1 + \lambda\, x)^{\alpha}]\}^n), \quad x > 0, \quad n \geq 1.$$

The Nadarajah-Haghighi-geometric (NHG) distribution is defined by the marginal cdf of X

$$F(x; \alpha, \lambda, p) = \frac{1 - \exp[1 - (1 + \lambda\, x)^{\alpha}]}{1 - p \exp[1 - (1 + \lambda\, x)^{\alpha}]}, \quad x > 0. \tag{2.5}$$

Hereafter, the random variable X following (2.5) with parameters α, λ and p is denoted by $X \sim \text{NHG}(\alpha, \lambda, p)$. The pdf of X is

$$f(x; \alpha, \lambda, p) = \frac{(1 - p)\, \alpha\, \lambda\, (1 + \lambda\, x)^{\alpha-1} \exp[1 - (1 + \lambda\, x)^{\alpha}]}{\{1 - p \exp[1 - (1 + \lambda\, x)^{\alpha}]\}^2}, \quad x > 0. \tag{2.6}$$

It can be shown that

$$\lim_{x \to 0} f(x; \alpha, \lambda, p) = \alpha\, \lambda/(1 - p) \ \text{ and } \ \lim_{x \to \infty} f(x; \alpha, \lambda, p) = 0.$$

We can then define the NHG distribution by (2.6) for any $p < 1$. The study of the new distribution is important since it extends some distributions previously considered in the literature. In fact, the NH distribution is obtained by taking $p = 0$ [see Nadarajah and Haghighi (2011)]. The EG distribution (Adamidis and Loukas, 1998) follows by taking $\alpha = 1$ and $0 < p < 1$, whereas the EEG distribution (Adamidis et al., 2005) is obtained when $\alpha = 1$ for any $p < 1$. Clearly, the EEG distribution extends the EG distribution. For $\alpha = 1$ and $p = 0$, Equation (2.6) reduces to the exponential distribution. When $p \to 1^-$, the NHG distribution distribution converges to a distribution degenerated at zero, i.e., $P(X = 0) = 1$. Hence, the parameter p can be interpreted as a degeneration parameter. Figure 2.1 displays the pdf for selected parameter values.

Proposition 2.2.1. *The NHG density function is log-convex if $\alpha < 1$ and $0 \leq p < 1$, and it is log-concave if $\alpha > 1$ and $p \leq 0$.*

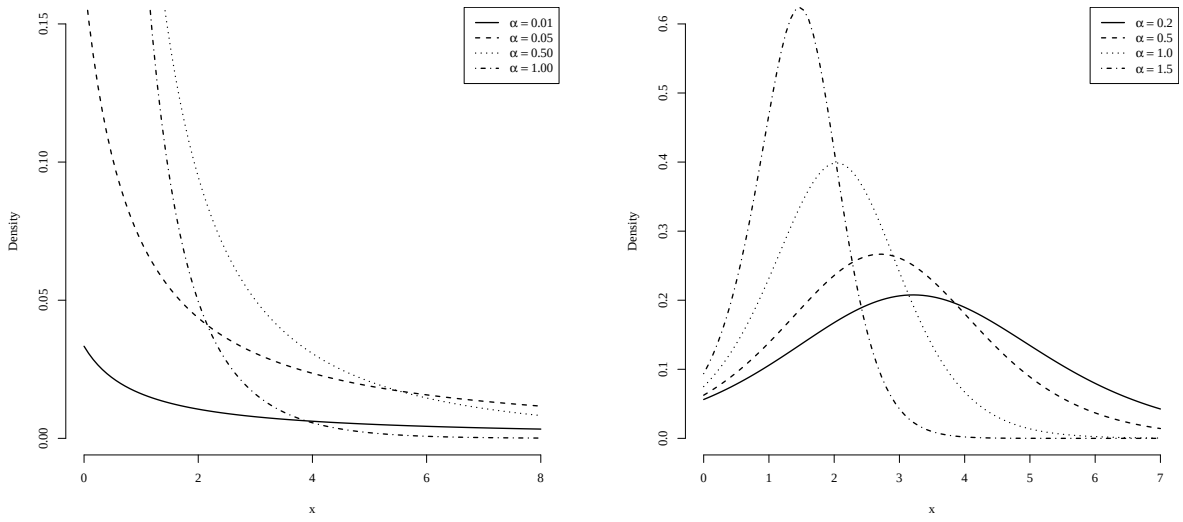
Proof. Let $z = (1 + \lambda\, x)^{\alpha}$. It implies that $z > 1$ for $x > 0$. We have $x = (z^{1/\alpha} - 1)/\lambda$. Rewriting the NHG density function as a function of z, $\delta(z)$ say, we obtain

$$\delta(z) = \frac{(1 - p)\, \alpha\, \lambda\, z^{(\alpha-1)/\alpha}\, \mathrm{e}^{1-z}}{[1 - p\, \mathrm{e}^{1-z}]^2}, \quad z > 1.$$

The second derivative of $\log[\delta(z)]$ with respect to z is given by

$$\frac{d^2\, \log[\delta(z)]}{d\, z^2} = -\left[\frac{(\alpha - 1)}{\alpha\, z^2} - \frac{2\, p\, \mathrm{e}^{1-z}}{[1 - p\, \mathrm{e}^{1-z}]^2}\right].$$

□



(a) $p = -15$ (b) $p = 0.9$

Figure 2.1: The NHG density function for some parameter values; $\lambda = 1$.

The main motivation for this new distribution is based on four points:

1. Ability (or the inability) of the NHG distribution to model data that have their mode fixed at zero.

2. As we shall see later, the NHG hazard rate function (hrf) can be constant, decreasing, increasing, decreasing-increasing-decreasing, upside-down bathtub, bathtub-shaped or constant.

3. If Y is a Marshall-Olkin extended Weibull random variable with shape parameters α and scale parameter λ, then the density in (2.6) is the same as that of the random variable $Z = Y - \lambda^{-1}$ truncated at zero; that is, the NHG distribution can be interpreted as a truncated Marshall-Olkin extended Weibull distribution.

4. It can be applied in some interesting situations such as:

   • Time to relapse of cancer under the first-activation scheme. Here, N is the number of carcinogenic cells for an individual left active after the initial treatment and X_i is the time spent for the ith carcinogenic cell to produce a detectable cancer mass, for $i \geq 1$;

   • Time to the first failure. Suppose that the failure of a device occurs due to the presence of an unknown number N of initial defects of same kind, which can be identifiable only after causing failure and are repaired perfectly;

   • Reliability. From the stochastic representations $X = \min\{X_i\}_{i=1}^{N}$ and $Z = \max\{X_i\}_{i=1}^{N}$, we note that the NHG model can arise in series and parallel systems with identical components, which appear in many industrial applications and biological organisms.

Proposition 2.2.2. *The distribution of the form (2.5) is geometric extreme stable.*

Proof. The proof follows easily using the arguments by [23]. We omit the details. □

Proposition 2.2.3. *The density function of X can be expressed as an mixture of densities of minimum order statistics of T.*

Proof. For any positive real number a, and for $|z| < 1$, we have the generalized binomial expansion

$$(1-z)^{-a} = \sum_{n=0}^{\infty} \frac{\Gamma(a+n)}{\Gamma(a)\, n!} z^n, \tag{2.7}$$

where $\Gamma(\cdot)$ is the gamma function. Applying (2.7) to (2.6), yields

$$f(x;\alpha,\lambda,p) = \sum_{n=1}^{\infty} p_n\, f_{T_{(1)}}(x;\alpha,\lambda,n), \quad x > 0,$$

where $\sum_{n=1}^{\infty} p_n = 1$ and $f_{T_{(1)}}(t;\alpha,\lambda,n)$ is the density function of $T_{(1)} = \min(T_1,\ldots,T_n)$, for fixed n, given by

$$f_{T_{(1)}}(t;\alpha,\lambda,n) = n\,\alpha\,\lambda\,(1+\lambda\,t)^{\alpha-1}\{\exp[1-(1+\lambda\,t)^{\alpha}]\}^n, \quad t > 0.$$

□

The distribution with cdf (2.8) is called the *complementary Nadarajah-Haghighi-geometric* (CNHG) distribution. This distribution is a suitable model in a complementary risk problem based in the presence of latent risks which arise in several areas such as public health, actuarial science, biomedical studies, demography and industrial reliability (Basu and Klein, 1982). However, in this work, we do not focus on this alternative class of distributions.

Remark. *Let $Y = \max(T_1,\ldots,T_N)$, then the cdf and pdf of Y are*

$$F_Y(y;\alpha,\lambda,p) = \frac{(1-p)\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\}}{1-p\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\}}, \quad y > 0 \tag{2.8}$$

and

$$f_Y(y;\alpha,\lambda,p) = \frac{(1-p)\,\alpha\,\lambda\,(1+\lambda\,x)^{\alpha-1}\{\exp[1-(1+\lambda\,x)^{\alpha}]\}}{(1-p\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\})^2}.$$

Proposition 2.2.4. *Let $X \sim NHG(\alpha,\lambda,p)$. Then:*

i) The cdf of the nth order statistic corresponding to the pdf (2.6) is given by

$$F_n(x) = [F(x;\alpha,\lambda,p)]^n = \left\{\frac{1-\exp[1-(1+\lambda\,x)^{\alpha}]}{1-p\exp[1-(1+\lambda\,x)^{\alpha}]}\right\}^n; \tag{2.9}$$

ii) The density function of the nth order statistic is given by

$$f_n(x) = \frac{n(1-p)f(x;\alpha,\lambda,p)[F(x;\alpha,\lambda,p)]^{n-1}}{\{1-p[1-F(x;\alpha,\lambda,p)]\}^{n+1}}.$$

Proof. The proof of Proposition 4 is trivial. □

The NHG survival function is given by

$$S(x;\alpha,\lambda,p) = \frac{(1-p)\exp[1-(1+\lambda\,x)^{\alpha}]}{1-p\exp[1-(1+\lambda\,x)^{\alpha}]}, \quad x > 0,$$

and the corresponding hrf (for $x > 0$) becomes

$$h(x;\alpha,\lambda,p) = \frac{\alpha\,\lambda\,(1+\lambda\,x)^{\alpha-1}}{1-p\,\exp[1-(1+\lambda\,x)^{\alpha}]} = \frac{h_{NH}(x;\alpha,\lambda)}{1-p\,\exp[1-(1+\lambda\,x)^{\alpha}]},$$

where $h_{NH}(x;\alpha,\lambda)$ is the NH hrf.

Note that NH hrf $h(x;\alpha,\lambda,p)/h(x;\alpha,\lambda)$ is increasing in x for $p < 0$ and decreasing in x for $0 < p < 1$. Further, we have

$$\lim_{x\to 0} h(x;\alpha,\lambda,p) = \frac{\alpha\,\lambda}{1-p} \text{ and } \lim_{x\to\infty} h(x;\alpha,\lambda,p) = \begin{cases} 0, & \alpha < 1, \\ \infty, & \alpha > 1, \\ \alpha\,\lambda, & \alpha = 1. \end{cases}$$

Proposition 2.2.5. *For $\lambda > 0$, the NHG distribution has an increasing hrf if $\alpha > 1$ and $p < 0$, and it has a decreasing hrf if $\alpha < 1$ and $0 < p < 1$. It is constant if $\alpha = 1$ e $p = 0$.*

Proof. Let $z = (1+\lambda\,x)^{\alpha}$. It implies that $z > 1$ for $x > 0$. We have $x = (z^{1/\alpha}-1)/\lambda$. Now, rewriting the NHG hrf as a function of z, $\eta(z)$ say, we obtain

$$\eta(z) = \frac{\alpha\,\lambda\,z^{(\alpha-1)/\alpha}}{1-p\,\exp(1-z)}, \quad z > 1.$$

The first derivative of $\eta(z)$ with respect to z is given by

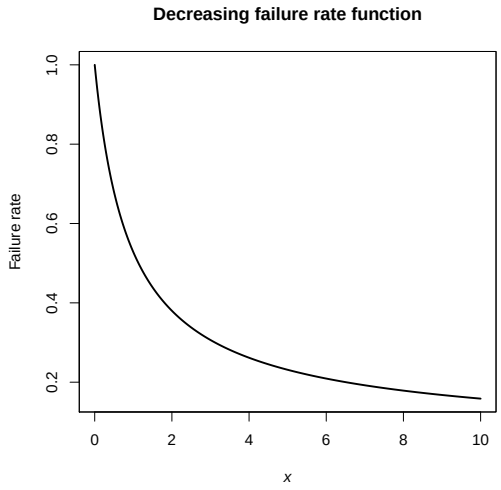
$$\eta'(z) = \frac{\lambda\,z^{-1/\alpha}}{[1-p\,\exp(1-z)]^2}\,\Delta(z), \quad z > 1,$$

where

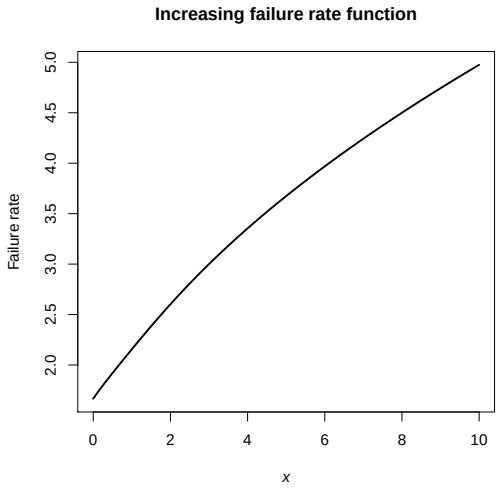
$$\Delta(z) = (\alpha-1)(1-p\,\mathrm{e}^{1-z}) - \alpha\,\lambda\,z\,\mathrm{e}^{1-z}.$$

For $\alpha > 1$ and $p < 0$, $\Delta(z) > 0$, and hence $\eta'(z) > 0$, i.e., $\eta(z)$ is increasing in $(1,\infty)$. For $\alpha < 1$ and $0 < p < 1$, $\Delta(z) < 0$, and hence $\eta'(z) < 0$, i.e., $\eta(z)$ is decreasing in $(1,\infty)$. □

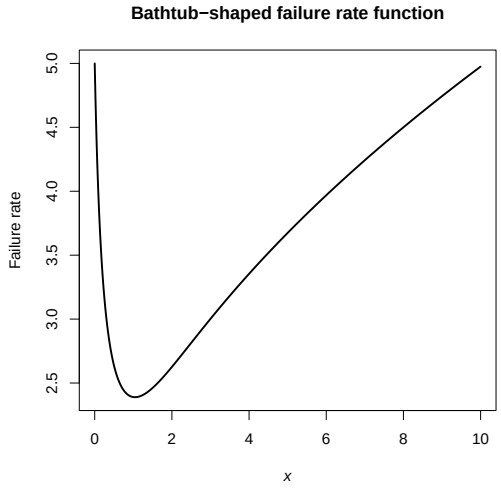
Figure 2.2 displays some plots of the NHG hrf for some parameter values. The parameter λ does not change the shape of the hrf since it is a scale parameter. It is evident that the NHG hrf can be decreasing, increasing, upside-down bathtub shaped or bathtub-shaped. It is difficult to determine analytically the parameter spaces corresponding to these shapes. It is interesting to point out that the NHG hrf can also be decreasing-increasing-decreasing. So, the NH distribution is quite flexible and can be used effectively in analyzing real data in several areas. Thus, the beauty and importance of the new distribution lies in its ability to model monotone as well as non-monotone failure rates, which are quite common in reliability and biological studies.



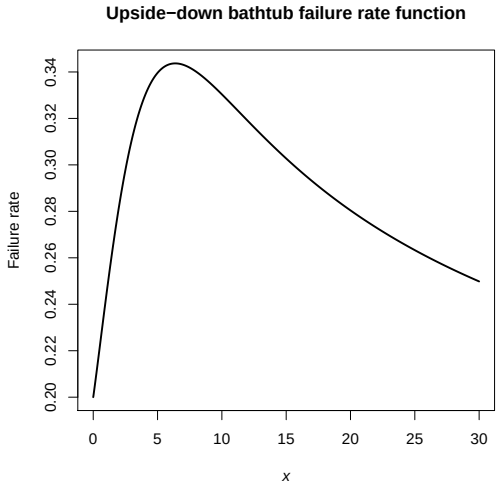
(a) $\alpha = 0.5$ and $p = 0.5$



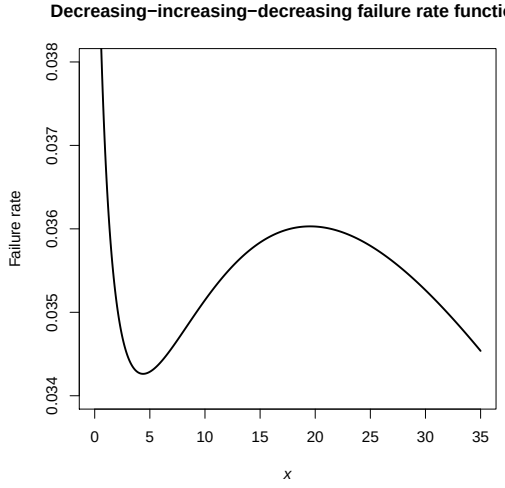
(b) $\alpha = 1.5$ and $p = 0.1$



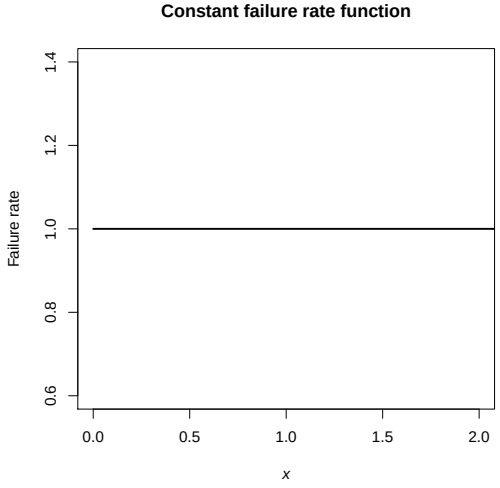
(c) $\alpha = 1.5$ and $p = 0.7$



(d) $\alpha = 0.7$ and $p = -2.5$



(e) $\alpha = 0.4$ and $p = -8.5$



(f) $\alpha = 1.0$ and $p = 0.0$

Figure 2.2: The NHG hrf for some parameter values; $\lambda = 1$.

2.3 Maximum likelihood estimation

In this section, we determine the maximum likelihood estimates (MLEs) of the parameters of the new distribution from complete samples only. Let $x_1, \ldots, x_n$ be observed values from the NHG distribution with parameters α, λ and p. Let $\boldsymbol{\theta} = (\alpha, \lambda, p)^\top$ be the parameter vector. The total log-likelihood function for $\boldsymbol{\theta}$ is given by

$$\begin{aligned}\ell_n(\boldsymbol{\theta}) &= n + n\,\log[(1-p)\,\alpha\,\lambda] + (\alpha-1)\sum_{i=1}^{n}\log(1+\lambda\,x_i) - \sum_{i=1}^{n}(1+\lambda\,x_i)^{\alpha} \\ &- 2\sum_{i=1}^{n}\log\{1-p\,\exp[1-(1+\lambda\,x_i)^{\alpha}]\}.\end{aligned}$$

By taking the partial derivatives of the log-likelihood function with respect to the parameters in $\boldsymbol{\theta}$, we obtain the components of the score vector $\boldsymbol{U}_{\boldsymbol{\theta}} = (U_\alpha, U_\lambda, U_p)^\top$:

$$\begin{aligned}U_\alpha &= \frac{n}{\alpha} + \frac{1}{\alpha}\sum_{i=1}^{n}(1-\dot{\nu}_i)\log(\dot{\nu}_i) - \frac{2\,p}{\alpha}\sum_{i=1}^{n}\dot{\pi}_i\,\dot{\nu}_i\log(\dot{\nu}_i), \\ U_\lambda &= \frac{n}{\lambda} + (\alpha-1)\sum_{i=1}^{n}x_i\,\dot{\nu}_i^{-1/\alpha} - \alpha\sum_{i=1}^{n}x_i\,(2p+\dot{\pi}_i)\,\dot{\nu}_i^{1-1/\alpha} \quad \text{and} \quad U_p = 2\sum_{i=1}^{n}\dot{\pi}_i - \frac{n}{1-p},\end{aligned}$$

where

$$\dot{\nu}_i = (1+\lambda\,x_i)^{\alpha} \text{ and } \dot{\pi}_i = \frac{\exp[1-(1+\lambda\,x_i)^{\alpha}]}{1-p\,\exp[1-(1+\lambda\,x_i)^{\alpha}]},\ i = 1, \ldots, n.$$

Setting U_α, U_λ and U_p equal to zero and solving the equations simultaneously yields the MLE $\widehat{\boldsymbol{\theta}} = (\widehat{\alpha}, \widehat{\lambda}, \widehat{p})^\top$ of $\boldsymbol{\theta} = (\alpha, \lambda, p)^\top$. These equations cannot be solved analytically and statistical software can be used to solve them numerically using iterative methods such as the Newton-Raphson type algorithms.

2.4 Optimization algorithm

In Computer Science, the particle swarm optimization (PSO) is a computational method for optimization of parametric and multiparametric functions. The PSO algorithm is a meta-heuristic method, which has been providing good solutions for problems of global optimization functions with box-constrained. The use of meta-heuristic methods such as PSO has proved to be useful for maximizing complicated log-likelihood functions without the need for early kick functions as the BFGS, L-BFGS-B, Nelder-Mead and simulated annealing methods. As in most heuristic methods that are inspired by biological phenomena, the PSO method is inspired by the behavior of flying birds. The philosophical idea of the PSO algorithm is based on the collective behavior of birds (particle) in search of food (point of global optimal). The PSO technique was first defined by Eberhart and Kennedy (1995) in a paper published in the Proceedings of the IEEE International Conference on Neural Networks IV. A modification of the PSO algorithm was proposed by Shi and Eberhart (1998). Further details on the philosophy of the PSO method are given in the book Swarm Intelligence (see Kennedy et al., 2001).

The PSO optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle's

position and velocity. The movement of the particles in the search space is randomized. Each iteration of the PSO algorithm, there is a leader particle, which is the particle that minimizes the objective function in the corresponding iteration. The remaining particles arranged in the search region will follow the leader particle randomly and sweep the area around this leading particle. In this local search process, another particle may become the new leader particle and the other particles will follow the new leader randomly. In each particle arranged in the search region has a velocity vector and position vector and its movement in the search region is given by changes in these vectors. The PSO algorithm is presented below, where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function to be minimized, S is the number of particles in the swarm (set of feasible points, i.e. search region), each particle having a vector position $x_i \in \mathbb{R}^n$ in the search-space and a vector velocity defined by $v_i \in \mathbb{R}^n$. Let p_i be the best known position of particle i and g the best position of all particles.

1. For each particle $i = 1, \ldots, S$ do:
   - Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
   - Initialize the particle's best known position to its initial position: $p_i \leftarrowtail x_i$.
   - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
   - Initialize the particle's velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$.
2. Until a termination criterion is met (e.g., number of iterations performed, or a solution with adequate objective function value is found), repeat:
   - For each particle $i = 1, \ldots, S$ do:
     - Pick random numbers: $r_p, r_g \sim U(0, 1)$.
     - For each dimension $d = 1, \ldots, n$ do:
       * Update the particle's velocity: $v_{i,d} \leftarrowtail \omega\, v_{i,d} + \varphi_p r_p(p_{i,d} - x_{i,d}) + \varphi_g r_g(g_d - x_{i,d})$.
     - Update the particle's position: $x_i \leftarrowtail x_i + v_i$
     - If $f(x_i) < f(p_i)$ do:
       * Update the particle's best known position: $p_i \leftarrowtail x_i$
       * If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
3. Now g holds the best found solution.

The parameter ω is called inertia coefficient and as the name implies controls the inertia of each particle arranged in the search region. The quantities ω_p and ω_g control the acceleration of each particle and are called acceleration coefficients. The PSO algorithm described above is implemented in the programming language `R` is presented below.

This algorithm with few modifications will be implemented in the `AdequacyModel` package available on the website of `R`. The algorithm above is quite general and can be applied to maximize any function involving or not a database. Using `pso`, a given function is maximized taking into consideration vectors of restrictions delimiting the search-space. In truth, the pso function above is constructed to minimize any function. However, to maximize f is equivalent to minimize $-f$. A brief description of the function arguments `pso` are listed below.

- `func`: Objective function that we want to maximize;

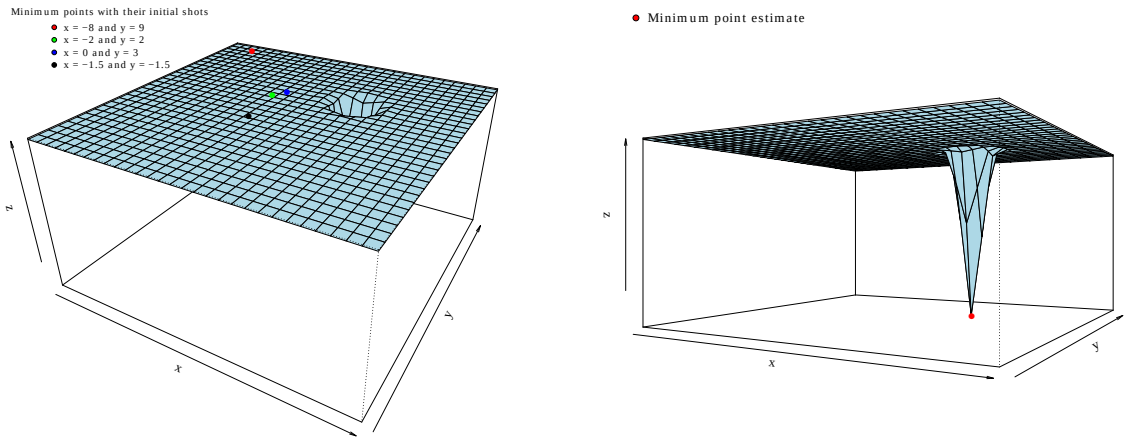
- `S`: Number of particles considered. By default, the number of particles is equal to 150;
- `lim_inf` e `lim_sup`: Vectors that restrict the region-search inferiorly and superiorly, respectively.
- `e`: Error considered. The algorithm stops if the variance in the last 10 iterations is less than or equal to `e`;
- `data`: By default `data = NULL`, but when the `func` is a log-likelihood function, `data` is a data vector;
- `N`: Maximum number of iterations.

One advantage of the PSO method is the we do not need to concern ourselves with initial parameter values. Problems with initial values are frequent in methods such as the BFGS when the objective function involves flat or nearly flat regions. Depending on the initial values provided, we can obtain estimates totally different. In general, it does not occur with great frequency in methodologies of heuristic search, whose update steps embed randomness (generation of pseudo-random number). The example below shows clearly this problem and the use of the function `pso`, especially how to specify the objective function for the argument `func`. The PSO method is described in the Appendix "A. Code in `R` language for the PSO method".

Example

Consider Easom function $f(x, y) = -\cos(x)\cos(y)\exp\{-[(x-\pi)^2 + (y-\pi)^2]\}$, and $-10 \leq x, y \leq 10$. The Easom function is minimized at $x = y = \pi$, with $f(\pi, \pi) = -1$. The use of the `pso` function to minimize the above function is

```
easom <- function(par,x){
   x1 = par[1]
   x2 = par[2]
   -cos(x1)*cos(x2)*exp(-((x1-pi)^2 + (x2-pi)^2))
}
set.seed(0)
pso(func=easom,S=350,lim_inf=c(-10,-10),lim_sup=c(10,10))
```



(a) Minimization by BFGS method

(b) Minimization by PSO method

Figure 2.3: Points of minimum obtained by `optim` and `pso` functions the using methods BFGS and PSO.

In both minimizations there was convergence. For minimization through BFGS method we use the `optim` function of the `R` linguage with the argument `method = "BFGS"`. For more details on the function `optim` do it `help(optim)`. Figure 2.3(a) presents four estimates obtained by the BFGS method whose points are the points of minimum obtained by the BFGS method, whose the BFGS procedure with different initial shots. In the legend are described the initial shots. In these four estimates for different shots the `optim` function indicates convergence.

Before performing the `pso` function, the seed of the Mersenne-Twister generator proposed by Matsumoto and Nishimura (1998) is fixed at zero, i.e, `set.seed(0)`. Is it possible to perceive clearly through Figure 2.3 that the minimization method using the PSO function `pso` is much more efficient and approaches better the global minimum.

2.5 Bootstrap confidence intervals

The bootstrap method was introduced in 1979 by Bradley Efron. This method was inspired by a previous methodology based on resampling called jackknife. Efron 1979 summarized the methodologies based on resampling that until then existed and established a new area of research. The idea of replacing complicated and often inaccurate simulation methods based on resampling approaches has attracted several researchers to develop methodologies based on bootstrap for various purposes. With the popularization of the bootstrap method, some researchers have begun to establish mathematical conditions under which the bootstrap is justifiable.

In the literature there are many jobs that make use of bootstrap methodologies. In general, the bootstrap method is used to correct the biases of the estimators, construct of confidence intervals, hypothesis tests, estimation of standard errors of estimators, among others.

The bootstrap methods present two approaches, namely the parametric bootstrap and nonparametric bootstrap. Parametric bootstrap refers to the case where the resampling is performed based on a distribution $F(\widehat{\theta})$ known or established, wherein $\widehat{\theta}$ is an estimator for θ. On the other hand, in bootstrap nonparametric there is the lack of a true distribution F. The resampling is based on the empirical distribution function $\widehat{F}_n$. Resample of $\widehat{F}_n$ is equivalent to resample data with replacement.

2.5.1 Bootstrap percentile interval

Let T_n be an estimator of the scale θ based on n observations and t its estimate. Let T_n^* be the same estimator based on n observations resampling from the original sample with replacement and t^* its estimate. For simplicity, suppose T_n is a continuous random variable. Denoting the pth quantile of the distribution of the random variable $T_n - \theta$ by a_p, we obtain

$$\Pr\left\{T_n - \theta \leq a_{\frac{\alpha_1}{2}}\right\} = \Pr\left\{T_n - \theta \geq a_{1-\frac{\alpha_2}{2}}\right\} = \frac{\alpha}{2}.$$

As the amount $Q = T_n - \theta$ is invertible and T_n depends only on the sample, we can construct the confidence interval for θ by rewriting the events above, i.e., we can replace the events $T_n - \theta \leq a_{\frac{\alpha_1}{2}}$ and $T_n - \theta \geq a_{1-\frac{\alpha_2}{2}}$ by $\theta > T_n - a_{\frac{\alpha_1}{2}}$ and $\theta < T_n - a_{1-\frac{\alpha_2}{2}}$, respectively. Thus, the confidence interval of level γ has the limits

$$\ell_{\alpha/2} = t - a_{1-\frac{\alpha_2}{2}},\ \ell_{1-\alpha/2} = t - a_{\frac{\alpha_1}{2}}.$$

In many situations, we do not know the distribution of $T_n - \theta$. In such cases, suppose that there is some transformation T_n, $U = h(T_n)$, such that U has a symmetric distribution. Suppose also that we can obtain the confidence interval of level $1 - \alpha$ to $\phi = h(\theta)$. According to Davison and Hinkley (1997), we may use bootstrapping to obtain an approximation of the distribution of $T_n - \theta$ using the distribution of $T_n^* - t$. Thus, we estimate the p quantile of $T_n - \theta$ by the $(J+1)p$-th ordered value of $t^* - t$, i.e., the p-th quantile of $T_n - \theta$ is estimated by $t^*_{((J+1)p)} - t$. Similarly, the p quantile of $h(T_n) - h(\theta) = U - \phi$ can be estimated by the $(J+1)p$-th ordered value of $h(T_n^*) - h(t) = u^* - u$. Let b_p be the p-th quantile of $U - \phi$. Since U has a symmetrical distribution, then $U - \phi$ also has a symmetrical distribution, as soon as it is true that $b_{\frac{\alpha}{2}} = -b_{1-\frac{\alpha}{2}}$. Using the symmetry of $U - \phi$, we have $h(\ell_{\alpha/2}) = u + b_{\alpha/2}$ and $h(\ell_{1-\alpha/2}) = u + b_{1-\alpha/2}$. As $b_{\alpha/2}$ and $b_{1-\alpha/2}$ are quantiles of $U - \phi$ and we can obtain these quantiles, the lower and upper limits of the confidence intervals are given by $u + (u^*_{((J+1)\alpha/2)} - u)$ and $u + (u^*_{((J+1)(1-\alpha/2))} - u)$, respectively, leading to the limits

$$u^*_{((J+1)\alpha/2)},\quad u^*_{((J+1)(1-\alpha/2))},$$

whose transformation to θ is

$$t^*_{(J+1)\alpha/2},\quad t^*_{(J+1)(1-\alpha/2)}. \tag{2.10}$$

Note that we do not know the transformation h. The confidence interval of level $1 - \alpha$ for the parameter θ does not involve h and it can be evaluated without knowledge of this transformation. The interval (2.10) is known as the bootstrap interval percentile. According to Davison and Hinkley (1997, p. 203) the percentile method can be applied to any statistic.

2.5.2 Interval Double Bootstrap Percentile

When using the percentile method, we can obtain a coverage different from the desired level $(1-\alpha)$. The interesting thing is that we can continue making use of bootstrap to correct this discrepancy. This fact reveals the flexibility of the bootstrap method.

The idea to obtain confidence intervals more accurate is to make use of the double bootstrap, i.e., for each replica of the original bootstrap there will be conducted another bootstrap. Consider the situation where only a trust boundary is of interest and it is the upper limit with nominal confidence level $1-\alpha$, where

$$\Pr\{T_n - \theta \leq a_\alpha(F) \mid F\} = \Pr\left\{t(\widehat{F}_n) - t(F) \leq a_\alpha(F) \mid F\right\} = \alpha.$$

Ignoring the errors of simulation, which is really evaluated is the confidence limit $t(\widehat{F}_n) - a_\alpha(\widehat{F}_n)$. The bias of the bootstrap percentile follows from the fact that $a_\alpha(\widehat{F}_n) \neq a_\alpha(F)$, which implies

$$\Pr\left\{t(F) \leq t(\widehat{F}_n) - a_\alpha(\widehat{F}_n) \mid F\right\} \neq 1-\alpha. \tag{2.11}$$

Davison and Hinkley (1997) propose to correct the bias by adding a correction to $a_\alpha(\widehat{F}_n)$, However, an approach more successful is to adjust the index α. Thus, we can replace $a_\alpha(\widehat{F}_n)$ by $a_{q(\alpha)}(\widehat{F}_n)$ and estimate what the adjusted value $(\widehat{q}_\alpha)$ it must be used. Therefore, we obtain $q(\alpha)$ satisfying

$$\Pr\left\{t(F) \leq t(\widehat{F}_n) - a_{q(\alpha)}(\widehat{F}_n) \mid F\right\} = 1-\alpha. \tag{2.12}$$

Note that the solution $q(\alpha)$ depends on F, i.e., $q(\alpha) = q(\alpha, F)$. Since the distribution F is unknown, we can estimate $q(\alpha)$ by $\widehat{q}(\alpha) = q(\alpha, \widehat{F}_n)$. Let $x_n^* = \{X_1^*, X_2^*, \ldots, X_n^*\}$ by a sample obtained randomly with replacement of x_n and $x_n^{**} = \{X_1^{**}, X_2^{**}, \ldots, X_n^{**}\}$ a new sample obtained by refitting x_n^* with empirical distribution functions given by $\widehat{F}_n^*$ and $\widehat{F}_n^{**}$, respectively. Let T_n^* and T_n^{**} be the statistics T_n evaluated at x_n^* and x_n^{**}, where t^* and t^{**} are the estimates, respectively. We denote $\Pr^*\{\cdot\}$ as a conditional probability in $\widehat{F}_n$ and $\Pr^{**}\{\cdot\}$ as a conditional probability in $\widehat{F}_n^*$. We can obtain $\widehat{q}(\alpha)$ using the bootstrap version of equation (2.12) given by

$$\Pr^*\left\{t(\widehat{F}_n) \leq t(\widehat{F}_n^*) - a_{\widehat{q}(\alpha)}(\widehat{F}_n^*) \mid \widehat{F}_n\right\} = 1-\alpha. \tag{2.13}$$

From the definition (2.13), a scheme involving a second level of bootstrap is given by

$$\Pr^*\left[\Pr^{**}\left\{T_n^{**} \leq 2T_n^* - t \mid \widehat{F}_n^*\right\} \geq \widehat{q}(\alpha) \mid \widehat{F}_n\right] = 1-\alpha. \tag{2.14}$$

Davison and Hinkley (1997), the coverage $1-\alpha+O(n^{-a})$ is corrected for $1-\alpha+O(n^{-a-1/2})$. For unilateral confidence limits, we have $a=\frac{1}{2}$ or $a=1$. For the cases where the bilateral confidence interval, the coverage $1-\alpha+O(n^{-1})$ is corrected to $1-\alpha+O(n^{-2})$.

In general, especially in non-parametric problems, the calculation of (2.14) can not be done exactly. Thus, approximate methods must be used. A basic algorithm is given as follows. Suppose we have J samples of $\widehat{F}_n$ and denote these estimates by $\widehat{F}_{n,1}^*, \ldots, \widehat{F}_{n,j}^*$, where $\widehat{F}_{n,j}^*$ is the j-th empirical distribution function. Set

$$u_j^* = \Pr(T_n^{**} \leq 2t_j^* - t \mid \widehat{F}_{n,j}^*). \tag{2.15}$$

The values $u_1^*, \ldots, u_j^*$ can be determined by approximation. We generate K samples of $\widehat{F}_{n,j}^*$ and for each of them we obtain $t_{j,1}^{**}, \ldots t_{j,k}^{**}$, for $k = 1, 2, \ldots, K$. So,

$$u_{K,j}^* = K^{-1} \sum_{k=1}^{K} I\{t_{j,k}^{**} \leq 2t_j^* - t\}, \tag{2.16}$$

wherein $I\{A\}$ is the indicator function for a event A. The Monte Carlo version of (2.14) is given by

$$J^{-1} \sum_{j=1}^{J} I\{u_{K,j}^* \geq \widehat{q}(\alpha)\} = 1 - \alpha, \tag{2.17}$$

where $\widehat{q}(\alpha)$ is the quantile α of $u_{K,j}^*$. The simplest way to obtain $\widehat{q}(\alpha)$ is by sorting the values $u_{K,j}^*$ such that $u_{K,1}^* \leq u_{K,2}^* \leq \cdots \leq u_{K,J}^* \in (0,1)$, and setting $\widehat{q}(\alpha) = u_{K,(\alpha(J+1))}^*$. The $(J+1)\alpha$-th quantile the $u_{K,j}^*$ is used to obtain the corrected quantile of $t_j^* - t$. The double bootstrap percentile intervals algorithm for bilateral is presented below:

1. For a given sample (original sample) calculate the quantity t;
2. Generate J samples and obtain t_j^*, for $j = 1, \ldots, J$;
3. Generate K new bootstrap samples for each of the J samples in the previous step, and for each one, calculate $t_{j,k}^{**}$, with $k = 1, 2, \ldots, K$;
4. Calculate

$$u_j^* = K^{-1} \sum_{k=1}^{K} I\left\{t_{j,k}^{**} \leq 2t_j^* - t\right\},$$

   where I is the indicator function;
5. Order vector u^* with J positions and obtain the lower and upper quantile of u^*, which are given, respectively by, $q_{\text{inf}} = u_{(J+1)*\alpha/2}^*$ and $q_{\text{sup}} = u_{(J+1)(1-\alpha/2)}^*$;
6. Order values $t_1^*, t_2^*, \ldots, t_J^*$ (i.e, $t_{(1)}^* \leq t_{(2)}^* \leq \cdots \leq t_{(J)}^*$) and build the confidence interval for the original sample using quantile estimates evaluated in the previous step. Considering the values of the ordered statistics t_j^*, for $j = 1, 2, \ldots, J$, the limits of the interval of level $1 - \alpha$ are given by

$$t_{((J+1)\alpha/2)}^*, \;\; t_{((J+1)(1-\alpha/2))}^*.$$

2.6 Simulations and hardware used

The function `pso` is computationally intensive, especially when the objective function involves some data sets. This situation is common when the goal is to maximize a log-likelihood function. The problem arises with great intensity when we use the `pso` function iteratively as in the case of Monte Carlo simulations. The simulations presented are extremely itensive because they will be studied using bootstrap percentile single and dual (two levels of bootstrap) to obtain interval estimates for the

NHG parameters. For the case of double bootstrap, in each replica Monte Carlo will have a bootstrap and in each bootstrap will be conducted another bootstrap render the simulations impractical in some hardware.

The simulations are performed using the language `R` version 3.2.0 in the hardware from the National Center for High Performance Processing of São Paulo, Brazil (CENAPAD-SP). The CENAPAD-SP provides a powerful computing environment, based on machines RISC (IBM) and Intel/Itanimum2 (SGI), with operating system based on Unix. The processing capacity theoretical of these two environments totals around 33 TFLOPS beyond of 250 TB external disk. Its use is measured in Service Units, what are accounted as users run commands and process jobs. In particular, the simulations performed make use of the SGI Altix ICE 8400 LX system installed in CENAPAD-SP that have 64 CPU's and 384 cores Intel Xeon Six Core 5680 of 3.33GHz, 1152 GB of RAM and Infiniband interconnect. The theoretical processing capability of the system is approximately 5 TFLOPS.

The jobs that are submitted for processing on SGI Altix ICE 8400 LX are managed and have allocated resources by the PBSPro software (Altair Portable Batch System Professional 11.0.2), which also has the function of managing the queues of jobs submitted to processing in the cluster. The cluster runs the opreracional system SUSE Linux Enterprise Server 11 SP1 (x86_64) kernel 2.6.32.13 64 bits. Access to the cluster is through a local machine using SSH (Secure Shell), which is part of the suite of protocols TCP/IP that makes secure remote administration of Unix-type servers.

It is possible to use in an efficient way all of the computing resources of SGI Altix ICE 8400 LX running parallel computing by OpenMPI (Message Passing Interface), which is nothing more than a standard for parallel computing in data communication. The use of standard OpenMPI in `R` is through the `Rmpi`, `SNOW`, `doSNOW` and `foreach` libraries available on the license terms GPL-2 (GNU General Public License).

This pattern allows using `R` by dividing each nonparametric bootstrap in several cores of different nodes in the cluster. Thus, the code executes sequentially until it finds a parallel builder, i.e., there is a flow of execution main (master thread) and when required new threads are triggered to divide the work and, finally, it is performed a join for recovery of the results. Divide each replica of bootstrap does not cause problems because each replica bootstarp is mathematically independent of another.

For the simulations we consider 10,000 replication of Monte Carlo, and in each replica it is carried out two levels of bootstrap percentile. The first level of bootstrap is composed of 500 samples ($J = 500$) and the second level is carried out 250 new samples ($K = 250$). The simulations are extremely intensive since for each Monte Carlo replication, we perform an optimization by PSO and within each replication of bootstrap (first level of bootstrap) another optimization is performed by PSO, closing a total of 5 million optimizations. Table 2.1 displays the times of the execution of `R` scripts, in hours, on hardware mentioned above. Note that for $n = 500$ whereas the double bootstrap scheme, some simulations exceeded 11 days.

The performance of interval estimates is evaluated by percentile bootstrap and double percentile bootstrap at a nominal level of 95% for different sample sizes ($n = 20$, $n = 60$, $n = 100$ and $n = 500$). The real model has fixed parameters $\alpha = 1.5$, $\lambda = 1.3$ and $p = 0.5$.

For the first level of bootstrap, we obtain the standard errors of the MLEs obtained by the PSO method. Small errors in the estimates are obtained for different sample sizes as shown in Figure 2.4. The performance of interval estimates by single and double bootstrap is assessed by the percentage of coverage, i.e., it is considered the percentage of confidence intervals containing the true parameter. It is noted that build interval estimates by single percentile bootstrap is not a good strategy for obtaining random intervals for the model parameters. In all cases there are much lower coverages to the nominal levels. However, using the second level of bootstrap to correct the estimate obtained by

simple percentile bootstrap method represents a good improvement in the interval estimates. Table 2.2 displays the improvement in coverage when using the double percentile bootstrap method. It is possible also note a small increase in the range of interval estimates by using the second level of bootstrap. Anyway, the small amplitudes do not represent a problem.

Table 2.1: Times of executions of Monte Carlo simulations to evaluate the interval estimates obtained by single and double percentile bootstrap.

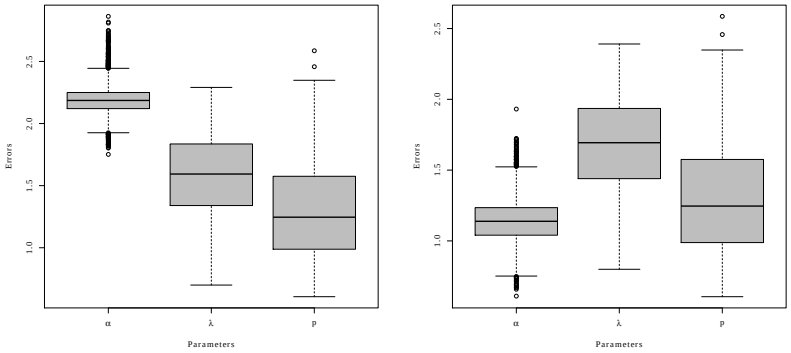
n	**Simple Bootstrap (In hours)**	**Double Bootstrap (In hours)**
20	105.3212	150.4235
60	175.3455	191.9864
100	213.7565	265.5455
500	400.3253	450.4535

Table 2.2: Coverage and range of interval estimates by simple bootstrap, double and nominal level of 95%.

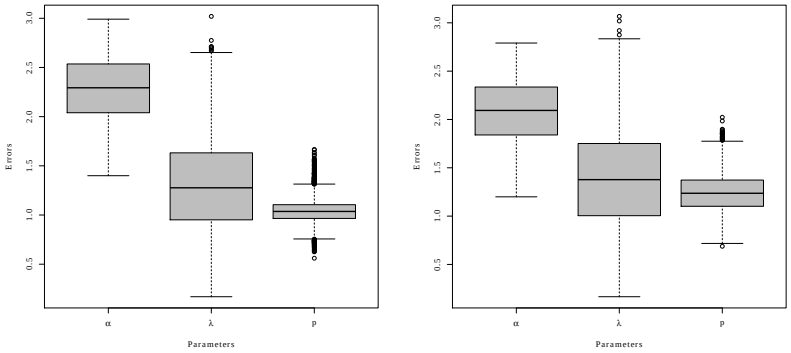
		Simple Bootstrap		**Double Bootstrap**	
n	**Parameters**	**Coverage (%)**	**Range**	**Coverage (%)**	**Range**
	α	65.0327	2.1562	92.3441	3.2432
$n = 20$	λ	62.1665	2.1605	93.4551	2.3423
	p	67.3002	0.9663	94.2138	1.3563
	α	71.1454	0.7446	93.5233	3.2124
$n = 60$	λ	77.7465	2.1454	92.3453	2.3567
	p	66.7213	1.2423	94.4354	2.8433
	α	72.2198	1.4354	92.4345	3.3315
$n = 100$	λ	69.2300	2.1322	94.2130	1.0035
	p	69.4357	0.5493	93.4365	2.1252
	α	83.1233	2.1215	94.7834	3.2123
$n = 500$	λ	71.9011	1.2321	94.3254	3.2122
	p	72.2965	1.1190	93.9342	1.2134

1 - The column for the range refers to the average range.

2 - We consider 10 thousand Monte Carlo replicas.



(a) Sample size 20 (b) Sample size 60



(c) Sample size 100 (d) Sample size 500

Figure 2.4: Errors evaluated by bootstrap the maximum likelihood using the PSO method with 500 bootstrap replicates.

2.7 Application

For this application, we use a real data set referred as the to times between failures (thousands of hours) of secondary reactor pumps. The data are presented in Table 2.3. A descriptive analysis of the data is given in Table 2.4.

There are certainly other competitive distributions to fit these data. It is also reasonable to understand that generated distributions can certainly be used in practice and depending on the problem a distribution will provide a better fit than others.

For this application, we consider the EXP(λ) distribution (exponential distribution), NHG(α, λ, p) distribution, NH(λ, β) distribution (Nadarajah-Haghighi model) proposed by Nadarajah and Haghighi (2011) and the ENG(α, λ, β) defined by Lemonte (2013). The cdfs are given in Table 2.5 presented more forward.

Table 2.3: Times between failures (thousands of hours) of secondary reactor pumps.

2.160	0.150	4.082	0.746	0.358	0.199	0.402	0.101	0.605	0.954
1.359	0.273	0.491	3.465	0.070	6.560	1.060	0.062	4.992	0.614
5.320	0.347	1.921							

Table 2.4: Descriptive statistics.

Statistics	**Real data sets**
	Time between failures (hours)
Mean	1.5779
Median	0.6140
Mode	0.5000
Variance	3.7275
Skewness	1.3643
Kurtosis	0.5445
Maximum	6.5600
Minimum	0.0620
n	23

In order to determine the shape of the most appropriate hazard function for modeling, a graphical analysis data can be performed. In this context, the total time in test (TTT) plot proposed by Aarset (1987) may be used. Let T be a random variable with non-negative values which represents the survival time. The TTT curve is obtained by constructing the plot of $G(r/n) = [(\sum_{i=1}^{r} T_{i:n}) + (n-r)T_{r:n}]/(\sum_{i=1}^{n} T_{i:n})$ versus r/n, for $r = 1, \ldots, n$, where $T_{i:n}$, $i = 1, \ldots, n$ are the order statistics of the sample [see Mudholkar and Hutson (1996)].

We shall use the programming language `R` version `3.0.2`. In particular, we use the `AdequacyModel` package in version `1.0.8` available for download at `http://cran.r-project.org/web/packages/AdequacyModel/index.html` under the terms of the GPL-2 and GPL-3. The plots can be easily obtained using the `TTT` function of the `AdequacyModel`. For more details on this function, see `help(TTT)`. The TTT plot for the current data is displayed in Figure 2.5, which alternates between convex and concave, suggesting according to Aarset (1987) that better risk function to the data in question has decreasing form.

Figure 2.6 displays the estimated distributions to the data obtained in a nonparametric manner using kernel density estimation with the Gaussian filter. Let $x_1, \ldots, x_n$ be independent and identically distributed observations, where each of them follows an unknown f distribution. The kernel density estimator is given by

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{2.18}$$

where $K(\cdot)$ is the kernel function usually symmetrical, $\int_{-\infty}^{\infty} K(x)dx = 1$, and $h > 0$ is a smoothing parameter known in the literature as bandwidth. Numerous kernel functions are available in the literature. The normal standard distribution is the most widely used because it has convenient mathematical properties. Silverman (1986) demonstrated that for the K standard normal, the bandwidth ideal is $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\,\hat{\sigma}\,n^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the sample.

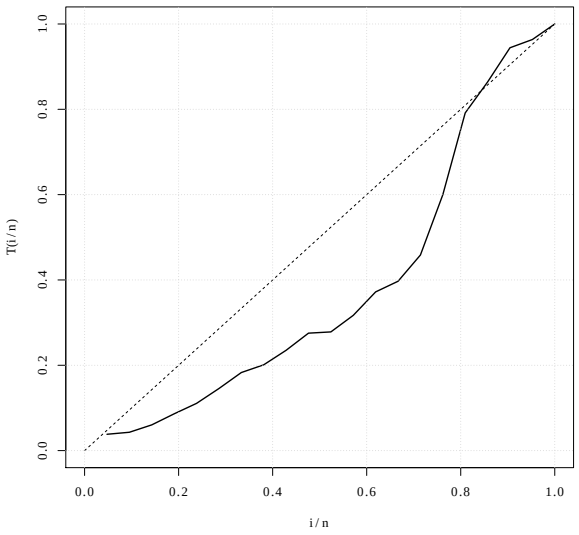


Figure 2.5: The TTT plot for the times between failures (thousands of hours) of secondary reactor pumps.

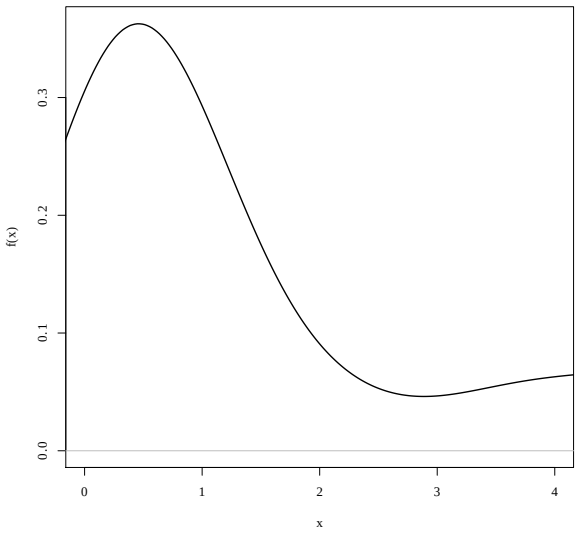


Figure 2.6: Gaussian kernel density estimation for the data of the time between failures (thousands of hours) of secondary reactor pumps.

The `AdequacyModel` package provides a computational support for researchers who want to work with continuos distributions, mainly in the survival analysis area. The `AdequacyModel` package is used to obtainment some adequacy statistics such as the AIC (Akaike Information Criterion), CAIC (Consistent Akaikes Information Criterion), BIC (Bayesian Information Criterion), HQIC (Hannan-Quinn information criterion), A^* (Anderson-Darling) and W^* (Camér-von Misses) and KS (Kolmogorov-Smirnov) statistics, which are described by Chen and Balakrishnan (1995). The

`goodness.fit` is the function in the `AdequacyModel` package used to avaluate these statistics. Other details can be obtained with the command `help(goodness.fit)`.

Chen and Balakrishnan (1995) constructed the Cramér-von Mises and Anderson-Darling corrected statistics based on the suggestions from Stephens (1986). We use these statistics, where we have a random sample $(x_1, \ldots, x_n)$ with empirical distribution function $F_n(x)$ and want to test if the sample comes from a special distribution. The statistics W^* and A^* are given by

$$W^* = \left\{ n \int_{-\infty}^{+\infty} \{F_n(x) - F(x; \widehat{\theta}_n)\}^2 dF(x; \widehat{\theta}_n) \right\} \left(1 + \frac{0.5}{n}\right) = W^2 \left(1 + \frac{0.5}{n}\right), \quad (2.19)$$

$$A^* = \left\{ n \int_{-\infty}^{+\infty} \frac{\{F_n(x) - F(x; \widehat{\theta}_n)\}^2}{\{F(x; \widehat{\theta})(1 - F(x; \widehat{\theta}_n))\}} dF(x; \widehat{\theta}_n) \right\} \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right)$$

$$= A^2 \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right), \quad (2.20)$$

respectively, where $F_n(x)$ is the empirical distribution function, $F(x; \hat{\theta}_n)$ is the specified distribution function evaluated at the MLE $\hat{\theta}_n$ of θ. Note that the statistics W^* and A^* are given by difference distances of $F_n(x)$ and $F(x; \hat{\theta}_n)$. Thus, the lower are the W^* and A^* statistics more evidence we have that $F(x; \hat{\theta}_n)$ generates the sample. The details to compute the statistics W^* and A^* are given by Chen and Balakrishnan.

The `R` language is also used to obtain the MLEs by heuristic method of global optimization PSO discussed in the previous section. Currently, the version `1.0.8` of the `AdequacyModel` package does not support the methodology PSO. However, possibly the standard algorithm or a modification will be present in future versions of the package. The standard errors of the MLEs of the model parameters are calculated by bootstrap non-parametric. For the calculation, we consider 500 bootstrap samples ($B = 500$) [see Davison (1997) and Efron and Tibshirani (1993)].

Let $x_1, \ldots, x_n$ be a random sample. Let also $F_\theta(x)$ be the distribution function of the sample, where θ (unknown) is the true parameter and $\hat{\theta}$ a estimator of θ. A bootstrap sample (non-parametric bootstrap) is obtained sampling with replacement from the original sample. The procedure generates a new sample $(x_1^*, \ldots, x_n^*)$ (bootstrap sample). Let $\hat{\theta}^*$ be the estimate of θ based on the bootstrap sample and B be the fixed number of bootstrap samples. The bootstrap estimate of the standard error of $\hat{\theta}$ is given by

$$\widehat{\text{se}}(\hat{\theta}) = \sqrt{\frac{1}{B} \sum_{i=1}^{B} (\hat{\theta}_i^* - \bar{\hat{\theta}}^*)} \ ,$$

where

$$\bar{\hat{\theta}}^* = \frac{1}{B} \sum_{i=1}^{B} \hat{\theta}^*.$$

Table 2.5 presents the MLEs of the parameters of the distributions obtained by the PSO method. They are also given for the estimated standard errors obtained via nonparametric bootstrap. Also, we construed interval estimates by double percentile bootstrap for the model parameters. The estimates are presented in Table 2.6 for a confidence level of 95%. We consider 500 samples for the first level and 250 bootstrap samples for the second level.

Table 2.5: The MLEs of the parameters for the NHG, ENH, NH and EXP models through of the PSO method (standard errors bootstrap in parentheses).

Distributions	**Estimates of the parameters**			**Equation**
NHG(α, λ, p)	0.4195	5.7294	-0.7929	$F(x) = \frac{1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}}{1-p\,\mathrm{e}^{1-(1+\lambda x)^{\alpha}}}$
	(0.0732)	(0.1230)	(0.0897)	
ENH(α, λ, β)	0.2856	42.0728	3.1569	$F(x) = \left[1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}\right]^{\beta}$
	(0.1125)	(0.1323)	(0.3774)	
NH(α, λ)	0.4934	2.5010		$F(x) = 1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}$
	(0.1281)	(0.2744)		
EXP(λ)	0.6172			$F(x) = 1-\mathrm{e}^{-\lambda x}$
	(0.1081)			

1 - We use rounding to the fourth decimal place.

Table 2.6: Interval estimates by double percentile bootstrap for the model parameters at a confidence level of 95%.

Distributions	**Interval estimates**		
NHG(α, λ, p)	(0.3578; 0.8177)	(0.8884; 13.8570)	$(-3.7684; -0.0001)$
ENH(α, λ, β)	(0.2621; 0.7341)	(0.8400; 49.9999)	(0.9062; 5.8346)
NH(α, λ)	(0.3804; 1.4111)	(0.3665; 5.9659)	
EXP(λ)	(0.4228; 1.1551)		

1 - We use rounding to the fourth decimal place.

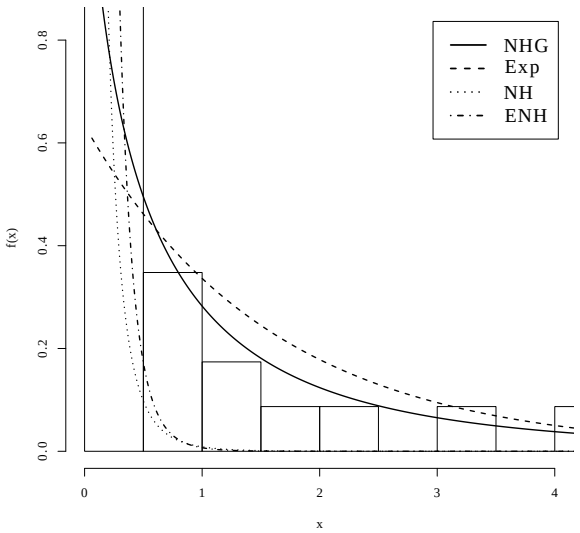


Figure 2.7: Estimated for densities fitted to the times between failures (thousands of hours) of secondary reactor pumps.

Table 2.7: Goodness-of-fit statistics.

	Statistics						
Distribution	**AIC**	**CAIC**	**BIC**	**HQIC**	**KS**	**A***	**W***
NHG(α, λ, p)	72.8293	74.0925	76.2358	73.6860	**0.1684**	**0.2547**	**0.0323**
ENH(α, λ, β)	289.7261	290.9892	293.1326	290.5828	0.6630	0.3020	0.0424
NH(α, λ)	239.9434	240.5434	242.2144	240.5146	0.7119	0.3546	0.0513
EXP(λ)	168.9795	169.1700	170.1150	169.2650	0.1994	0.5069	0.0795

1 - The statistics were obtained in the package `AdequacyModel` version `1.0.8`.
2 - Was used rounding to the fourth decimal place.

2.8 Concluding remarks

We study the Nadarajah-Haghighi-geometric (NHG) distribution, which can be viewed as an improved extension of the Nadarajah-Haghighi distribution (NH distribution). The NHG density function can take various forms depending on its shape parameters. Moreover, its failure rate function can have different forms: decreasing, increasing, upside-down bathtub, bathtub-shaped, constant and decreasing-increasing-decreasing. Then, it can be used quite effectively in analyzing real data in practice. The estimation of parameters is approached by the maximum likelihood method. The new distribution may attract wider applications for modeling failure times due to fatigue and lifetime data in fields such as engineering, finance, economics, and insurance, among several others. One applications to real data illustrate the potentiality of the family. Future research could be addressed to study the complementary Nadarajah-Haghighi-geometric distribution. This class of distributions is a suitable model in a complementary risk, problem based in the presence of latent risks which arise in several areas such as public health, actuarial science, biomedical studies, demography and industrial reliability.

APPENDIX A

Code in R language for the PSO method

```
pso <- function(func,S=150,lim_inf,lim_sup,e=0.0001,data=NULL,N=100){
b_lo = min(lim_inf)
b_up = max(lim_sup)
integer_max = .Machine$integer.max

if(length(lim_sup)!=length(lim_inf)){
stop("The vectors lim_inf and lim_sup must have the same dimension.")
}
dimension = length(lim_sup)
swarm_xi = swarm_pi = swarm_vi = matrix(NA,nrow=S,ncol=dimension)

# The best position of the particles.
g = runif(n=dimension,min=lim_inf,max=lim_sup)

# Objective function calculated in g.
f_g = func(par=as.vector(g),x=as.vector(data))

if(NaN%in%f_g==TRUE || Inf%in%abs(f_g)==TRUE){
while(NaN%in%f_g==TRUE || Inf%in%abs(f_g)==TRUE){
g = runif(n=dimension,min=lim_inf,max=lim_sup)
f_g = func(par=g,x=as.vector(data))
}
}

# Here begins initialization of the algorithm.
x_i = mapply(runif,n=S,min=lim_inf,max=lim_sup)

# Initializing the best position of particularities i to initial position.

swarm_pi = swarm_xi = x_i
f_pi = apply(X=x_i,MARGIN=1,FUN=func,x=as.vector(data))

is.integer0 <- function(x){
```

```
is.integer(x) && length(x)==0L
}

if(NaN%in%f_pi==TRUE || Inf%in%abs(f_pi)){
while(NaN%in%f_pi==TRUE || Inf%in%abs(f_pi)){
id_inf_fpi = which(abs(f_pi)==Inf)
if(is.integer0(id_inf_fpi)!=TRUE){
f_pi[id_inf_fpi] = integer_max
}
id_nan_fpi = which(f_pi==NaN)
if(is.integer0(id_nan_fpi)!=TRUE){
x_i[id_nan_fpi,] = mapply(runif,n=length(id_nan_fpi),min=lim_inf,
max=lim_sup)
swarm_pi = swarm_xi = x_i
f_pi = apply(X=x_i,MARGIN=1,FUN=func,x=as.vector(data))
}
}
}

minimo_fpi = min(f_pi)
if(minimo_fpi < f_g) g = x_i[which.min(f_pi),]

# Initializing the speeds of the particles.

swarm_vi = mapply(runif,n=S,min=-abs(rep(abs(b_up-b_lo),dimension)),
max=abs(rep(abs(b_up-b_lo),dimension)))

# Here ends the initialization of the algorithm

omega = 0.5
phi_p = 0.5
phi_g = 0.5

m=1
vector_f_g <- vector()

while(is.na(var(vector_f_g)) || m<50 ||
var(vector_f_g[length(vector_f_g):(length(vector_f_g)-10)])>e){
# r_p and r_g are randomized numbers in (0.1).
r_p = runif(n=dimension,min=0,max=1)
r_g = runif(n=dimension,min=0,max=1)

# Updating the vector speed.
swarm_vi = omega*swarm_vi+phi_p*r_p*(swarm_pi-swarm_xi)+
phi_g*r_g*(g-swarm_xi)
```

```
# Updating the position of each particle.
swarm_xi = swarm_xi+swarm_vi

myoptim = function(...) tryCatch(optim(...), error = function(e) NA)

f_xi = apply(X=swarm_xi,MARGIN=1,FUN=func,x=as.vector(data))
f_pi = apply(X=swarm_pi,MARGIN=1,FUN=func,x=as.vector(data))
f_g =  func(par=g,x=as.vector(data))

if(NaN%in%f_xi==TRUE || NaN%in%f_pi==TRUE){
while(NaN%in%f_xi==TRUE){
id_comb = c(which(is.na(f_xi)==TRUE),which(is.na(f_pi)==TRUE))
if(is.integer0(id_comb)!=TRUE){
new_xi = mapply(runif,n=length(id_comb),min=lim_inf,
max=lim_sup)
swarm_pi[id_comb,]=swarm_xi[id_comb,] = new_xi
if(length(id_comb)>1){
if_xi[id_comb] = apply(X=swarm_xi[id_comb,],MARGIN=1,
FUN=func,x=as.vector(data))
f_pi[id_comb] = apply(X=swarm_pi[id_comb,],MARGIN=1,FUN=func,
x=as.vector(data))
}else{
f_xi[id_comb] = func(par=new_xi,x=as.vector(data))
}
}
}
}

if(Inf%in%abs(f_xi)==TRUE){
f_xi[which(is.infinite(f_xi))]=integer_max
}
if(Inf%in%abs(f_pi)==TRUE){
f_pi[which(is.infinite(f_pi))]=integer_max
}

# There are values below the lower limit of restrictions?
id_test_inf=
which(apply(swarm_xi<t(matrix(rep(lim_inf,S),dimension,S)),1,sum)>=1)
id_test_sup=
which(apply(swarm_xi>t(matrix(rep(lim_sup,S),dimension,S)),1,sum)>=1)

if(is.integer0(id_test_inf)!=TRUE){
swarm_pi[id_test_inf,] = swarm_xi[id_test_inf,] =
mapply(runif,n=length(id_test_inf),
min=lim_inf,max=lim_sup)
```

```
}

if(is.integer0(id_test_sup)!=TRUE){
swarm_pi[id_test_sup,] = swarm_xi[id_test_sup,] =
mapply(runif,n=length(id_test_sup),
min=lim_inf,max=lim_sup)
}

if(is.integer0(which((f_xi<=f_pi)==TRUE))){
swarm_pi[which((f_xi<=f_pi)),] = swarm_pi[which((f_xi<=f_pi)),]
}

if(f_xi[which.min(f_xi)] <= f_pi[which.min(f_pi)]){
swarm_pi[which.min(f_pi),] = swarm_xi[which.min(f_xi),]
if(f_pi[which.min(f_pi)] < f_g) g = swarm_pi[which.min(f_pi),]
} # Here ends the block if.

vector_f_g[m] = f_g
m = m+1
if(m>N){
break
}

} # Here ends the block while.

f_x = apply(X=swarm_xi,MARGIN=1,FUN=func,x=as.vector(data))
list(par_pso=g,f_pso=vector_f_g)

} # Here ends the function.
```

References

[1] Aarset, M V. How to identify a bathtub hazard rate. *IEEE Transactions on Reliability*, v. 36, p. 106–108, 1987.

[2] Adamidis, K, and Loukas, S. A lifetime distribution with decreasing failure rate. *Statistics & Probability Letters*, v. 39, p. 35–42, 1998.

[3] Adamidis, K, Dimitrakopoulou, T, and Loukas, S. On an extension of the exponential-geometric distribution. *Statistics & probability letters*, v. 73, p. 259–269, 2005.

[4] Bourguignon, M, Silva, R B and Cordeiro, G M. A new class of fatigue life distributions. *Journal of Statistical Computation and Simulation*, v. 84, p. 2619–2635, 2014.

[5] Basu, A P and Klein, J P. Some recent results in competing risks theory. *Lecture Notes-Monograph Series*, v. 2, p. 216–229, 1982.

[6] Barreto-Souza, Wagner and Santos, HS, Alessandro and Cordeiro, Gauss M. *Journal of Statistical Computation and Simulation*, v. 80, p. 159–172, 2010.

[7] Chahkandi, M, and Ganjali, M. On some lifetime distributions with decreasing failure rate. *Computational Statistics & Data Analysis*, v. 53, p. 4433–4440, 2009.

[8] Chen, G, and Balakrishnan, N. A general purpose approximate goodness-of-fit test. *Journal of Quality Technology*. v. 27. p. 154–161, 1995.

[9] Cordeiro, Gauss M, and de Castro, Mário. A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, v. 81, p. 883–898, 2011.

[10] Cordeiro, Gauss M, Ortega, Edwin MM, and Nadarajah, Saralees. The Kumaraswamy Weibull distribution with application to failure data. *Journal of the Franklin Institute*, v. 347, p. 1399–1429, 2010.

[11] Davison, A C, and Hinkley, D V. Bootstrap methods and their application. v. 1, Cambridge University Press, 1997.

[12] Dempster, A P, Laird, N M and Rubin, Donald B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, v. 39, p. 1–38, 1977.

[13] Efron, B. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*. v. 7, p. 1–26, 1979.

[14] Eugene, Nicholas, Lee, Carl, and Famoye, Felix. Beta-normal distribution and its applications, *Communications in Statistics - Theory and Methods*, v. 31, p. 497–512, 2002.

[15] Efron, B and Tibshirani, Robert. An introduction to the bootstrap. v. 57, Ed. 1, CRC Press, New York, 1993.

[16] Gompertz, Benjamin. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philosophical transactions of the Royal Society of London*, v. 115, p. 513–583, 1825.

[17] Gupta, Ramesh C., Pushpa L. Gupta, and Rameshwar D. Gupta. Modeling failure time data by Lehman alternatives. *Communications in Statistics - Theory and Methods*, v. 27, p. 887–904, 1998.

[18] Gomes, Antonio E, Silva, Cibele Q, Cordeiro, Gauss M, and Ortega, Edwin MM. A new lifetime model: the Kumaraswamy generalized Rayleigh distribution. *Journal of Statistical Computation and Simulation*, v. 84, p. 290–309, 2014.

[19] Kennedy, J, Kennedy, J F and Eberhart, R C. Swarm intelligence, Ed. Morgan Kaufmann, 2001.

[20] Lee, Carl, Famoye, Felix, and Olumolade, Olugbenga. Beta-Weibull distribution: some properties and applications to censored data. *Journal of modern applied statistical methods*. v. 6, p. 17, 2007.

[21] Lemonte, A J. A new exponential-type distribution with constant, decreasing, increasing, upside-down bathtub and bathtub-shaped failure rate function. *Computational Statistics & Data Analysis*. v. 62, p. 149–170, 2013.

[22] Mahmoudi, E, and Jafari, AA. Generalized exponential power series distributions. *Computational Statistics & Data Analysis*. v. 56, p. 4047–4066, 2012.

[23] Marshall, Albert W., and Olkin, Ingram. A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. *Biometrika*, v. 84, p. 641–652, 1997.

[24] Matsumoto, M, and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, v. 8, p. 3–30, 1998.

[25] Morais, AL, and Barreto-Souza, W. A compound class of Weibull and power series distributions. *Computational Statistics & Data Analysis*, v. 55, p. 1410–1425, 2011.

[26] Mudholkar, G S and Hutson, A D. The exponentiated Weibull family: some properties and a flood data application. *Communications in Statistics - Theory and Methods*, v. 25, p. 3059–3083, 1996.

[27] Nadarajah, Saralees; Haghighi, Firoozeh. An extension of the exponential distribution. *Statistics*, v. 45, n. 6, p. 543–558, 2011.

[28] Nadarajah, Saralees and Kotz, Samuel. *Reliability engineering and system safety*. v. 91, p. 689–697, 2006.

[29] Shi, Y and Eberhart, R. Evolutionary Computation Proceedings, 1998. *IEEE World Congress on Computational Intelligence*. IEEE International Conference on, p. 69-73, 1998.

[30] Silva, Giovana O, Ortega, Edwin MM, and Cordeiro, Gauss M. The beta modified Weibull distribution. *Lifetime Data Analysis*, v. 16, p. 409–430, 2010.

[31] Silva, R B, Bourguignon, M, Dias, C R B, and Cordeiro, G M. The compound class of extended Weibull power series distributions. *Computational Statistics & Data Analysis*, v. 58, p. 352–367, 2013.

[32] Silva, R B, and Gauss M C. The Burr XII power series distributions: A new compounding family. *Brazilian Journal of Probability and Statistics*, v. 29, p. 565–589, 2015.

[33] Stephens, M A. Goodness-of-Fit Techniques, RB D'Agostino and MS Stephens, Marcel Dekker, New York and Basel, p. 331–366, 1986.

CHAPTER 3

The exponentiated logarithmic generated family of distributions

Abstract: Generalized distributions are very common in statistics and in many applied areas. In this paper, we study some mathematical properties of a new generator of continuous distributions with three additional parameters called the exponentiated logarithmic generated family to extend the normal, Weibull, gamma and Gumbel distributions, among other well-known distributions. Some special models are discussed. Seveal properties of this family are studied, some inference procedures developed and simulation studies performed to verify the adequacy of the estimators of the model parameters. We prove empirically the potentiality of the new family by means of two real data sets. We also provide a simulation study for different samples sizes to assess the performance of the maximum likelihood estimates obtained by the Swarm Optimization method. We also evaluate the performance of single and dual bootstrap methods in the construction of interval estimates for the parameters. Because of the intensive simulations, we use parallel computing on a supercomputer.

Keywords: Bootstrap. Generalized distribution. Lifetime. Logarithmic distribution. Mixture.

3.1 Introduction

The modeling and analysis of lifetime distributions play an important role in a wide variety of practical fields such as biological and engineering sciences. However, in many practical situations, well-known continuous distributions do not provide an adequate fit. For example, if the data are asymmetric, the normal distribution will not be a good choice. So, several methods of introducing extra shape parameters to expand a class of distributions have been studied.

The use of new generators of continuous distributions from classic ones has become very common in recent years. The beta-generated family was proposed by Eugene et al. (2002) and further discussed in Zografos and Balakrishnan (2009), who introduced the gamma-generated family. More recently, Cordeiro and de Castro (2011) defined the Kumaraswamy-G family.

The chief motivation of the generalized distributions for modeling failure time data lies in its flexibility to model both monotonic and non-monotonic failure rates even though the baseline failure rate may be monotonic. The additional shape parameters aim to introduce skewness and to vary tail weights. Furthermore, various distributions have been constructed by mixing some useful life distributions and analyzed them with respect to different characteristics.

In this chapter, we propose a new method to add three parameters to a parent distribution with the hope that it yields a "better" fit in certain practical situations. Several properties are also given.

Some inferential aspects of this family are studied in details, and four special cases are discussed. The new family of distributions shares an attactive interpretation (see Section 2). Further, two successful empirical applications show its flexibility and also motivate its applications.

3.2 New Generator

For an interpretation of this family, first consider a parallel system with N independent components, N be a random variable with probability mass function (pmf)

$$P(N=n) = \frac{-1}{\log(1-a)} \times \frac{a^n}{n} \quad 0 < a < 1, n \in \mathbb{N}.$$

Suppose that $X_1, \ldots, X_N$ are independent identically distributed (i.i.d.) random variables with common cdf $G(x)^b$. Then, $M_N = \max(X_1, \ldots, X_N)$ represents the lifetime of the system and

$$\begin{aligned} \Pi_{M_N}(x) &= \sum_{n=1}^{\infty} P(M_N \leq x | N=n) P(N=n) \\ &= \sum_{n=1}^{\infty} \frac{-1}{\log(1-a)} \frac{a^n}{n} \left[G(x)^b\right]^n = \frac{\log\left[1 - a\, G(x)^b\right]}{\log(1-a)}. \end{aligned}$$

For c positive integer, we consider a system formed by independent components following the cdf given by

$$\Pi(x) = \frac{\log\left[1 - a\, G(x)^b\right]}{\log(1-a)}.$$

Suppose the system fails if all c components fail. Then, the cdf of X is (3.1). The generator of continuous distributions presented is called the *Exponentiated Logarithmic Generated* (ELG) family with cumulative distribution function (cdf) given by

$$F(x) = \left\{ \frac{\log[1 - a\, G(x,\xi)^b]}{\log(1-a)} \right\}^c, \tag{3.1}$$

where $a \in (0,1)$ (scale parameter), $b > 0$ (shape parameter) and $c > 0$ (shape parameter). The probability density function (pdf) obtained by differentiating $F(x)$ is given by

$$f(x) = \frac{a\, b\, c \left[\log(1-a)\right]^{-c} g(x,\xi) G(x,\xi)^{b-1} \left\{\log\left[1 - a\, G(x,\xi)^b\right]\right\}^{c-1}}{a\, G(x,\xi)^b - 1}. \tag{3.2}$$

For a given (cdf) $G(x;\xi)$, denote by $f(x)$ the (pdf) of the *Exponentiated Logarithmic Generated* (ELG-G) distribution.

Lemma 3.2.1. *$f(x)$ given in (3.2) is a well-defined density function.*

Proof. We observe that $f(x)$ is nonnegative. We prove that the integration over the support of the random variable is one. Consider the case when the support of $f(x)$ is $(-\infty, \infty)$. In this case, we

have

$$\begin{aligned}\int_{-\infty}^{\infty} f(x)dx &= \int_{-\infty}^{\infty} \frac{-a\,b\,c\,[\log(1-a)]^{-c}\,g(x,\xi)G(x,\xi)^{b-1}\left\{\log\left[1-a\,G(x,\xi)^b\right]\right\}^{c-1}}{1-a\,G(x)^b}dx \\ &= \frac{-c}{[\log(1-a)]^c}\int_0^a \frac{[\log(1-u)]^{c-1}}{1-u}du = 1.\end{aligned}$$

□

For simulating the data from the ELG-G distribution, let $u \sim U(0,1)$ [u generate the uniform distribution in (0,1)]. The solution of non-linear equation

$$x_u = Q_G\left\{\left[\frac{1}{a}\left(1-[1-a]^{u^{\frac{1}{c}}}\right)\right]^{\frac{1}{b}}\right\} \tag{3.3}$$

has cdf (3.1), where $Q_G = G^{-1}(\cdot)$ is the quantile function (qf) of G.

Remark. *The following properties use Stirling polynomials. We consider the basic formula for the Stirling polynomial proposed by Ward (1934, p. 87-95). The notation for the Stirling polynomial adopted is $\psi_{n-1}(x)$ in accordance with the notation presented by Nielsen (1906, p. 70-72) and Ward (1934, p. 87). The Stirling polynomial is defined by*

$$\begin{aligned}\psi_{n-1}(x) &= \frac{(-1)^{n-1}}{(n+1)!}\left[H_n^{n-1} - \frac{x+2}{n+2}H_n^{n-2} + \frac{(x+2)(x+3)}{(n+2)(n+3)}H_n^{n-3}\right. \\ &- \left. \cdots + (-1)^{n-1}\frac{(x+2)(x+3)\cdots(x+n)}{(n+2)(n+3)\cdots(2n)}H_n^0\right],\end{aligned} \tag{3.4}$$

where the H_n^m's are positive integers defined recursively by $H_{n+1}^m = (2n+1-m)H_n^m + (n-m+1)H_n^{m-1}$, and $H_0^0 = 1$, $H_{n+1}^0 = 1 \times 3 \times 5 \times \cdots \times (2n+1)$ and $H_{n+1}^n = 1$.

To avoid recursion in equation (3.4), the quantities H_{n+1}^m can follow the Stirling polynomial given in `http://mathworld.wolfram.com/StirlingPolynomial.html`. *Let*

$$S_n(m) = \frac{(-1)^n}{\binom{m}{n}}s(m+1, m-n+1), \tag{3.5}$$

where $m \geq n$, and $s(n,m)$ is the Stirling number of the first kind defined by Roman (1984, p. 129). We can obtain $s(n,m)$ from the Stirling numbers of the second kind defined by

$$S(n,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^i\binom{m}{i}(n-i)^n. \tag{3.6}$$

Based on (3.6), we have

$$s(n,m) = \sum_{k=0}^{n-m}(-1)^k\binom{k+n-1}{k+n-m}\binom{2n-m}{n-k-m}S(k-m+n,k). \tag{3.7}$$

More details about equation (3.7) and the relationship between first order Stirling numbers and second order Stirling numbers can be obtained in `http://mathworld.wolfram.com/StirlingNumberoftheFirstKind.html`.

According with Castellares and Lemonte (2014, p. 2), we have $\psi_{n-1}(x) = S_n(x)/[n!(x+1)]$. *So, we can calculate* $\psi_{n-1}(x)$ *without explicitly obtain the values of* H_n^m. *Castellares and Lemonte (2014, p. 5) developed scripts in the* `R` *language to evaluate the quantities* H_m^n *and* $\psi_n(\cdot)$. *We give in Appendix A an implemented function in the* `Julia` *programming language (Bezanzon et al., 2012) to evaluate the Stirling polynomial at the point* x *defined by (3.4) and functions of the Stirling numbers of first and second order given by equations (3.5) and (3.6), respectively.*

Proposition 3.2.2. *Let* $\left[-\frac{\log(1-z)}{z}\right]^\delta = 1 + \delta z \sum_{n=0}^{\infty} \psi_n(n+\delta) z^n$, *with* $\delta \in \mathbb{R}$ *and* $|z| < 1$. *This expansion is absolutely convergent.*

Proof. The proof is given by Flajonet and Odlyzko (1990) and Flajonet and Sedgewick (2009) (see Theorem VI.2, p. 385). □

If we know an expansion for $\left[-\frac{\log(1-z)}{z}\right]^\delta$ for $\delta > 0$ and $|z| < 1$, and that this expansion converges absolutely, it is easy to obtain another expansion for $[-\log(1-z)]^\delta$ that is also absolutely convergent. Thus, it follows that

$$[-\log(1-z)]^\delta = z^\delta + \delta \sum_{n=0}^{\infty} \psi_n(n+\delta) z^{\delta+n+1}. \tag{3.8}$$

Proposition 3.2.3. *Based on the assumptions below, it is possible to obtain equation (3.8). Let*

$$[-\log(1-z)]^\delta = z^\delta \sum_{m=0}^{\infty} \rho_m(\delta)\, z^m, \tag{3.9}$$

where $\delta \in \mathbb{R}$, $|z| < 1$, $\rho_0(\delta) = 1$, $\rho_m(\delta) = \delta\psi_{m-1}(m+\delta-1)$ *for* $m \geq 1$ *and* $\psi_m(\cdot)$ *is a Stirling polynomial.*

Proof. Note that

$$\begin{aligned} z^\delta \sum_{m=0}^{\infty} \rho_m(\delta) z^m &= z^\delta \left[\rho_0(\delta) z^0 + \sum_{m=1}^{\infty} \rho_m(\delta) z^m\right] \\ = \quad z^\delta + \sum_{m=1}^{\infty} \rho_m(\delta)\, z^{m+\delta} &= z^\delta + \delta \sum_{m=0}^{\infty} \psi_m(m+\delta)\, z^{\delta+m+1}. \end{aligned}$$

□

Thus, it follows that the cdf $F(x)$ can be expressed as a mixture of exponentiated-G (exp-G) cdfs with power parameters $(m+c)b$, say exp-G$([m+c]b)$. We have

$$F(x) = \sum_{m=0}^{\infty} b_m\, H_{(m+c)b}(x), \tag{3.10}$$

where $H_{(m+c)b}(x)$ is the cdf of the exp-G$([m+c]b)$ distribution and b_m (for $m \geq 0$) are constants defined by

$$b_m = \frac{\rho_m(c)a^{m+c}}{[-\log(1-a)]^c}.$$

By differentiating (3.10), we obtain

$$f(x) = \sum_{m=0}^{\infty} b_m \, h_{(m+c)b}, \tag{3.11}$$

where $h_{(m+c)b}(x)$ is the exp-G density with power parameter $(m+c)b$. We have $\sum_{m=0}^{\infty} b_m = 1$. Figure 3.1 displays the convergence of $\sum_{m=0}^{n} b_m$ with $n = 1, 2, \ldots, 15$, $a = 0.5$ and $c = 1.2$.

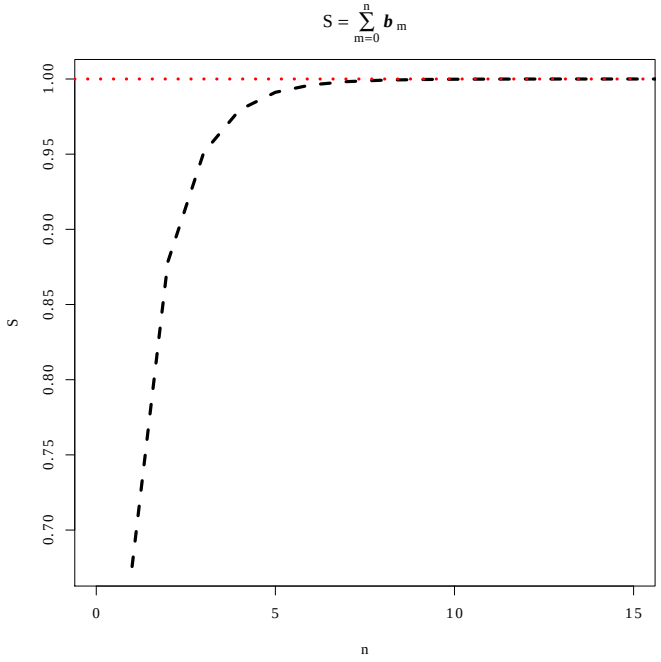


Figure 3.1: Sums of the constants of the linear combination in equation (3.10).

3.3 Special distributions

Four of the many distributions obtained as special models of the ELG-G family are given in this section. We consider the baseline distributions: Normal, Weibull, Gamma and Log-Logistic. The last three distributions are largely used in survival analysis. In the following, we shall provide the pdf and some plots of the hazard rate function (hrf) for each of these four cases. Clearly, a, b and c are the generator-G parameters.

3.3.1 ELG-Normal

In applied statistics, the normal distribution is widely used because when the sample size is large, it can serve as an approximate distribution for other models. The ELG-normal distribution is defined

from (3.1) by taking $G(x)$ and $g(x)$ to be the cdf and pdf of the normal $\mathrm{N}(\mu, \sigma^2)$ distribution. Its density function is given by

$$f_{GN}(x) = \frac{-a\,b\,c\,[\log(1-a)]^{-c}\,\Phi(\frac{x-\mu}{\sigma})^{b-1}\phi(\frac{x-\mu}{\sigma})\left[\log\left(1-a\Phi(\frac{x-\mu}{\sigma})^{b}\right)\right]^{c-1}}{\sigma\left\{1-a\,\Phi(\frac{x-\mu}{\sigma})^{b}\right\}},$$

where $x \in \mathbb{R}, \mu \in \mathbb{R}$ is a location parameter, $\sigma > 0$ and $a \in (0,1)$ are the scale parameters, b, c are the shape parameters, and $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of the standard normal distribution, respectively.

Plots of the ELG-normal density for some parameter values are displayed in Figure 3.2. It is evident that this distribution is much more flexible than the normal distribution.

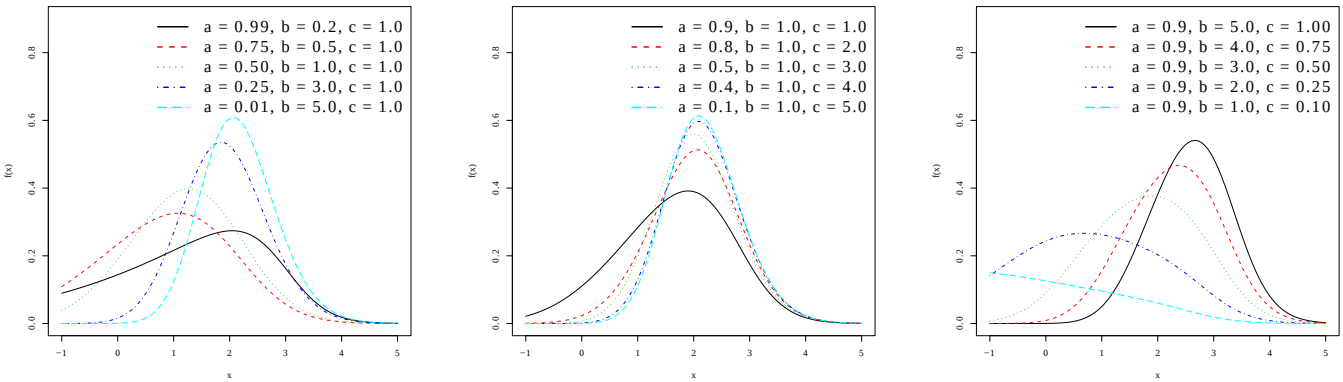


Figure 3.2: The ELG-normal pdf for $\mu = 1$, $\sigma = 1$ and some values of a, b and c.

3.3.2 ELG-Weibull

The Weibull distribution is a very popular model due to its large applicability in survival analysis. Let $G(x)$ be the Weibull cdf with scale parameter $\beta > 0$ and shape parameter $\alpha > 0$, say $G(x) = 1 - \exp\{-(\beta x)^{\alpha}\}$, for $x > 0$. The ELG-Weibull density function is obtained from (3.2) as

$$f_{ELG-W}(x) = \frac{-a\,b\,c\,[1-\mathrm{e}^{-(x\beta)^{\alpha}}]^{b-1}\alpha\beta^{\alpha}\,x^{\alpha-1}\,\mathrm{e}^{-(x\beta)^{\alpha}}\,\{\log\{1-a[1-\mathrm{e}^{-(x\beta)^{\alpha}}]^{b}\}\}^{c-1}}{[\log(1-a)]^{c}\,\{1-a[1-\mathrm{e}^{-(x\beta)^{\alpha}}]^{b}\}}. \tag{3.12}$$

Plots of the ELG-Weibull density function for selected parameter values are given in Figure 3.3.

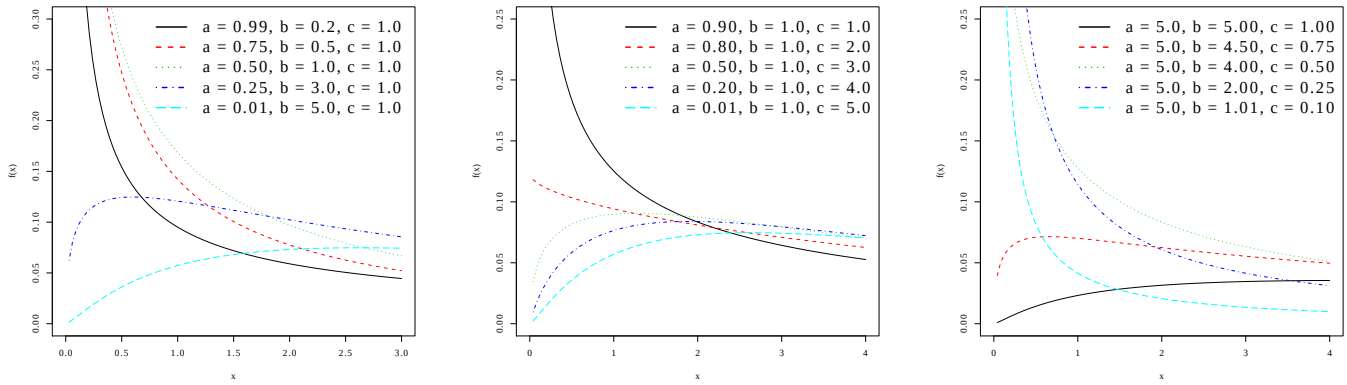


Figure 3.3: The ELG-Weibull pdf for $\alpha = 0.5$, $\beta = 2.0$ and some values of a, b and c.

Figure 3.4 illustrates some possible shapes of the ELG-Weibull hrf for selected parameter values.

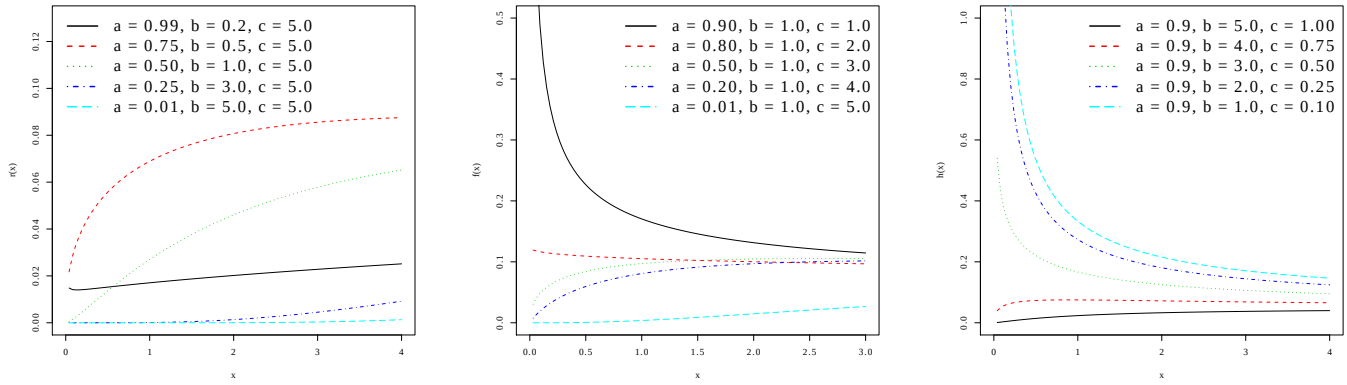


Figure 3.4: The ELG-Weibull hrf for $\alpha = 0.5$, $\beta = 2.0$ and some values of a, b and c.

3.3.3 ELG-Gamma

Another distribution that frequently used in survival analysis is the gamma distribution. Taking $G(x)$ to be gamma cdf with shape $\alpha > 0$ and scale $\beta > 0$, say $G(x) = \gamma(\alpha, x/\beta)/\Gamma(\alpha)$, where $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}\mathrm{e}^{-t}\mathrm{d}t$ denotes the gamma function, and $\gamma(\alpha, z) = \int_0^z t^{\alpha-1}\mathrm{e}^{-t}\mathrm{d}t$ denotes the incomplete gamma function, the ELG-gamma density (for $x > 0$) becomes

$$f_{ELG-Ga}(x) = \frac{-a\,b\,c\,x^{\alpha-1}\,\mathrm{e}^{-x/\beta}\left[\frac{\gamma(\alpha,x/\beta)}{\Gamma(\alpha)}\right]^{b-1}\left\{\log\left\{1-a\left[\frac{\gamma(\alpha,x/\beta)}{\Gamma(\alpha)}\right]^{b}\right\}\right\}^{c-1}}{\beta^{\alpha}\,\Gamma(\alpha)\,[\log(1-a)]^{c}\left\{1-a\left[\frac{\gamma(\alpha,x/\beta)}{\Gamma(\alpha)}\right]^{b}\right\}}. \tag{3.13}$$

Figure 3.5 displays some possible shapes of the ELG-gamma density. These plots reveal that this distribution has great flexibility.

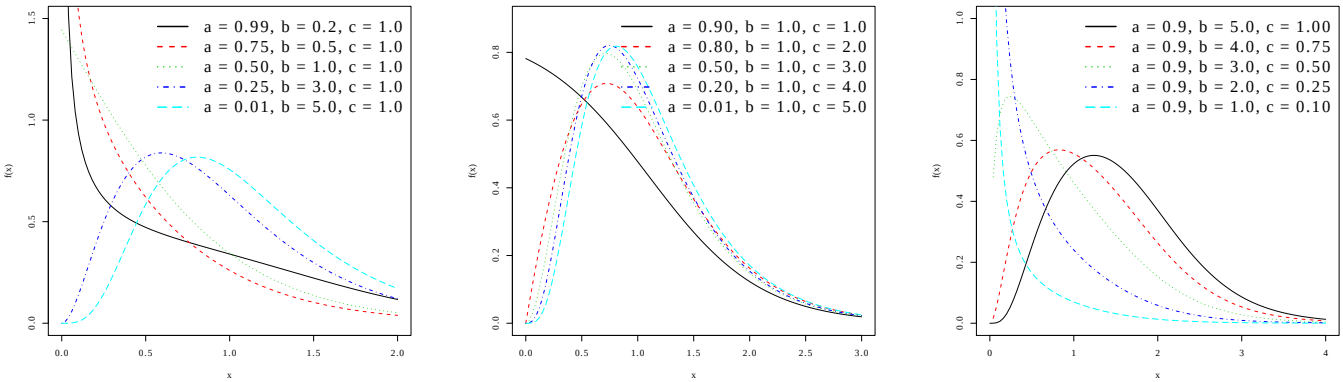


Figure 3.5: The ELG-gamma pdf for $\alpha = 1.0$, $\beta = 2.0$ and some values of a, b and c.

Plots of the ELG-gamma hrf for selected parameter values are displayed in Figure 3.6. The monotonically increasing and bathtub shapes are evident.

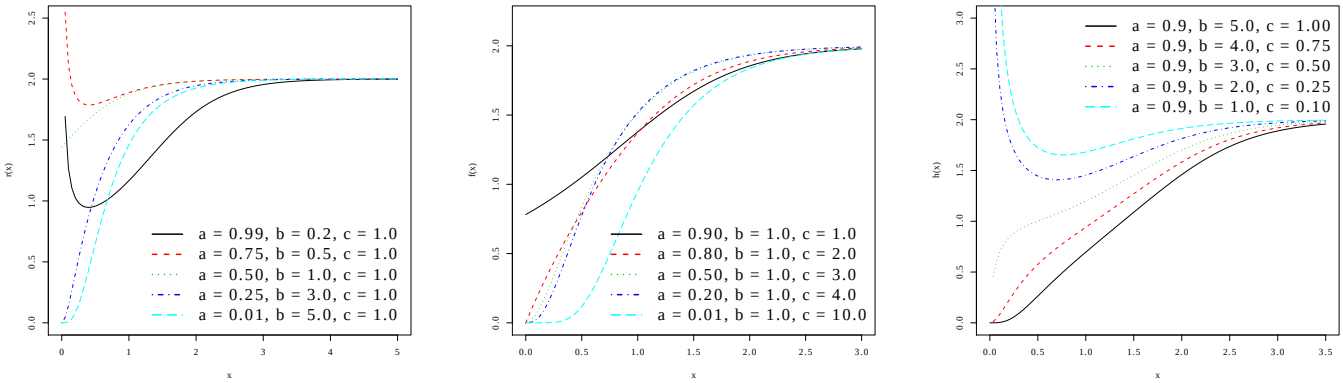


Figure 3.6: The ELG-Gamma hrf for the parameters $\alpha = 1.0$, $\beta = 2.0$ and some values of a, b and c.

3.3.4 ELG-Log-logistic (ELGLL)

Consider the log-logistic distribution with scale parameter $\alpha > 0$ and shape parameter $\beta > 0$, where the pdf and cdf (for $x > 0$) are

$$g(x) = \frac{\beta}{\alpha^{\beta}} x^{\beta-1} \left[1 + \left(\frac{x}{\alpha}\right)\right]^{-2} \quad \text{and} \quad G(x) = 1 - \left[1 + \left(\frac{x}{\alpha}\right)^{\beta}\right]^{-1}.$$

Inserting these expressions into (3.2) gives the ELGLL pdf

$$f_{ELGLL}(x) = \frac{-a\,b\,c\,\beta\,x^{\beta-1}\left[1-\frac{1}{1+(x/\alpha)^{\beta}}\right]^{b-1}\left\{\log\left\{1-a\left[1-\frac{1}{1+(x/\alpha)^{\beta}}\right]^{b}\right\}\right\}^{c-1}}{\alpha^{\beta}\,[\log(1-a)]^{c}\left(1+\frac{x}{\alpha}\right)^{2}\left\{1-a\left[1-\frac{1}{1+(x/\alpha)^{\beta}}\right]^{b}\right\}}. \tag{3.14}$$

Figure 3.7 provides some possible shapes of the ELGLL density function for selected parameters values.

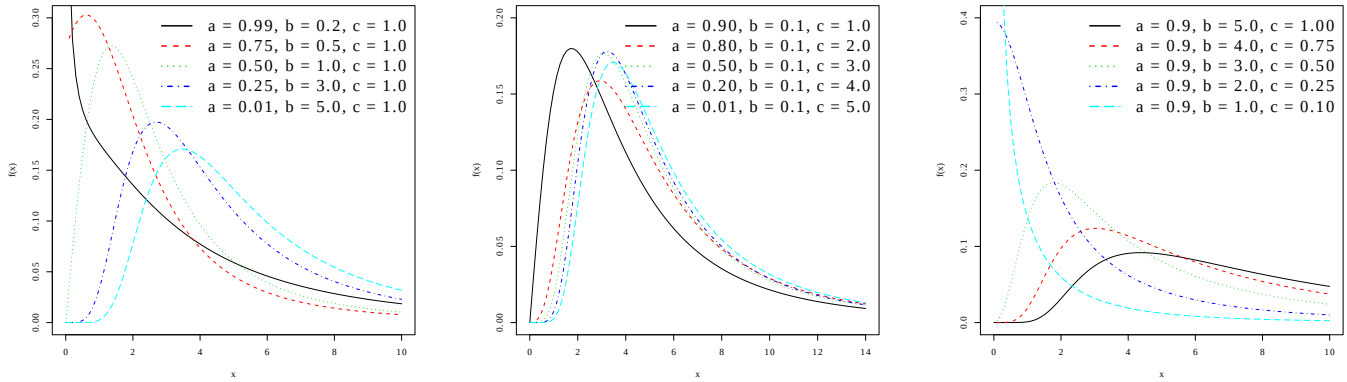


Figure 3.7: The ELGLL pdf for $\alpha = 2.0$, $\beta = 2.0$ and some values of a, b and c.

A random variable with density (3.14) is denoted by $X \sim ELGLL(a, b, c, \alpha, \beta)$. For $\alpha = 2$ and $\beta = 2$, possible shapes for the hrf of X are given in Figure 3.8.

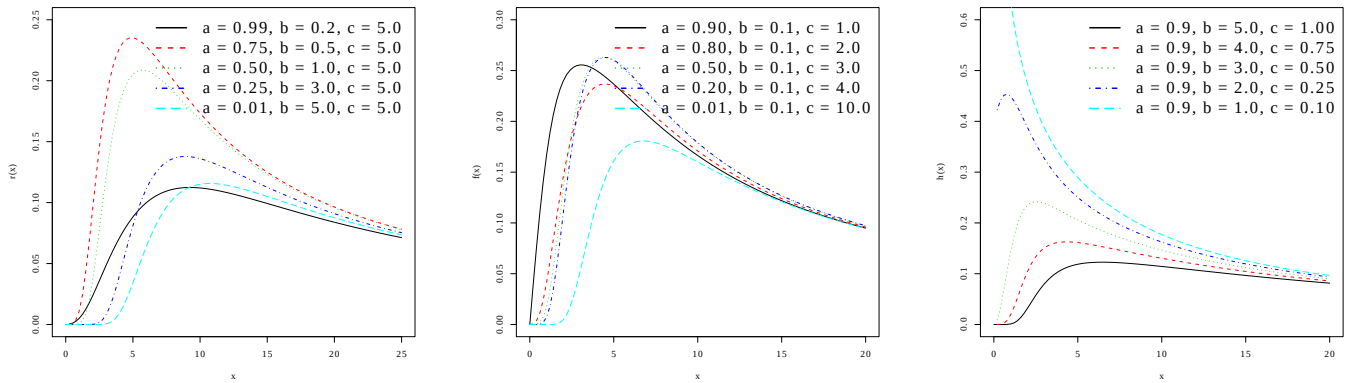


Figure 3.8: The ELGLL hrf for $\alpha = 2.0$, $\beta = 2.0$ and some values of a, b and c.

3.4 Mathematical properties

3.4.1 Moments

We assume that Y is a random variable having the baseline cdf $G(x)$. The moments of X can be obtained from the (r,k)th probability weighted moment (PWM) of Y defined by

$$\omega_{r,k} = \mathrm{E}[Y^r\, G(Y)^k] = \int_{-\infty}^{\infty} x^r\, G(x)^k\, g(x)dx.$$

We can write from equation (3.10),

$$\mu'_r = \mathrm{E}(X^r) = \sum_{m=0}^{\infty}[(m+c)b]\, b_m\, \omega_{r,(m+c)b-1},$$

where $\omega_{r,(m+c)b-1} = \int_0^1 Q_G(u)^r\, u^{(m+c)b-1}du$ can be evaluated at least numerically from any baseline qf.

We provide two formulae for the moment generating function (mgf) $M(s) = \mathrm{E}(\mathrm{e}^{s\,X})$ of X. The first formula for $M(s)$ comes from equation (3.10) as

$$M(s) = \sum_{m=0}^{\infty} b_m\, M_{(m+c)b}(s), \tag{3.15}$$

where $M_{(m+c)b}(s)$ is the exp-G generating function with power parameter $(m+c)b$.

Equation (3.15) can also be expressed as

$$M(s) = \sum_{m=0}^{\infty}[(m+c)b]\ b_m\, \rho_{(m+c)b}(s), \tag{3.16}$$

where the quantity $\rho_k(s) = \int_0^1 \exp\left[s\, Q_G(u)\right] u^{(m+c)b}du$ can be evaluated numerically. Equations (3.15) and (3.16) are the main results of this section.

3.4.2 Incomplete moments

Incomplete moments of a income distribution form natural building blocks for measuring inequality: for example, the Lorenz and Bonferroni curves depend upon the incomplete moments of the income distribution.

The nth incomplete moment of X is defined as $m_n(y) = \int_{-\infty}^{y} x^n\, f(x)dx$. Then, it can be expressed as

$$m_n(y) = \sum_{m=0}^{\infty}\, b_m \int_0^{G(y;\boldsymbol{\xi})}\, Q_G(u)^n\, u^{(m+c)b}\, du. \tag{3.17}$$

The integral in (3.17) can be evaluated at least numerically for most baseline distributions.

3.5 Maximum Likelihood Estimation

Let $x_1, \ldots, x_n$ be the observed values from the ELG-G distribution with the $p \times 1$ parameter vector $\boldsymbol{\theta} = (a, b, c, \boldsymbol{\xi})^\top$. We determine the MLEs of the parameters in $\boldsymbol{\theta}$ from complete samples only. The total log-likelihood function for $\boldsymbol{\theta}$ is given by

$$\ell(\boldsymbol{\Theta}) = n\,\log(a\,b\,c) - n\,c\,\log[-\log(1-a)] + \sum_{i=1}^{n}\log[g(x_i;\boldsymbol{\xi})] + (b-1)\sum_{i=1}^{n}G(x_i;\boldsymbol{\xi})$$
$$+ (c-1)\sum_{i=1}^{n}\log\left\{-\log[1-a\,G(x_i;\boldsymbol{\xi})^b]\right\} - \sum_{i=1}^{n}\log[1-a\,G(x_i;\boldsymbol{\xi})^b]. \tag{3.18}$$

The MLEs $\widehat{a}, \widehat{b}, \widehat{c}$ and $\widehat{\boldsymbol{\xi}}$ are determined as the values of a, b, c and $\boldsymbol{\xi}$ that maximize the log-likelihood function (3.18). There will be, in general, no closed-forms for these estimates, which require, in practice, numerical methods.

3.6 Bootstrap confidence intervals

The bootstrap method was introduced in 1979 by Efron (1979), who revitalized the jackknife resampling methodology and established a new area of research. The bootstrap methods present two approaches: the parametric bootstrap and nonparametric bootstrap. Parametric bootstrap refers to the case where the resampling is performed based on a distribution $F(\widehat{\theta})$ known or established, where $\widehat{\theta}$ is an estimator of θ. On the other hand, in bootstrap nonparametric, there is the lack of the distribution F. The resampling technique is based on the empirical distribution function $\widehat{F}_n$. Resample of $\widehat{F}_n$ is equivalent to resample data with replacement.

3.6.1 Bootstrap percentile interval

Let T_n be an estimator of the scale θ based on n observations and t its estimate. Let T_n^* be the same estimator based on n observations resampling from the original sample with replacement and t^* its estimate. For simplicity, suppose T_n is a continuous random variable. Denoting the pth quantile of the distribution of the random variable $T_n - \theta$ by a_p, we obtain

$$\Pr\left\{T_n - \theta \le a_{\frac{\alpha_1}{2}}\right\} = \Pr\left\{T_n - \theta \ge a_{1-\frac{\alpha_2}{2}}\right\} = \frac{\alpha}{2}.$$

As the amount $Q = T_n - \theta$ is invertible and T_n depends only on the sample, we can build the confidence interval for θ rewriting the events above, i.e., we can rewrite the events $T_n - \theta \le a_{\frac{\alpha_1}{2}}$ and $T_n - \theta \ge a_{1-\frac{\alpha_2}{2}}$ with $\theta > T_n - a_{\frac{\alpha_1}{2}}$ and $\theta < T_n - a_{1-\frac{\alpha_2}{2}}$, respectively. Thus, the confidence interval of level γ is given by the limits

$$\ell_{\alpha/2} = t - a_{1-\frac{\alpha_2}{2}},\ \ell_{1-\alpha/2} = t - a_{\frac{\alpha_1}{2}}.$$

In many situations, we do not know the distribution of $T_n - \theta$. In such cases, suppose that there is some transformation T_n, $U = h(T_n)$, such that U has a symmetric distribution. Suppose also that we can obtain the confidence interval of level $1-\alpha$ to $\phi = h(\theta)$. According to Davison and Hinkley (1997), we may use bootstrapping to obtain an approximation of the distribution $T_n - \theta$ using the distribution of $T_n^* - t$. Thus, we estimate the p quantile of $T_n - \theta$ by the $(J+1)p$-th ordered value

of $t^* - t$, i.e., the p-th quantile of $T_n - \theta$ is estimated by $t^*_{((J+1)p)} - t$. Similarly, the p quantile of $h(T_n) - h(\theta) = U - \phi$ can be estimated by the $(J+1)p$-th ordered value of $h(T_n^*) - h(t) = u^* - u$. Let b_p be the p-th quantile of $U - \phi$. Since U has symmetrical distribution, then $U - \phi$ also has a symmetrical distribution, as soon as it is true that $b_{\frac{\alpha}{2}} = -b_{1-\frac{\alpha}{2}}$. Using the symmetry of $U - \phi$, we have $h(\ell_{\alpha/2}) = u + b_{\alpha/2}$ and $h(\ell_{1-\alpha/2}) = u + b_{1-\alpha/2}$. As $b_{\alpha/2}$ and $b_{1-\alpha/2}$ are quantiles of $U - \phi$ and we know calculate these quantiles, the lower and upper limits of the confidence intervals are given by $u + (u^*_{((J+1)\alpha/2)} - u)$ and $u + (u^*_{((J+1)(1-\alpha/2))} - u)$, respectively, leading to the limits

$$u^*_{((J+1)\alpha/2)}, \quad u^*_{((J+1)(1-\alpha/2))},$$

whose transformation to θ is

$$t^*_{(J+1)\alpha/2}, \quad t^*_{(J+1)(1-\alpha/2)}. \tag{3.19}$$

Note that we do not know the transformation h. The confidence interval of level $1 - \alpha$ for the parameter θ does not involve h and it can be evaluated without knowledge of this transformation. The interval (3.19) is known as the bootstrap interval percentile. According to Davison and Hinkley (1997, p. 203) the percentile method can be applied to any statistic.

3.7 Optimization algorithm

3.7.1 Swarm intelligence and particle swarm optimization

A package developed for `R` language widely used in the area of probability distributions is the `AdequacyModel` package, version 1.0.8, available for download in `https://cran.r-project.org/web/packages/AdequacyModel/index.html` under the terms of the GPL license (≥ 2).

This package focuses on construction of statistics of adequacy of adjustment of probabilistic models and these statistics depend on the MLEs also provided by the package. However, it is noted that several users of the package are having difficulties to obtain these estimates using optimization methods very popular as is the case of Nelder-Mead method (Nelder and Mead, 1965), L-BFGS-B (Nelder and Mead, 1965) and simulated annealing (Belisle, 1992). In general, these users are working with new distributions that have a large number of parameters. The main problem is summed up by the difficulty to obtain initial values necessary for these methods and non-convergence of the algorithm for global optimization used for maximization of the log-likelihood function of these new models. In this sense, the swarm intelligence proved to be a good strategy for the optimization of these functions and, in general, produce better results and with the advantage of not having to provide initial values. The Swarm Intelligence (SI) is the term used to designate systems of artificial intelligence, where the collective behavior of individuals in a population provides simple solutions or consistent patterns emerge.

The SI was employed using the Particle Swarm Optimization (PSO) method developed by Eberhart and Kennedy (1995) to obtain the MLEs of the model parameters and in the study of simulation about the interval estimates for bootstrap percentile as presented in Section 3.7.3, as well as to obtain the MLEs for different models considered in applying the ELG-G model (see Section 3.8). The implementation of the PSO method built in this work will be improved and adapted to later versions of the `AdequacyModel` package.

Eberhart and Kennedy (1995) developed the PSO method using as a basis the studies of Reynolds (1987) and Heppner and Grenander (1990) that provided models of simulations of flock of birds. The

PSO method optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. The movement of the particles in the search space is randomized. The PSO algorithm is presented below, where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function to be minimized, S is the number of particles in the swarm (set of feasible points, i.e. search region), each particle having a vector position $x_i \in \mathbb{R}^n$ in the search-space and a vector velocity defined by $v_i \in \mathbb{R}^n$. Let p_i be the best known position of particle i and g the best position of all particles.

1. For each particle $i = 1, \ldots, S$ do:
   - Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
   - Initialize the particle's best known position to its initial position: $p_i \leftarrowtail x_i$.
   - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
   - Initialize the particle's velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$.
2. Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
   - For each particle $i = 1, \ldots, S$ do:
     - Pick random numbers: $r_p, r_g \sim U(0, 1)$.
     - For each dimension $d = 1, \ldots, n$ do:
       - Update the particle's velocity: $v_{i,d} \leftarrowtail \omega\, v_{i,d} + \varphi_p r_p(p_{i,d} - x_{i,d}) + \varphi_g r_g(g_d - x_{i,d})$.
     - Update the particle's position: $x_i \leftarrowtail x_i + v_i$
     - If $f(x_i) < f(p_i)$ do:
       - Update the particle's best known position: $p_i \leftarrowtail x_i$
       - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
3. Now g holds the best found solution.

The parameter ω is called inertia coefficient and as the name implies controls the inertia of each particle arranged in the search region. The quantities ω_p and ω_g control the acceleration of each particle and are called accelerated coefficients. The variance of the best candidates can be used as a stopping criterion, i.e., the algorithm will stop if the variance is less than some real $\varepsilon > 0$.

3.7.2 Hardware Used

The law that established the II PLANIN (Plan Informatics and Automation), approved by the Brazilian National Congress in October 1991, proposed the installation of a National Center for Supercomputing (CESUP) to provide advanced computing services to brazilian researchers. This center was set up at the Federal University of Rio Grande do Sul (UFRGS). The Brazil has some High Performance Processing National Centers (CENAPAD).

We use the hardware available in the National Supercomputing Center - CESUP (CENAPAD UFRGS). The CESUP has two clusters: Cluster Sun Fire, dubbed of Newton, and the Altix cluster SGI, also known as Gauss. The cluster settings are described below.

Cluster Sun Fire (Newton):

- 45 processing nodes and 3 nodes of management; 8 GPUs nVIDIA Tesla; 1 GPU AMD FireStream; 1 switch Voltaire InfiniBand; total of 1296GB of RAM memory; total of 188TB of storage capacity, wherein 158TB are shared with the cluster SGI Altix Gauss; theoretical performance peak of 12.94 Tflops.

Cluster SGI Altix (Gauss):

- 64 blades of the processing and 3 nodes of service; total of 4TB of RAM memory; total of 174TB of storage capacity, wherein 158TB are shared with the cluster Sun Fire (Newton); theoretical performance peak of 15.97 Tflops.

We use the cluster SGI Altix (Gauss). Each of the 64 units processing of the SGI Altix cluster has 2 dodeca-core processors (24 cores) AMD Opteron working at a frequency of 2.3GHz, 128KB of cache L1 per core (data + instructions), 512KB of cache L2 per core and 12MB of cache L3 per socket. Cluster Gauss has integrated DDR3 memory controller that supports frequencies of 1333MHz and bandwidth of 42.7GB/s per CPU, totaling 64GB of RAM per unit. More information about the hardware available by CESUP can be found in `http://www.cesup.ufrgs.br/`.

These hardware provide greater speed in Monte Carlo simulations to be presented later. We use the `R` programming language and the `Rmpi`, `doSNOW` and `foreach` packages to create the parallel processes. The `Rmpi` package provides an interface (wrapper) to MPI APIs. It also provides interactive `R` slave environment. The `doSNOW` package provides a parallel backend for the `%dopar%` function (function available in the `foreach` package for parallelization of loops) using the `Rmpi` package.

3.7.3 Simulation study

We present some Monte Carlo simulations to evaluate the performance of interval estimates by bootstrap percentile. We simulate 20,000 trails of the ELG-Weibull true model with fixed parameters $a = 0.5$, $b = 1.5$ and $c = 1.5$, with the baseline Weibull distribution having parameters $\alpha = 1.5$ and $\beta = 1.5$. We take the nominal level of 95% and sample sizes: 20, 60, 100 and 500. We consider 500 bootstrap resampling in each Monte Carlo iteration.

The evaluation of the random intervals obtained by bootstrap percentile is taken by the level of coverage of these intervals, i.e., we determine the percentage of confidence intervals containing the true parameter within the interval. Taking advantage of the bootstrap used for construction the percentile intervals, we evaluate the standard errors of the MLEs obtained by the PSO method.

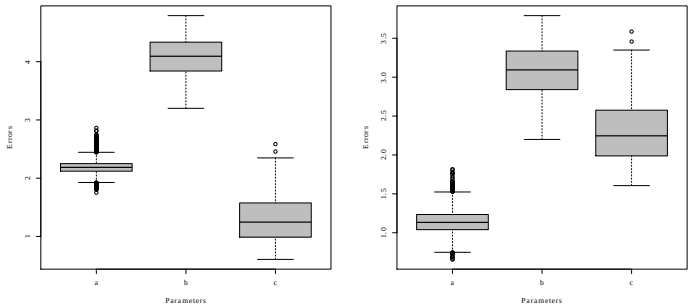
Table 1 gives the average of the standard errors obtained using non-parametric bootstrap. The table also shows the average of the amplitudes of interval estimates, the coverage of the intervals for different sample sizes and the time of the simulations presented in hours. For small samples, in particular, for $n = 20$, we note that the interval estimates by bootstrap percentiles coverage far below the fixed nominal level. The coverage becomes reasonably close to the 95% nominal level in larger samples (100 and 500).

The amplitudes of interval estimates are small for all scenarios of the simulations. We also register the bootstrap errors for each iteration of Monte Carlo. For different sample sizes, we note small errors for all parameters of the ELG-Weibull model. We evaluate the errors by bootstrap because there is

Table 3.1: Percentage of coverage and average amplitude of interval estimates for the parameters added by the generator by bootstrap percentile at a nominal level of 95%.

n	**Parameter**	**Coverage (%)**	**Amplitude**	**Error**	**Time (in hours)**
	a	86.7854	1.3456	2.3325	
$n = 20$	b	87.4416	1.7376	4.1464	100.4356
	c	89.7584	2.1222	1.4353	
	a	91.5443	2.1210	1.2756	
$n = 60$	b	90.5432	1.5436	3.2542	170.1102
	c	92.3453	2.0787	2.4252	
	a	93.4545	1.5345	2.3431	
$n = 100$	b	92.4539	2.4536	2.3014	213.4464
	c	94.4235	3.3445	3.3945	
	a	93.9716	1.5646	2.1455	
$n = 500$	b	94.2325	2.5345	2.5514	432.5643
	c	95.1215	3.2148	3.8789	

no exact way to obtain the errors of the MLEs using the PSO method. Figure 3.9 displays boxplots of the errors for the parameters added in the ELG-Weibull distribution.



(a) Sample size 20 (b) Sample size 60

(c) Sample size 100 (d) Sample size 500

Figure 3.9: Errors evaluated by bootstrap and the MLEs obtained by the PSO method with 500 bootstrap replicates.

3.8 Applications

In this section, the usefulness of the ELG-Weibull distribution is proved empirically by means of two real data sets. In the applications, we use the AdequacyModel package version 1.0.8 available in the R programming language. First, we consider a data set from Smith and Naylor (1987). The data are the strengths of 1.5 cm glass fibres, measured at the National Physical Laboratory, England. Unfortunately, the units of measurement are not given in the paper. The data set is also available for download at `http://www.stat.ncsu.edu/research/sas/sicl/data/`. The second application takes into account the data related to the percentages of body fat determined by underwater weighting and various body circumference measurements for 250 men. For details about the data set, see `http://lib.stat.cmu.edu/datasets/`. Table 3.2 gives some descriptive statistics for the two data sets. They are obtained in the `AdequacyModel` package (version `1.0.8`).

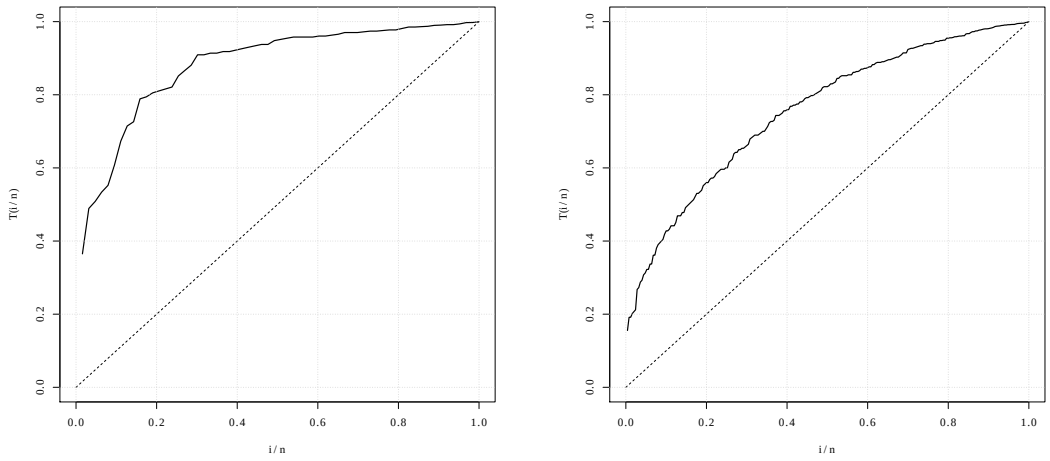
Table 3.2: Descriptive statistics.

Statistics	Real data sets	
	Glass Fibres	Body Fat (%)
Mean	1.5068	19.3012
Median	1.5900	19.2500
Mode	1.7000	22.5000
Variance	0.1051	67.7355
Skewness	-0.8999	0.1953
Kurtosis	0.9238	-0.3815
Maximum	2.2400	47.5000
Minimum	0.5500	3.0000
n	63	250

One of the important devices, which can help selecting a particular model, is the total time on test (TTT) plot (Aarset, 1987). The TTT plots for the fibres data and for the number of successive failure data are displayed in Figure 3.10. Both TTT plots in Figure 3.10 yield a concave curve and then an increasing hrf. Then, these plots indicate the appropriateness of the ELG-Weibull distribution to fit these data. Figure 3.11 displays the Gaussian kernel density estimation for the glass fibres data and percentage of body fat data.

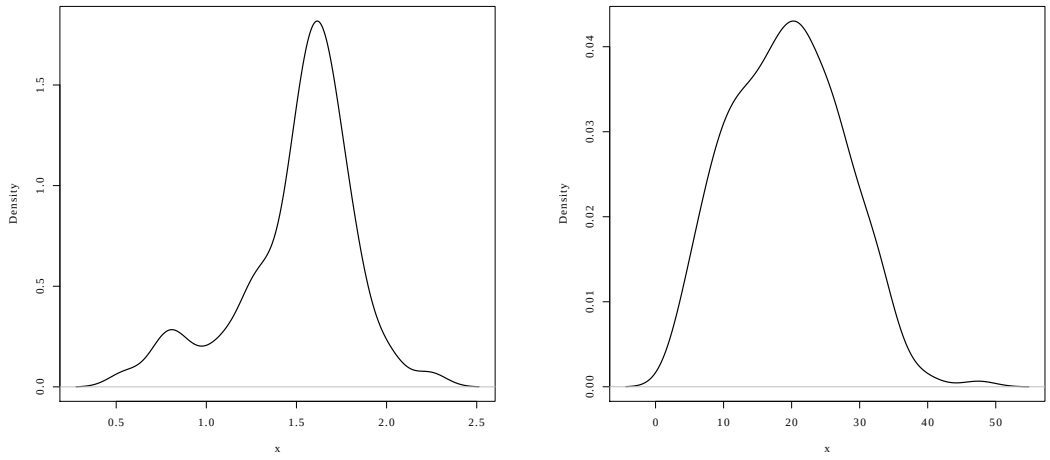
For these data sets, we fit the ELG-Weibull (ELG-W) distribution defined by (3.12) and compare it with the Kumarasuamy Weibull Poisson (KW-WP) (Ramos et al., 2015), Kumaraswamy Weibull (KW-W) (Cordeiro et al., 2010), exponentiated Weibull (EW) (Mudholkar and Srivastava, 1993), New-type Nadarajah-Haghighi (NTNH) (Lemonte, 2013), modified Weibull (MW) (Xie et al., 2002), Chen (Chen, 2000), gamma and Nadarajah-Haghighi (NH) (Nadarajah and Haghighi, 2011) distributions. The MLEs of the model parameters (with standard errors in parentheses) for the ELG-W, KW-WP, EW, NTNH and the other models are listed in Table 3.3 for the two data sets. The `R` language was also used to obtain the MLEs by heuristic method of global optimization PSO discussed in the previous section.

We can also perform formal goodness-of-fit tests in order to verify which distribution fits better to these data. We consider the Cramér-von Mises (W) and Anderson-Darling (A), described in details by Chen and Balakrishnan (1995), and Kolmogorov-Smirnov (KS) statistics. In general, the smaller the values of these statistics, the better the fit. Table 3.4 gives the values of the Akaike information criterion (AIC), Bayesian information criterion (BIC), consistent Akaike information



(a) (b)

Figure 3.10: The TTT plots for the: (a) glass fibres and (b) percentage of body fat.



(a) (b)

Figure 3.11: The Gaussian kernel density estimation for: (a) the glass fibres and (b) percentage of body fat.

criterion (CAIC) and Hannan-Quinn information criterion (HQIC), and the A, W and KS statistics for the models fitted to both data sets. Thus, according to these formal tests, the ELG-W model fits the data better than the other distributions. Since the values of the AIC, CAIC and HQIC statistics are smaller for the ELG-W distribution compared to those values of the other fitted models, the new distribution is a very competitive model to explain the data.

Table 3.3: MLEs for the glass fibres data denoted by (I) and percentage of body fat data denoted by (II).

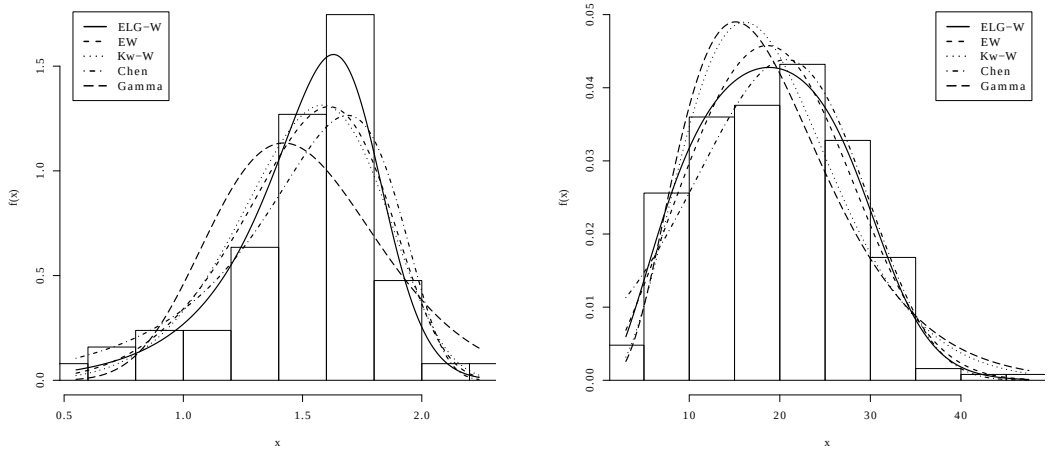
Data set	**Distribution**	**Estimates and standard errors in parenthesis**				
I	ELG-W (α, β, a, b, c)	3.5684	1.1824	0.9944	0.1650	2.7926
		(0.0132)	(0.0111)	(0.0018)	(0.0442)	(0.0922)
	KW-WP (a, b, c, λ, β)	0.9377	3.2395	6.4021	17.2578	0.3163
		(0.0055)	(0.0921)	(0.0025)	(0.7777)	(0.2853)
	KW-W (a, b, c, β)	7.7418	2.6226	0.7262	14.1218	
		(0.0171)	(0.2583)	(0.0543)	(0.3187)	
	EW (α, β, a)	7.2846	1.7181	0.6712		
		(0.0099)	(0.1129)	(0.0724)		
	NTNH (α, λ, β)	14.2416	0.0648	9.9259		
		(0.3474)	(0.0667)	(0.3701)		
	MW (α, β, λ)	3.1640	5.6882	13.6191		
		(0.3537)	(0.1323)	(0.2122)		
	Chen (λ, β)	0.0720	1.9604			
		(0.0042)	(0.0335)			
	Gamma (α, β)	0.0864	17.4396			
		(0.0014)	(0.1240)			
	NH (α, λ)	24.884	0.0212			
		(0.2120)	(0.00014)			
II	ELG-W (α, β, a, b, c)	1.8303	12.8023	0.9733	2.5701	0.6136
		(0.0221)	(0.0043)	(0.0345)	(0.0627)	(0.0225)
	KW-WP (a, b, c, λ, β)	22.1310	24.9999	0.2568	13.9977	0.2074
		(0.2221)	(0.0030)	(0.0053)	(0.102)	(0.0022)
	KW-W (a, b, c, β)	0.5013	14.8283	8.9928	24.9999	
		(0.0002)	(0.5876)	(0.2531)	(1.1113)	
	EW (α, β, a)	3.0043	23.8105	0.7558		
		(0.0553)	(0.0077)	(0.1137)		
	NTNH (α, λ, β)	1.8273	0.0419	3.7999		
		(0.0033)	(0.0033)	(0.0003)		
	MW (α, β, λ)	21.6646	1.5020	0.0241		
		(0.0033)	(0.0028)	(0.0534)		
	Chen (λ, β)	0.0065	0.5192			
		(0.0001)	(0.0448)			
	Gamma (α, β)	4.1876	4.6091			
		(0.3322)	(0.2232)			
	NH (α, λ)	17.3791	0.0022			
		(0.1840)	(0.7525)			

Plots of the estimated pdfs, cdfs and survival functions of the ELG-W and other models fitted to both data sets are displayed in Figures 3.12, 3.13 and 3.14, respectively. They reveal that the ELG-W distribution is superior to the other distributions in terms of model fitting.

Table 3.4: Goodness-of-fit statistics for the glass fibres data denoted by (I) and percentages of body fat data denoted by (II).

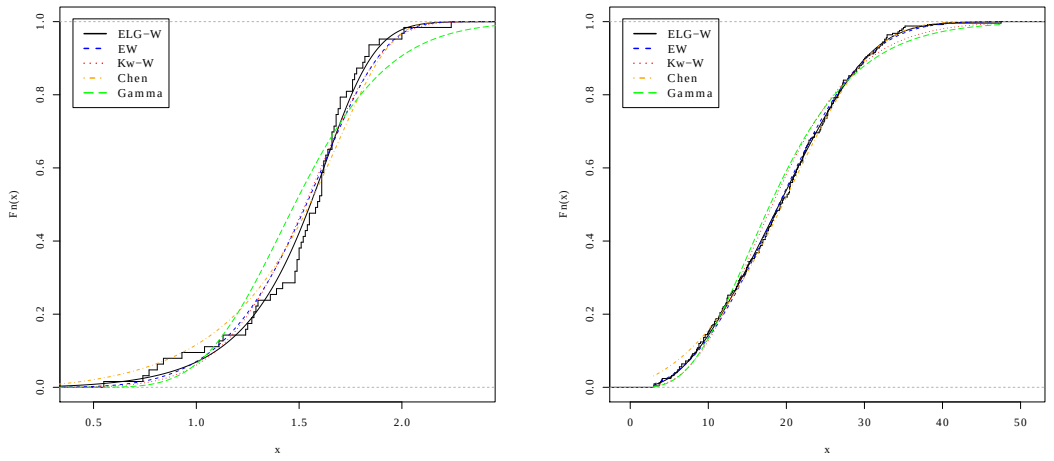
Data set	**Distribution**	A^*	W^*	KS	AIC	CAIC	BIC	HQIC
I	ELG-W (α, β, a, b, c)	0.5513	0.0959	0.1105	34.4784	35.5311	45.1941	38.6930
	KW-WP (a, b, c, λ, β)	1.7612	0.3216	0.1689	44.3267	45.3794	55.0424	48.5413
	KW-W (a, b, c, β)	1.2689	0.2304	0.1516	38.2397	38.9293	46.8122	41.6113
	EW (α, β, a)	1.1118	0.2000	0.1462	35.3510	35.7578	41.7804	37.8798
	NTNH (α, λ, β)	2.7480	0.5006	0.2105	50.0737	50.4805	56.5031	52.6024
	MW (α, β, λ)	1.2901	0.2345	0.1514	36.3860	36.7927	42.8154	38.9147
	Chen (λ, β)	0.9623	0.1615	0.1373	36.9227	37.1227	41.2090	38.6085
	Gamma (α, β)	3.1174	0.5684	0.2164	51.9031	52.1031	56.1893	53.5889
	NH (α, λ)	2.3541	0.4294	0.4513	143.1259	143.3259	147.4121	144.8117
II	ELG-W (α, β, a, b, c)	0.1246	0.0164	0.0242	1758.6590	1758.9050	1776.2660	1765.7460
	KW-WP (a, b, c, λ, β)	1.3448	0.2200	0.0627	1772.8220	1773.0680	1790.4290	1779.9080
	KW-W (a, b, c, β)	1.3420	0.2204	0.0667	1770.3700	1770.5330	1784.4560	1776.0390
	EW (α, β, a)	0.2549	0.0347	0.0327	1757.7540	1757.8510	1768.3180	1762.0050
	NTNH (α, λ, β)	1.6362	0.2702	0.0743	1771.6980	1771.7950	1782.2620	1775.9490
	MW (α, β, λ)	0.4229	0.0525	0.0399	1771.6800	1771.7770	1782.2440	1775.9310
	Chen (λ, β)	0.5479	0.0688	0.0360	1768.4090	1768.4580	1775.4520	1771.2440
	Gamma (α, β)	1.9585	0.3240	0.0764	1773.3760	1773.4240	1780.4180	1776.2100
	NH (α, λ)	0.4922	0.0772	0.2186	1872.0240	1872.0730	1879.0670	1874.8590

1 - The statistics are obtained in the `AdequacyModel` package, version `1.0.8`.
2 - It is used rounding to the fourth decimal place.



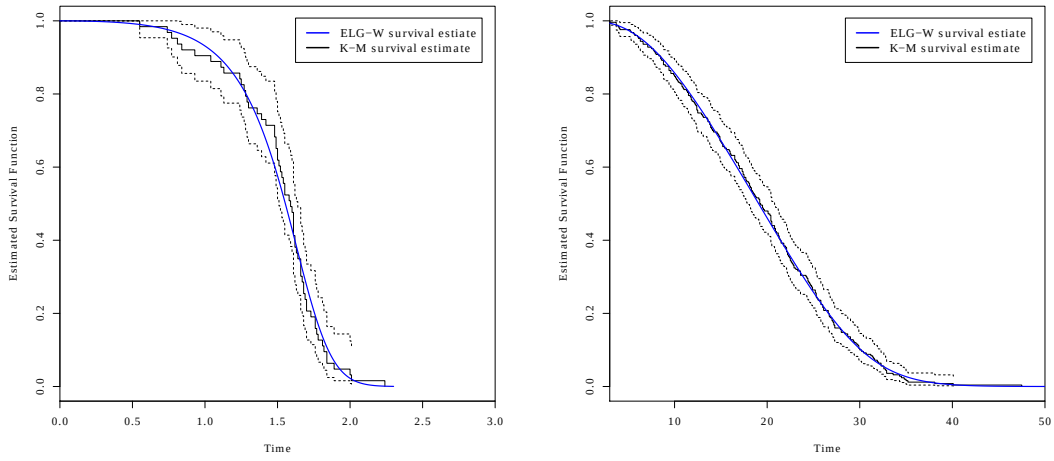
(a) (b)

Figure 3.12: Estimates of the density functions for the: (a) glass fibres (b) percentages of body fat.



(a) (b)

Figure 3.13: Estimates of the distribution functions and empirical distribution for the: (a) glass fibre and (b) percentages of body fat.



(a) (b)

Figure 3.14: Kaplan-Meier (K-M) estimates compared with the ELG-W survival estimates for the: (a) glass fibres and (b) percentages of body fat.

3.9 Conclusions

We define a new class of distributions called the exponentiated logarithmic-G ("ELG") family. The proposed family can be motivated by compounding the exponentiated generated construction and the logarithmic distribution. It can provide better fits than some well-known lifetime distributions, which represents a remarkable feature of this family. We derive some of its structural properties including moments, quantile and generating functions. We use the maximum likelihood method to

estimate the model parameters. We provide a simulation study to show the accuracy of the estimates. Further, we adopt the bootstrap percentile technique to obtain confidence intervals for the model parameters. We give two applications to real data to illustrate the potentiality of the proposed family. We hope this generalization may attract wider applications in Statistics.

APPENDIX B

Script in Julia language

```
# Binomial coefficient to be generalized to noninteger
# arguments (including complex x and y) as
function binomialG(x,y)
   return gamma(BigFloat(x)+1)/(gamma(BigFloat(y)+1)*
   gamma(BigFloat(x)-BigFloat(y)+1))
end

# Stirling numbers of the second kind.
function S(n,m)
   a = 1/gamma(m+1)
   v = zeros(Float64,int(trunc(m))+1,1)
   for i = 0:(length(v)-1)
      v[i+1] = (-1)^i*binomialG(m,i)*(m-i)^n
   end
   return a*sum(v)
end

# Stirling numbers of the first kind.
function s(n,m)
   if n<m
      return 0
      else
      v = zeros(Float64,int(trunc(n-m))+1,1)
      for k = 0:(length(v)-1)
         v[k+1]=(-1)^k*binomialG(k+n-1,k+n-m)*
         binomialG(2*n-m,n-k-m)*S(k-m+n,k)
      end
      return sum(v)
   end
end

# Stirling polynomial defined in
# http://mathworld.wolfram.com/StirlingPolynomial.html
```

```
function poly(n,m)
    return (-1)^n/binomialG(m,n)*s(m+1,m-n+1)
end

# Fundamental formula for the Stirling polynomial defined by
# MORGAN WARD (1934, p. 2).
function psi(n,m)
   if n>m
      error("Sorry. With this algorithm is only
            possible to calculate for n <= m.")
   end
   return poly(n,m)/(factorial(BigInt(n))*(m+1))
end
```

References

[1] Aarset, M V. How to identify bathtub hazard rate. *IEEE Transactions on Reliability*, v. 36, p. 106–108, 1987.

[2] Belisle, C J P. Convergence theorems for a class of simulated annealing algorithms on Rd. *Journal Applied Probability*, v. 29, p. 885–895, 1992.

[3] Bezanson, J, Karpinski, S, Shah, V B and Edelman, A. *Julia: A fast dynamic language for technical computing*. arXiv preprint arXiv:1209.5145, 2012.

[4] Castellares, F and Lemonte, A J. A new generalized Weibull distribution generated by gamma random variables. *Journal of the Egyptian Mathematical Society*, v. 23, p. 382–390, 2015.

[5] Chen, G and Balakrishnan, N. A general purpose approximate goodnessof-fit test. *Journal of Quality Technology*, v. 27, p. 154–161, 1995.

[6] Chen, Z. A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function. *Stat.Probability Lett.*, v. 49, p. 155-161, 2000.

[7] Cordeiro, G M, Ortega, E M M and Nadarajah, S. The Kumaraswamy Weibull distribution with application to failure data. *Journal of the Franklin Institute*, v. 347, p. 1399–1429, 2010.

[8] Cordeiro, G. M. and de Castro M. A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, v. 81, p. 883–898, 2011.

[9] Davison, A and Hinkley, D. *Bootstrap methods and their application*. Cambridge: Cambridge University Press, 1997.

[10] Eberhart, R C and Kennedy, J. A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human science*, v. 1, p. 39–43, 1995.

[11] Efron, B. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, v. 7, p. 1–26, 1979.

[12] Eugene, N, Lee, C and Famoye, F. Beta-normal distribution and its applications. *Communications in Statistics-Theory and Methods*, v. 31, p. 497–512, 2002.

[13] Flajonet, P. and Odlyzko, A. Singularity analysis of generating function. *SIAM: SIAM J. Discr. Math.*, v. 3, p. 216–240, 1990.

[14] Flajonet, P and Sedgewick, R. *Analytic Combinatorics*, Cambridge University Press, 2009.

[15] Heppner, F. and Grenander, U. A stochastic nonlinear model for coordinated bird flocks. v. *The Ubiquity of Chaos*, p. 233–238, 1990.

[16] Lemonte, J A. A new exponential-type distribution with constant, decreasing, increasing, upside-down bathtub and bathtub-shaped failure rate function. *Computational Statistics and Data Analysis*, v. 62, p. 149–170, 2013.

[17] Mudholkar, G S and Srivastava, D K. Exponentiated Weibull family for analyzing bathtub failure-rate data. *IEEE Transactions on Reliability*, v. 42, p. 299–302, 1993.

[18] Nadarajah, S. and Haghighi, F. An extension of the exponential distribution. *Statistics*, v. 45, p. 543–558, 2011.

[19] Nelder, J A and Mead, R. A simplex algorithm for function minimization. Computer Journal, v. 7, p. 308–313, 1965.

[20] Nielsen, N. *Handbuch der Theorie der Gammafunktion*, Leipzig, 1906.

[21] Reynolds, Craig W. Flocks, herds and schools: A distributed behavioral model. ACM Siggraph Computer Graphics, v. 4, p. 25–34, 1987.

[22] Ramos, M W A, Marinho, P R D, Cordeiro, G M, Silva, R V and Hamedani, G G. The Kumaraswamy-G Poisson Family of Distributions. *Journal of Statistical Theory and Applications*, v. 14, p. 222–239, 2015.

[23] Reynolds, C. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, v. 21, p. 25–34, 1987.

[24] Roman, S. The Umbral Calculus. *Academic Press*, New York, p. 59–63, 1984.

[Smith and Naylor(1987)] Smith, R L and Naylor, J C. A comparison of maximum likelihood and Bayesian estimators for the three-parameter Weibull distribution. *Applied Statistics*, v. 36, p. 358–369, 1987.

[25] Wang, J, and Gerardo B. Object analysis of multi-valued images. U.S. Patent, v. 4, p. 847, 1989.

[26] Ward, M. The representation of Stirling's numbers and Stirling's polynomials as sums of factorial. *Am. J. Math.*, v. 56, p. 87–95, 1934.

[27] Xie, M, Tang, Y and Goh, T N. A modified Weibull extension with bathtub-shaped failure rate function. *Reliab. Eng. Syst. Saf.*, v. 76, p. 279–285, 2002.

[28] Zografos, K. and Balakrishnan, N. (2009). On families of beta- and generalized gamma- generated distributions and associated inference. *Statistical Methodology*, v. 6, p. 344–362.

CHAPTER 4

The Type I Half-Logistic Family of Distributions

Abstract: We study general mathematical properties of a new class of continuous distributions with an extra positive parameter called the type I half-logistic family. We present some special models and investigate the asymptotics and shapes. The new density function can be expressed as a linear combination of exponentiated densities based on the same baseline distribution. We derive a power series for the quantile function. Explicit expressions for the ordinary and incomplete moments, quantile and generating functions, Bonferroni and Lorenz curves, Shannon and Rényi entropies and order statistics are determined. We introduce a bivariate extension of the new family. We discuss the estimation of the model parameters by maximum likelihood and illustrate its potentiality by means of two applications to real data.

Keywords: Half-logistic distribution. Maximum likelihood. Moment. Order statistic. Quantile function. Rényi entropy.

4.1 Introduction

Recently, there has been an increased interest in defining new generated families of univariate continuous distributions by introducing additional shape parameters to the baseline model. The generated distributions have attracted several statisticians to develop new models because the computational and analytical facilities available in most symbolic computation software platforms. Several mathematical properties of the extended distributions may be easily explored using mixture forms of exponentiated-G (exp-G for short) distributions. These last distributions are studied by Mudholkar and Hutson (1996), Gupta and Kundu (2001) and Nadarajah and Kotz (2006), among others. We define the cumulative distribution function (cdf) of the new *type I half-logistic* (TIHL) family of distributions by

$$F(x;\lambda,\boldsymbol{\xi}) = \int_0^{-\log[1-G(x;\boldsymbol{\xi})]} \frac{2\lambda \mathrm{e}^{-\lambda \mathrm{t}}}{(1+\mathrm{e}^{-\lambda \mathrm{t}})^2}\, dt = \frac{1-[1-G(x;\boldsymbol{\xi})]^{\lambda}}{1+[1-G(x;\boldsymbol{\xi})]^{\lambda}}, \tag{4.1}$$

where $G(x;\boldsymbol{\xi})$ is the baseline cdf depending on a parameter vector $\boldsymbol{\xi}$ and $\lambda > 0$ is an additional shape parameter. For each baseline G, we can generate the *type I half-logistic-G* ("TIHL-G") distribution by the cdf (4.1). Equation (4.1) is a wider class of continuous distributions.

The corresponding probability density function (pdf) to (4.1) is given by

$$f(x;\lambda,\boldsymbol{\xi}) = \frac{2\lambda\, g(x;\boldsymbol{\xi})[1-G(x;\boldsymbol{\xi})]^{\lambda-1}}{\{1+[1-G(x;\boldsymbol{\xi})]^{\lambda}\}^2}, \tag{4.2}$$

where $g(x;\boldsymbol{\xi})$ is the baseline pdf. Equation (4.2) will be most tractable when $G(x;\boldsymbol{\xi})$ and $g(x;\boldsymbol{\xi})$ have simple expressions. Hereafter, a random variable X having pdf (4.2) is denoted by $X \sim \text{TIHL}(\boldsymbol{\xi}, \lambda)$. Further, we omit the dependence on the parameters $\boldsymbol{\xi}$ and λ and write simply $f(x) = f(x;\boldsymbol{\xi},\lambda)$.

The hazard rate function (hrf) of X becomes

$$h(x) = \frac{\lambda\, g(x;\boldsymbol{\xi})}{[1-G(x;\boldsymbol{\xi})]\,\{1+[1-G(x;\boldsymbol{\xi})]^{\lambda}\}}. \tag{4.3}$$

The TIHL family class is easily simulated by inverting (4.1) as follows: if u has a uniform $U(0,1)$ distribution, the solution of the nonlinear equation

$$x_u = G^{-1}\left(1-\left[\frac{1-u}{1+u}\right]^{\frac{1}{\lambda}}\right), \tag{4.4}$$

has density function (4.2).

4.2 Special TIHL distributions

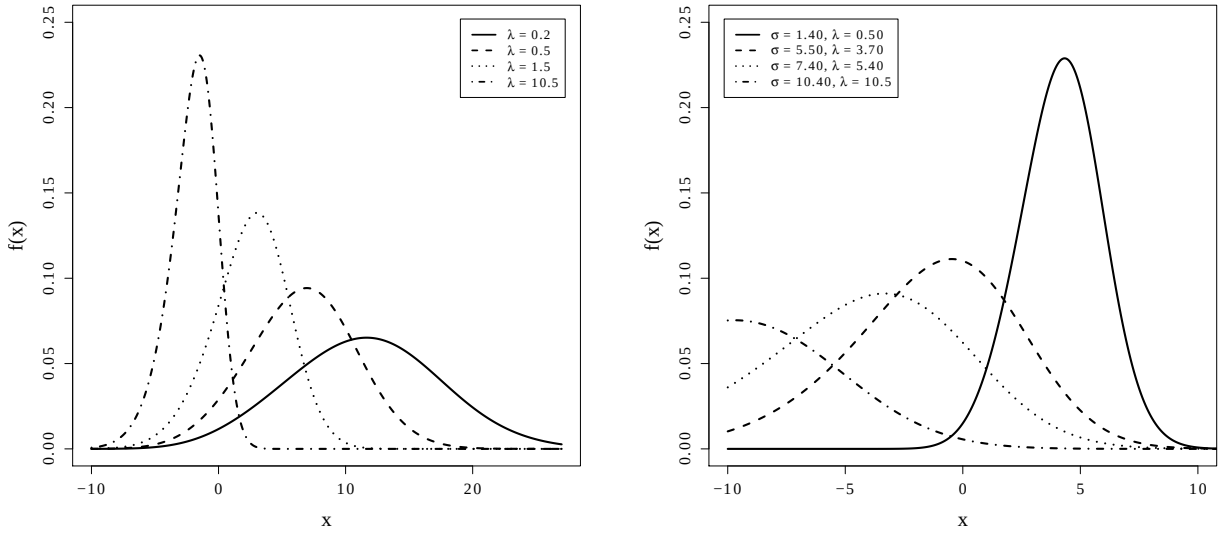
4.2.1 The type I half-logistic normal (TIHL-N) model

The TIHL-N distribution is defined from (4.2) by taking $G(x;\boldsymbol{\xi}) = \Phi(\frac{x-\mu}{\sigma})$ and $g(x;\boldsymbol{\xi}) = \phi(\frac{x-\mu}{\sigma})$ to be the cdf and pdf of the normal distribution with parameters μ and σ^2, respectively, where $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of the standard normal distribution, respectively, and $\boldsymbol{\xi} = (\mu, \sigma^2)$. The TIHL-N pdf is

$$f(x;\lambda,\mu,\sigma) = \frac{2\lambda\phi(\frac{x-\mu}{\sigma})[1-\Phi(\frac{x-\mu}{\sigma})]^{\lambda-1}}{\sigma\left\{1+[1-\Phi(\frac{x-\mu}{\sigma})]^{\lambda}\right\}^2}, \tag{4.5}$$

where $x \in \mathbb{R}$, $\mu \in \mathbb{R}$ is a location parameter and $\sigma > 0$ is a scale parameter.

A random variable with density (4.5) is denoted by $X \sim \text{TIHL-N}(\lambda, \mu, \sigma^2)$. Plots of the TIHL-N density function for some parameter values are displayed in Figures 4.1(a) and 4.1(b). These plots indicate that increasing λ causes a flattening of the pdf curves.



(a) (b)

Figure 4.1: The TIHL-N densities for (a) $\mu = 2.5$ and $\sigma = 3.4$ and for (b) $\mu = 2.5$.

4.2.2 The type I half-logistic gamma (TIHL-Ga) model

Consider the gamma distribution with shape parameter $\beta_1 > 0$ and scale parameter $\beta_2 > 0$, where the pdf and cdf (for $x > 0$) are

$$g(x;\beta_1,\beta_2) = \frac{\beta_2^{\beta_1}}{\Gamma(\beta_1)}\, x^{\beta_1-1}\, \mathrm{e}^{-\beta_2 x} \quad \text{and} \quad G(x;\beta_1\beta_2) = \frac{\gamma(\beta_1;\beta_2\, x)}{\Gamma(\beta_1)},$$

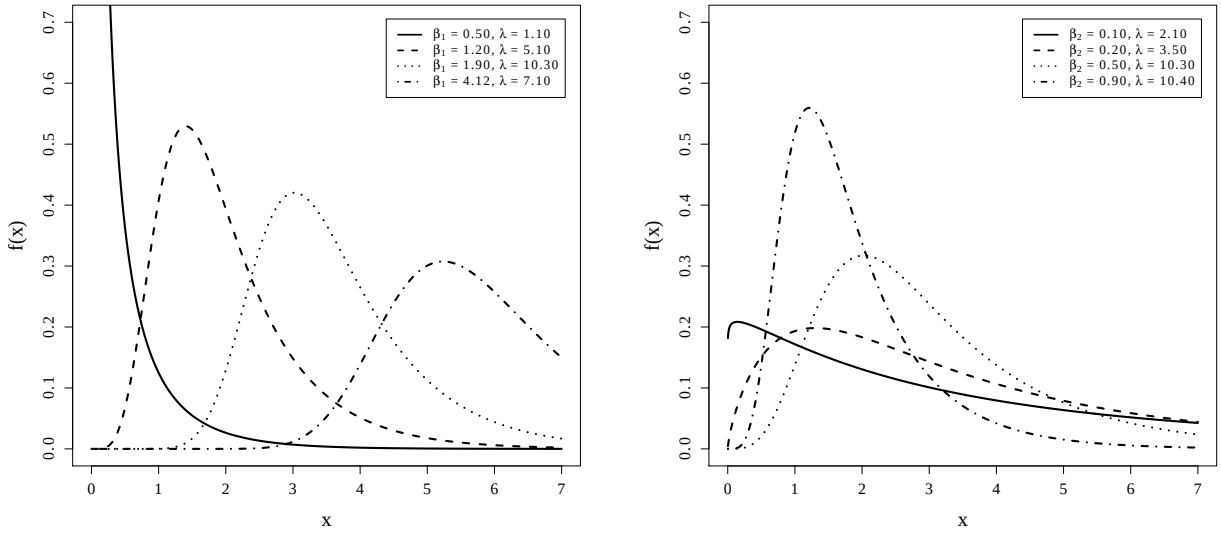
where $\gamma(\beta_1;\beta_2\, x) = \int_0^{\beta_2 x} t^{\beta_1-1}\, \mathrm{e}^{-t}\, \mathrm{d}t$ is the incomplete gamma function. Inserting these expressions in (4.2) gives the TIHL-Ga density function

$$f(x;\lambda,\beta_1,\beta_2) = \frac{2\lambda\frac{\beta_2^{\beta_1}}{\Gamma(\beta_1)}\, x^{\beta_1-1}\, \mathrm{e}^{-\beta_2 x}\big[1-\frac{\gamma(\beta_1,\beta_2 x)}{\Gamma(\beta_1)}\big]^{\lambda-1}}{\Big\{1+\big[1-\frac{\gamma(\beta_1,\beta_2 x)}{\Gamma(\beta_1)}\big]^{\lambda}\Big\}^2}.$$

The hrf of the TIHL-Ga distribution reduces to

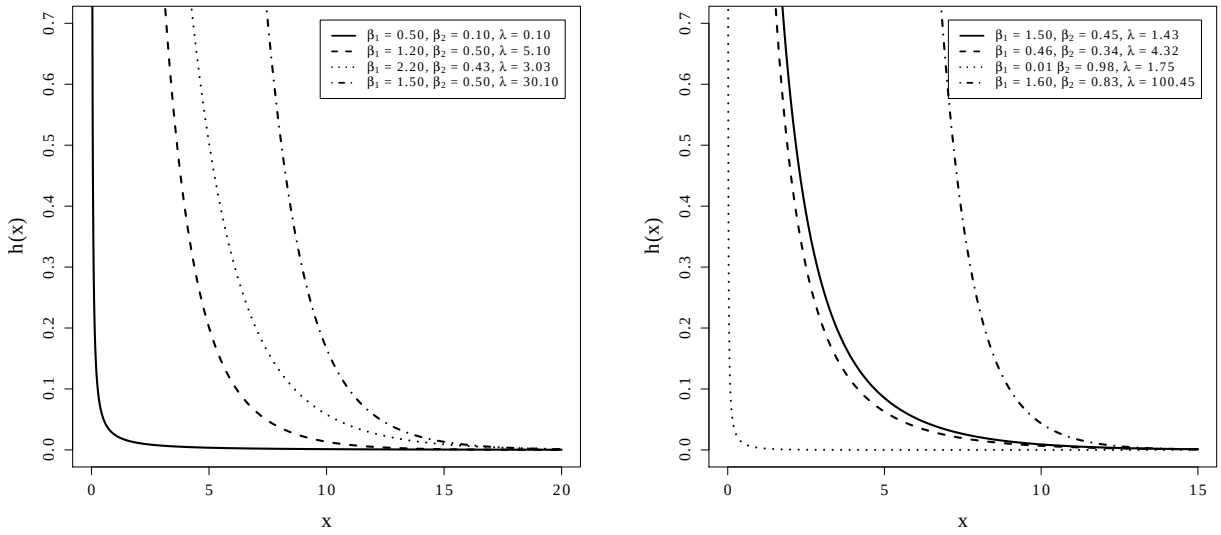
$$h(x;\lambda,\beta_1,\beta_2) = \frac{\lambda\beta_2^{\beta_1} x^{\beta_1-1}\mathrm{e}^{-\beta_2 x}}{\Gamma(\beta_1)\left[\left(1-\frac{\gamma(\beta_1,x\beta_2)}{\Gamma(\beta_1)}\right)^{\lambda}+1\right]\left[1-\frac{\gamma(\beta_1,\beta_2 x)}{\Gamma(\beta_1)}\right]}. \tag{4.6}$$

Plots of the TIHL-Ga density for selected parameter values are displayed in Figures 4.2(a) and 4.2(b). The plots indicate great flexibility of this distribution. It is interesting for modeling data with asymmetry to the right. Figure 4.2(a) and Figure 4.2(b) are given by varying the parameters β_1 and λ with $\beta_2 = 1.1$ fixed and by varying the parameters β_2 and λ with $\beta_1 = 0.5$ fixed. The hrf of the TIHL-Ga distribution has the following forms: monotonically decreasing and inverted bathtub as shown in Figures 4.3(a) and 4.3(b).



(a) (b)

Figure 4.2: The TIHL-Ga densities for (a) $\beta_2 = 1.1$ and for (b) $\beta_1 = 0.5$.



(a) (b)

Figure 4.3: The hrf of the TIHL-Ga model.

4.2.3 The type I half-logistic Fréchet (TIHL-Fr) model

The pdf and cdf of the Fréchet distribution with scale parameter α and shape parameter β are

$$g(x;\alpha,\beta) = \beta\,\alpha^{\beta}\,x^{-\beta-1}\,\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}} \quad \text{and} \quad \mathrm{G(x};\alpha,\beta) = \mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}.$$

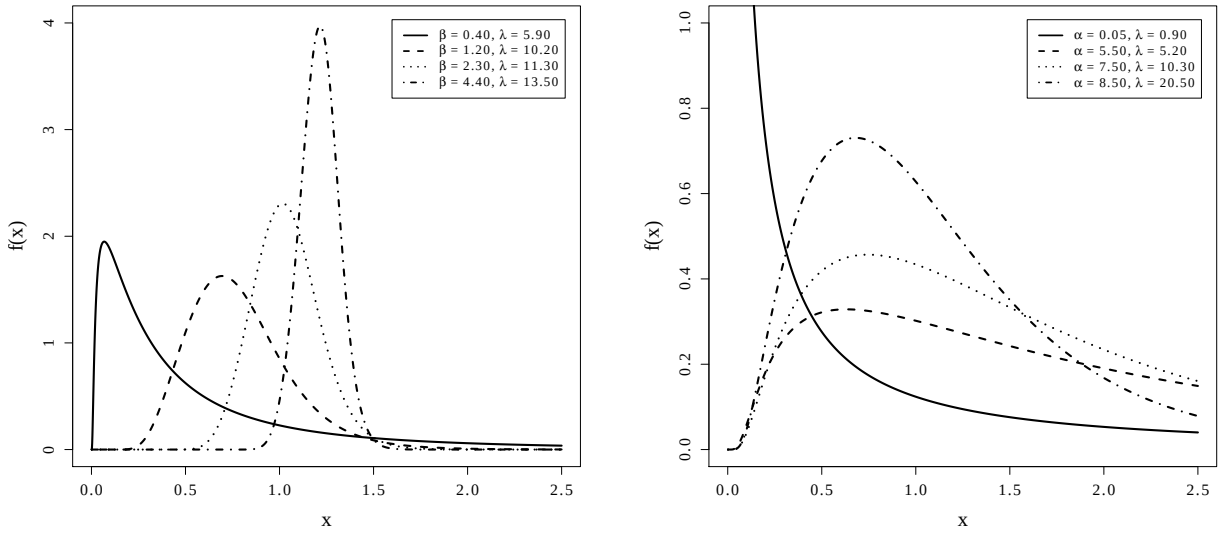
Inserting these expressions in (4.2) yields the TIHL-Fr density function

$$f(x;\lambda,\alpha,\beta) = \frac{2\lambda\beta\,\alpha^{\beta}\,x^{-\beta-1}\,\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}\,[1-\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}]^{\lambda-1}}{\left\{1+[1-\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}]^{\lambda}\right\}^{2}}. \tag{4.7}$$

The hrf of the TIHL-Fr model is given by

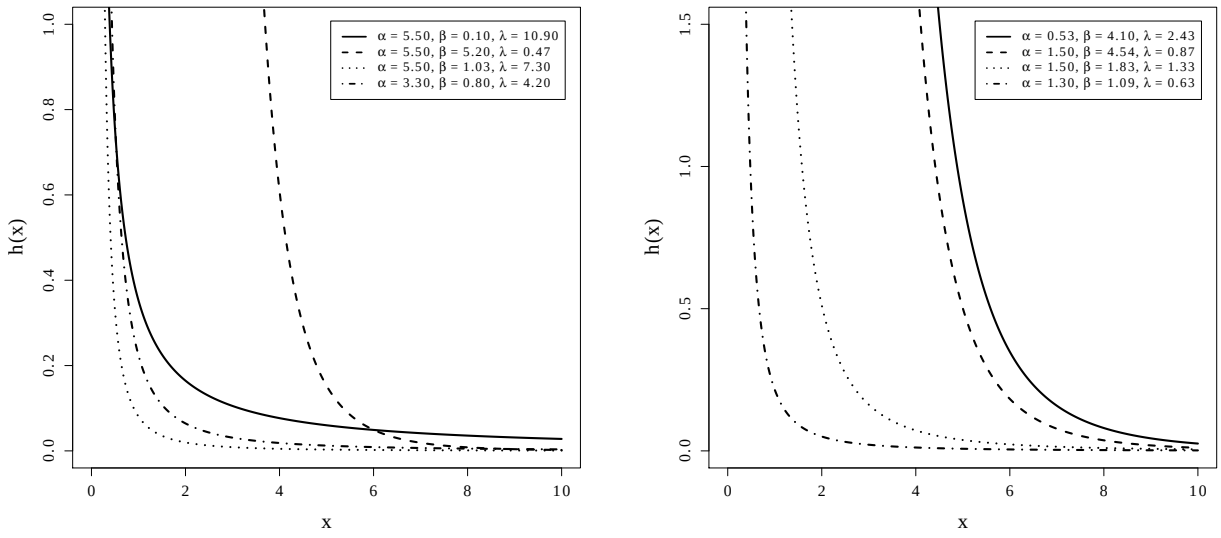
$$h(x;\lambda,\alpha,\beta) = \frac{\lambda\,\beta\,\alpha^{\beta}x^{-\beta-1}\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}}{\left[\left(1-\mathrm{e}^{-\left(\frac{\alpha}{\mathrm{x}}\right)^{\beta}}\right)^{\lambda}+1\right]\left[1-\mathrm{e}^{-(\frac{\alpha}{\mathrm{x}})^{\beta}}\right]}. \tag{4.8}$$

Plots of the TIHL-Fr density function for some parameter values are displayed in Figures 4.4(a) and 4.4(b). The TIHL-Fr(λ,α,β) model is a very competitive model for analysis of lifetime data due to its great flexibility. Figure 4.4(a) and 4.4(b) are constructed by varying β and λ with $\alpha = 1.5$ and by varying α and λ with $\beta = 0.5$. Plots of the hrf of the TIHL-Fr distribution are displayed in Figures 4.5(a) and 4.5(b).



(a) (b)

Figure 4.4: The TIHL-Fr densities for (a) $\alpha = 1.5$ and for (b) $\beta = 0.5$.



(a) (b)

Figure 4.5: The hrf of the TIHL-Fr model.

4.3 Asymptotes and Shapes

Proposition 4.3.1. *The asymptotics of equations (4.1), (4.2) and (4.3) as* $G(x) \to 0$ *are*

$$\begin{aligned} F(x) &\sim \frac{\lambda}{2} G(x) && \text{as} \quad \mathrm{G(x)} \to 0, \\ f(x) &\sim \frac{\lambda}{2} g(x) && \text{as} \quad \mathrm{G(x)} \to 0, \\ h(x) &\sim \frac{\lambda}{2} g(x) && \text{as} \quad \mathrm{G(x)} \to 0. \end{aligned}$$

Proposition 4.3.2. *The asymptotics of equations (4.1), (4.2) and (4.3) as* $x \to \infty$ *are*

$$\begin{aligned} 1 - F(x) &\sim 2\,\bar{G}(x)^{\lambda} && \text{as} \quad \mathrm{x} \to \infty, \\ f(x) &\sim 2\lambda\, g(x)\, \bar{G}(x)^{\lambda-1} && \text{as} \quad \mathrm{x} \to \infty, \\ h(x) &\sim \frac{\lambda\, g(x)}{\bar{G}(x)} && \text{as} \quad \mathrm{x} \to \infty, \end{aligned}$$

wherein $\bar{G}(x) = 1 - G(x)$.

The shapes of the density and hazard rate functions of X can be described analytically. The critical points of the TIHL density function are the roots of the equation:

$$\frac{g'(x)}{g(x)} - (\lambda - 1)\frac{g(x)}{1 - G(x)} + \frac{2\lambda g(x)[1 - G(x)]^{\lambda-1}}{1 + [1 - G(x)]^{\lambda}} = 0. \tag{4.9}$$

There may be more than one root to (4.9). Let $\lambda(x) = \frac{d^2 \log[f(x)]}{dx^2}$. We have

$$\begin{aligned}\lambda(x) &= \frac{g''(x)g(x) - g'(x)^2}{g(x)^2} - (\lambda-1)\frac{g'(x)[1-G(x)] + g(x)^2}{[1-G(x)]^2} \\ &+ \frac{2\lambda g'(x)[1-G(x)]^{\lambda-1}}{1+[1-G(x)]^{\lambda}} - \frac{2\lambda(\lambda-1)g(x)^2[1-G(x)]^{\lambda-2}}{1+[1-G(x)]^{\lambda}} \\ &+ 2\left\{\frac{\lambda g(x)[1-G(x)]^{\lambda-1}}{1+[1-G(x)]^{\lambda}}\right\}^2.\end{aligned}$$

If $x = x_0$ is a root of (4.9) then it corresponds to a local maximum if $\lambda(x) > 0$ for all $x < x_0$ and $\lambda(x) < 0$ for all $x > x_0$. It corresponds to a local minimum if $\lambda(x) < 0$ for all $x < x_0$ and $\lambda(x) > 0$ for all $x > x_0$. It refers to a point of inflexion if either $\lambda(x) > 0$ for all $x \neq x_0$ or $\lambda(x) < 0$ for all $x \neq x_0$.

The critical points of $h(x)$ are obtained from

$$\frac{g'(x)}{g(x)} + \frac{g(x)}{1-G(x)} + \frac{\lambda g(x)[1-G(x)]^{\lambda-1}}{1+[1-G(x)]^{\lambda}} = 0. \tag{4.10}$$

There may be more than one root to (4.10). Let $\tau(x) = d^2 \log[h(x)]/dx^2$. We have

$$\begin{aligned}\tau(x) &= \frac{g''(x)g(x) - g'(x)^2}{g(x)^2} + \frac{g'(x)[1-G(x)] + g(x)^2}{[1-G(x)]^2} \\ &+ \frac{\lambda g'(x)[1-G(x)]^{\lambda-1}}{1+[1-G(x)]^{\lambda}} - \frac{\lambda(\lambda-1)g(x)^2[1-G(x)]^{\lambda-2}}{1+[1-G(x)]^{\lambda}} \\ &+ \left\{\frac{\lambda g(x)[1-G(x)]^{\lambda-1}}{1+[1-G(x)]^{\lambda}}\right\}^2.\end{aligned}$$

If $x = x_0$ is a root of (4.10) then it refers to a local maximum if $\tau(x) > 0$ for all $x < x_0$ and $\tau(x) < 0$ for all $x > x_0$. It corresponds to a local minimum if $\tau(x) < 0$ for all $x < x_0$ and $\tau(x) > 0$ for all $x > x_0$. It gives an inflexion point if either $\tau(x) > 0$ for all $x \neq x_0$ or $\tau(x) < 0$ for all $x \neq x_0$.

4.4 Useful expansions

We can demonstrate that the cdf (4.1) of X admits the expansion

$$F(x) = \sum_{k=0}^{\infty} b_k \, H_k(x), \tag{4.11}$$

where

$$b_k = \sum_{i=0}^{\infty} (-1)^{i+k} \left[\binom{i\lambda}{k} - \binom{(i+1)\lambda}{k}\right] \tag{4.12}$$

and $H_a(x) = G(x)^a$ denotes the exponentiated-G ("exp-G" for short) cdf with power parameter $a > 0$.

The density function of X can be expressed as an infinite linear combination of exp-G densities

$$f(x) = \sum_{k=0}^{\infty} b_{k+1}\, h_{k+1}(x), \tag{4.13}$$

where $h_{k+1}(x) = (k+1)\, G(x)^k\, g(x)$ (for $k \geq 0$) is the exp-G density with power parameter $k+1$. Equation (4.13) reveals that the TIHL density function is a linear combination of exp-G density functions. Thus, some mathematical properties of the new model can be derived from those exp-G properties. For example, the ordinary and incomplete moments and moment generating function (mgf) of X follow from those exp-G quantities.

We provide some mathematical properties of X in the next sections. Established algebraic expansions to determine some structural properties of the TIHL family can be more efficient than computing those directly by numerical integration of its density function, which can be prone to rounding off errors among others. The formulae derived throughout the chapter can be easily handled in softwares such as Mathematica and Maple.

4.5 Generating function

In this section, we provide two formulae for the mgf $M(t) = E(\mathrm{e}^{\mathrm{t\,X}})$ of X. A first formula comes from (4.13) as

$$M(t) = \sum_{k=0}^{\infty} b_{k+1}\, M_{k+1}(t), \tag{4.14}$$

where $M_{k+1}(t)$ is the generating function of the exp-G distribution with power parameter $k+1$. Hence, $M(t)$ can be determined from the exp-G generating function.

A second formula for $M(t)$ can be derived from equation (4.14) as

$$M(t) = \sum_{k=0}^{\infty} (k+1)\, b_{k+1}\, \rho_k(t), \tag{4.15}$$

where the quantity $\rho_k(t)$ is given by

$$\rho_k(t) = \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{tx}}\, \mathrm{G(x)^k}\, \mathrm{g(x)dx} = \int_0^1 \exp[\mathrm{t\, Q_G(u)}]\, \mathrm{u^k du}. \tag{4.16}$$

We can derive the mgf's of several TIHL-G distributions directly from equation (4.16). For example, the mgf's of the type I half-logistic exponential (with parameter λ and $t < \lambda^{-1}$) and type I half-logistic Pareto (with parameter $\nu > 1$) distributions are

$$M(t) = \sum_{k=0}^{\infty} (k+1)\, b_{k+1}\; B(k+1, 1-\lambda\, t)$$

and

$$M(t) = \mathrm{e}^{-\mathrm{t}} \sum_{\mathrm{k,r=0}}^{\infty} \frac{(\mathrm{k+1})\, \mathrm{b_{k+1}}\, \mathrm{B}(\mathrm{k+1}, 1-\mathrm{r}\nu^{-1})}{\mathrm{r!}}\, \mathrm{t^r},$$

respectively.

4.6 Estimation

Several approaches for parameter estimation were proposed in the literature but the maximum likelihood method is the most commonly employed. The maximum likelihood estimators (MLEs) enjoy desirable properties and can be used when constructing confidence intervals and regions and also in test statistics. The normal approximation for these estimators in large sample distribution theory is easily handled either analytically or numerically. So, we consider the estimation of the unknown parameters for this family from complete samples only by maximum likelihood. Let $x_1, \ldots, x_n$ be observed values from the TIHL-G distribution with parameters λ and $\boldsymbol{\xi}$. Let $\Theta = (\lambda, \boldsymbol{\xi})^\top$ be the $r \times 1$ parameter vector. The total log-likelihood function for Θ is given by

$$\begin{aligned} \ell_n = \ell_n(\Theta) &= n \log[2\lambda] + \sum_{i=1}^{n} \log\left[g(x_i;\boldsymbol{\xi})\right] + (\lambda - 1)\sum_{i=1}^{n} \log\left[G(x_i;\boldsymbol{\xi})\right] \\ &- 2\sum_{i=1}^{n} \log\left\{1 + \left[1 - G(x_i;\boldsymbol{\xi})\right]^{\lambda}\right\}. \end{aligned} \tag{4.17}$$

Equation (4.17) can be maximized either directly by using the `R` (`optim` function) R Core Team (2016), `SAS` (`PROC NLMIXED`), `Ox` program (sub-routine `MaxBFGS`) [see Doornik and Ooms (2007)] or `Julia` language through the `optimize` function of the `NLopt` package Bezanson et al. (2012) or by solving the nonlinear likelihood equations obtained by differentiating (4.17). The components of the score function $U_n(\Theta) = (\partial\ell_n/\partial a, \partial\ell_n/\partial b, \partial\ell_n/\partial p, \partial\ell_n/\partial\boldsymbol{\xi})^\top$ are given by

$$\frac{\partial \ell_n}{\partial \lambda} = \frac{n}{\lambda} + \sum_{i=1}^{n} \log[1 - G(x_i,\boldsymbol{\xi})] - 2\sum_{i=1}^{n} \frac{[1 - G(x_i,\boldsymbol{\xi})]^{\lambda} \log[1 - G(x_i,\boldsymbol{\xi})]}{1 + [1 - G(x_i,\boldsymbol{\xi})]^{\lambda}}$$

and

$$\frac{\partial \ell_n}{\partial \boldsymbol{\xi}} = \sum_{i=1}^{n} \frac{g^{(\xi)}(x_i,\boldsymbol{\xi})}{g(x_i,\boldsymbol{\xi})} + (1-\lambda)\sum_{i=1}^{n} \frac{G^{(\xi)}(x_i,\boldsymbol{\xi})}{1 - G(x_i,\boldsymbol{\xi})} + 2\lambda\sum_{i=1}^{n} \frac{G^{(\xi)}(x_i,\boldsymbol{\xi})[1 - G(x_i,\boldsymbol{\xi})]^{\lambda-1}}{1 + [1 - G(x_i,\boldsymbol{\xi})]^{\lambda}},$$

where $h^{(\xi)}(\cdot)$ means the derivative of the function h with respect to ξ. Likelihood ratio tests can be performed for the proposed family of distributions in the usual way.

4.7 Applications

Here, we provide two applications to real data in order to illustrate the potentiality of the TIHL family. In both applications, we consider the TIHL-Weibull (TIHL-W) distribution. In the applications, we use the `AdequacyModel` package version 1.0.8 available for the programming language `R`. The package is currently maintained by one of the authors of this paper and more information can be obtained from `http://cran.rstudio.com/web/packages/AdequacyModel/index.html`. The package is distributed under the terms of the licenses GNU General Public License (GPL $\geq$ 2).

The first data set (Smith and Naylor, 1987) represents the strengths of 1.5 cm glass fibres, measured at the National Physical Laboratory, England. The authors do not provide the unit measurement. The second application take into account the percentage of body fat data determined by underwater weighing and various body circumference measurements for 250 men. For details

Table 4.1: Descriptive statistics.

Statistics	Real data sets	
	Glass Fibres	Body Fat (%)
Mean	1.5068	19.3012
Median	1.5900	19.2500
Mode	1.7000	22.5000
Variance	0.1051	67.7355
Skewness	-0.8999	0.1953
Kurtosis	0.9238	-0.3815
Maximum	2.2400	47.5000
Minimum	0.5500	3.0000
n	63	250

about the data set, see `http://lib.stat.cmu.edu/datasets/`. Table 4.1 gives some descriptive statistics for the two data sets.

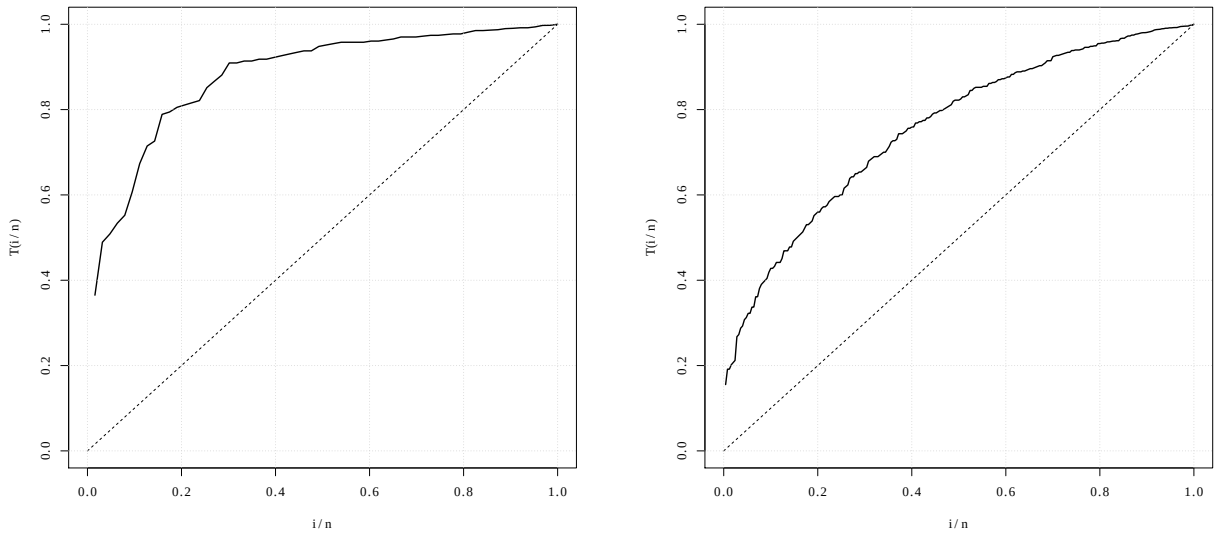
In situations, where the data are censored or uncensored, we can obtain qualitative information about the hrf by means of plot analysis. We emphasize that the data sets here are uncensored. For uncensored data, the total time in test (TTT) plot proposed by Aarset (1987) may be used. Let T be a random variable with non-negative values, which represents the survival time. The TTT curve is constructed by plotting $G(r/n) = [(\sum_{i=1}^{r} T_{i:n}) + (n-r)T_{r:n}]/(\sum_{i=1}^{n} T_{i:n})$ versus r/n $(r = 1, \ldots, n)$, where $T_{i:n}$, for $i = 1, \ldots, n$, are the order statistics of the sample (see Mudholkar and Hutson, 1996). The plots can be easily obtained using the `TTT` function of the `AdequacyModel` package. For more details, see `help(TTT)`. The TTT plots for the data sets in this study are given in Figures 4.6(a) and 4.6(b). For both plots, the TTT curve is concave, which according to Aarset (1987), provides evidence that a monotonic increasing hrf is adequate. Figures 4.7(a) and 4.7(b) display the fitted densities to the current data obtained in a nonparametric manner using the kernel density estimation with the Gaussian filter. Let $X_1, \ldots, X_n$ be a random vector of random variables independent and identically distributed where each variable follows an unknown f distribution. The kernel density estimator is given by

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{4.18}$$

where $K(\cdot)$ is the kernel function usually symmetrical and $\int_{-\infty}^{\infty} K(x)dx = 1$, and $h > 0$ is a smoothing parameter known in literature as bandwidth. Numerous kernel functions are adopted in the literature. The normal standard distribution is the most widely used because it has convenient mathematical properties. Silverman (1986) demonstrated that for the K standard normal, the bandwidth ideal is $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\,\hat{\sigma}\,n^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the sample.

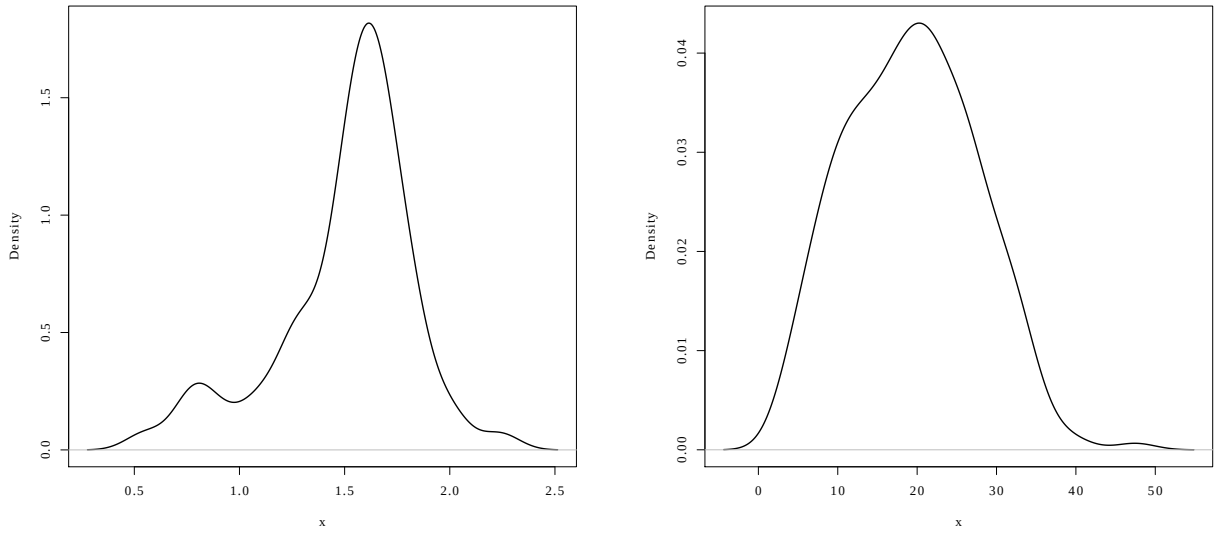
We compare the fits of five distributions to two real data sets, namely the TIHL-W, gamma, Kumaraswamy Weibull (see Cordeiro et al., 2010) and Kumarasuamy Weibull Poisson defined by

$$\begin{aligned} f(x) &= \frac{\lambda\, a\, b\, c\, \beta^c}{\mathrm{e}^{\lambda} - 1}\, x^{c-1} \left[1 - \mathrm{e}^{-(\beta x)^c}\right]^{a-1} \left\{1 - \left[1 - \mathrm{e}^{-(\beta x)^c}\right]^a\right\}^{b-1} \times \\ &\quad \exp\left[\lambda \left\{1 - \left[1 - \mathrm{e}^{-(\beta x)^c}\right]^a\right\}^b - (\beta x)^c\right], \end{aligned} \tag{4.19}$$



(a) (b)

Figure 4.6: The TTT plot for: (a) the glass fibres data (b) and percentage of body fat.



(a) (b)

Figure 4.7: Gaussian kernel density estimation for: (a) the glass fibres data and (b) for percentage of body fat.

where λ, a, b, c and β are non-negative constants. The MLEs of the model parameters are obtained by simulated annealing method, this being a generic probabilistic meta-heuristic method for the global optimization problem proposed by Kirkpatrick and Vecchi (1983). We choose the simulated annealing method because of the complexity of the log-likelihood functions for some distributions.

Some classes of distributions that have been proposed recently in the literature, in general, have log-likelihood functions with approximately planar regions which complicates the optimization by traditional methods (quasi-Newton methods) as is the case of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm proposed by Broydenet et al. (1970). We believe that a heuristic approach offers good solutions to these problems. Optimization by simulated annealing can be easily obtained using the `optim` function of the `R` language by minimizing the negative log-likelihood.

In order to verify which distribution fits better the current data sets, we consider the Cramér-von Mises (W^*) and Anderson-Darling (A^*) statistics described by Chen and Balakrishnan (1995). If we have a random sample $(x_1, \ldots, x_n)$ with empirical distribution function $F_n(x)$, we use these statistics to test if the sample comes from a specified distribution. They are given by

$$W^* = \left\{ n \int_{-\infty}^{+\infty} \{F_n(x) - F(x;\widehat{\theta}_n)\}^2 dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.5}{n}\right) = W^2 \left(1 + \frac{0.5}{n}\right), \tag{4.20}$$

$$\begin{aligned} A^* &= \left\{ n \int_{-\infty}^{+\infty} \frac{\{F_n(x) - F(x;\widehat{\theta}_n)\}^2}{\{F(x;\widehat{\theta})(1 - F(x;\widehat{\theta}_n))\}} dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right) \\ &= A^2 \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right), \end{aligned} \tag{4.21}$$

respectively, where $F_n(x)$ is the empirical distribution function, $F(x;\widehat{\theta}_n)$ is the postulated cdf evaluated at the MLE $\widehat{\theta}_n$ of θ. The statistics W^* and A^* are given by the differences of $F_n(x)$ and $F(x;\hat{\theta}_n)$. The lower are their values more evidence we have that $F(x;\hat{\theta}_n)$ generates the sample. The details to compute the statistics W^* and A^* are discussed by Chen and Balakrishnan.

Table 4.2 lists the MLEs (and the corresponding standard errors in parentheses) of the unknown parameters for the lifetime models fitted to the glass fiber data, whereas Table 4.3 lists those values for the percentage body fat data. The MLEs and their standard errors can also be calculated through the `goodness.fit` function of the `AdequacyModel` package. In addition to the MLEs and standard errors, the `goodness.fit` function also provides various goodness-of-fit statistics such as the Cramér-von Mises (W^*) and Anderson-Darling (A^*) statistics Chen and Balakrishnan (1995), Consistent Akaike Information Criterion (CAIC) Akaike (1974), Bayesian Information Criterion (BIC) Schwarz (1978) and Hannan-Quinn Information Criterion (HQIC) Hannan and Quinn (1979). These statistics can be used to assess the adequacy of the fitted distributions to the real data sets.

The MLEs are obtained by simulated annealing and the corrected biases using the bootstrap method Efron and Tibshirani (1973). Let B be the number of bootstrap samples obtained from the data with replacement (nonparametric bootstrap) and

$$\widehat{\theta}^* = \sum_{b=1}^{B} \widehat{\theta}^*(b)/B, \tag{4.22}$$

where $\widehat{\theta}^*(b)$ is the vector of the estimates of the model parameters obtained by the simulated annealing method for each of the B samples, for $b = 1, \ldots, B$. The estimated biases $\widehat{bias}_B = \widehat{\theta}^* - \widehat{\theta}$ are given by Efron and Tibshirani (p. 225, 1973). Thus, the corrected MLEs by bootstrap can be expressed as

$$\widehat{\theta}_c = \widehat{\theta} - \widehat{bias}_B. \tag{4.23}$$

We adopt the nonparametric bootstrap for calculating the estimated standard errors taking the following steps:

1. Generate B independent random samples $(x_1^*, \ldots, x_B^*)$ with replacement from the original sample $(x_1, \ldots, x_n)$ and calculate $\widehat{\theta}^*(b)$, for $b = 1, \ldots, B$;

2. Calculate the bootstrap estimate of the standard error of $\widehat{\theta}$ by

$$\widehat{se}_B(\widehat{\theta}) = \left\{ \sum_{b=1}^{B} [\widehat{\theta}^*(b) - \widehat{\theta}^*(\cdot)]^2/(B-1) \right\}^{1/2}, \tag{4.24}$$

   where $\widehat{\theta}^*(\cdot) = \sum_{b=1}^{B} \widehat{\theta}^*(b)/B$.

We can obtain the standard errors of the corrected estimates by bootstrap $(\widehat{\theta}_c)$. However, the computational cost is higher. The estimates of the standard errors are calculated for $\widehat{\theta}$ and listed in Tables 4.2 and 4.3.

The plots displayed in Figures 4.8(a) and 4.8(b) indicate that the TIHL-W distribution provides the best fit compared to the other fitted distributions. We note the good adequacy of the fitted TIHL-W distribution in the plots given in Figures 4.9(a) and 4.9(b). Figure 12 gives the estimates of the TIHL-W survival function and Kaplan-Meier (K-M) for the two data sets.

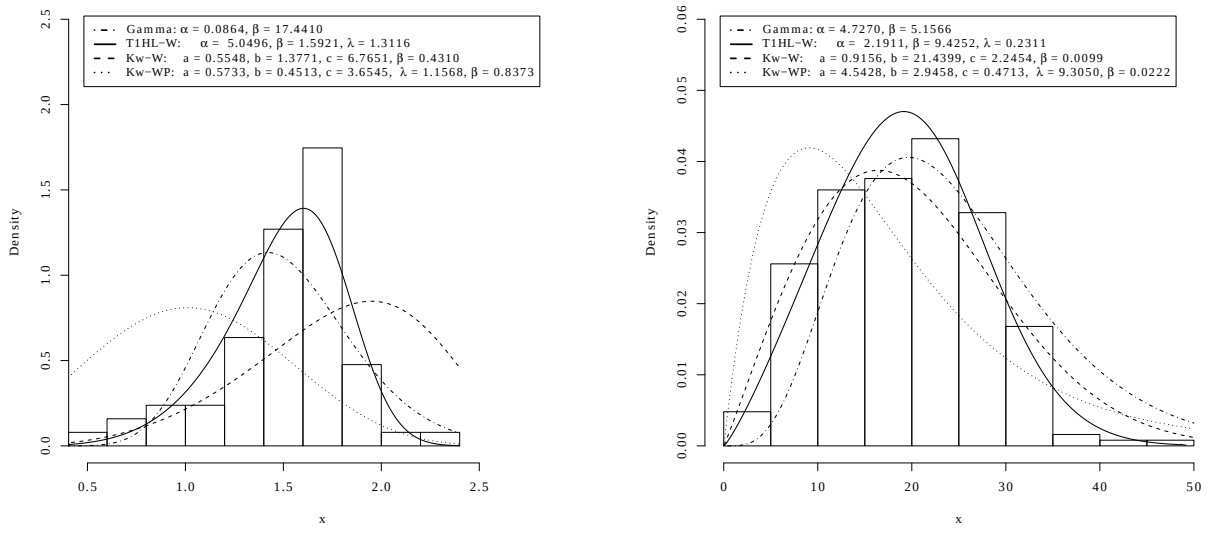
Table 4.4 provides the goodness-of-fit statistics obtained by the `goodness.fit` function of the `AdequacyModel` script of the `R` language. For more details on the `goodness.fit` function, run `help(goodness.fit)`. There is a great adjustment of the TIHL-W distribution for both data sets. Such statistics are highlighted in Table 4.4. They confirm the good fit of the TIHL-W model as indicated by the previous plots.

Table 4.2: MLEs for the of glass fiber data (standard errors in parentheses).

Distribution	**Estimates** - ($\widehat{\theta}_c$ and $\widehat{se}_B(\widehat{\theta})$, with $B = 1000$)				
Gamma (α, β)	0.0864	17.4410			
	(0.0042)	(0.0021)			
TIHL-W (α, β, λ)	5.0496	1.5921	1.3116		
	(0.0019)	(0.0111)	(0.0188)		
Kw-W (a, b, c, β)	0.5548	1.3771	6.7651	0.4310	
	(0.0002)	(0.0086)	(0.0667)	(0.0090)	
Kw-WP (a, b, c, λ, β)	0.5733	0.4513	3.6545	1.1568	0.8373
	(0.0055)	(0.0032)	(0.0002)	(0.0097)	(0.0551)

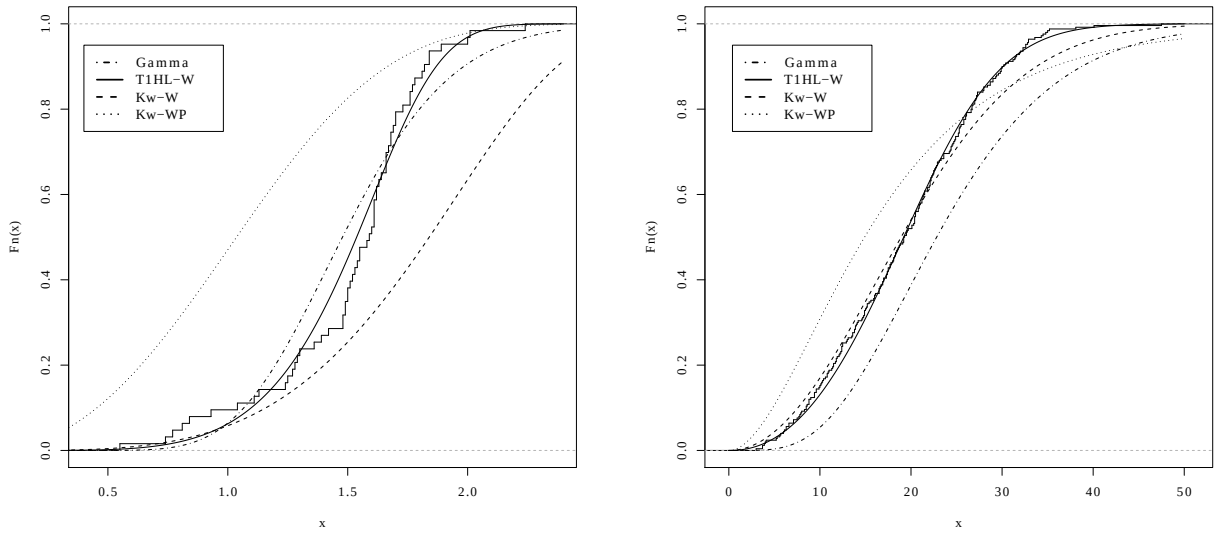
Table 4.3: MLEs for the percentage of body fat data (standard errors in parentheses).

Distribution	**Estimates** - ($\widehat{\theta}_c$ and $\widehat{se}_B(\widehat{\theta})$, com $B = 1000$)				
Gamma (α, β)	4.7270	5.1566			
	(0.0321)	(0.0243)			
TIHL-W (α, β, λ)	2.1911	9.4252	0.2311		
	(0.0001)	(0.0075)	(0.0044)		
Kw-W (a, b, c, β)	0.9156	21.4399	2.2454	0.0099	
	(0.0437)	(0.0054)	(0.0077)	(0.0043)	
Kw-WP (a, b, c, λ, β)	4.5428	2.9458	0.4713	9.3050	0.0222
	(0.1223)	(0.0673)	(0.0445)	(0.0066)	(0.0059)



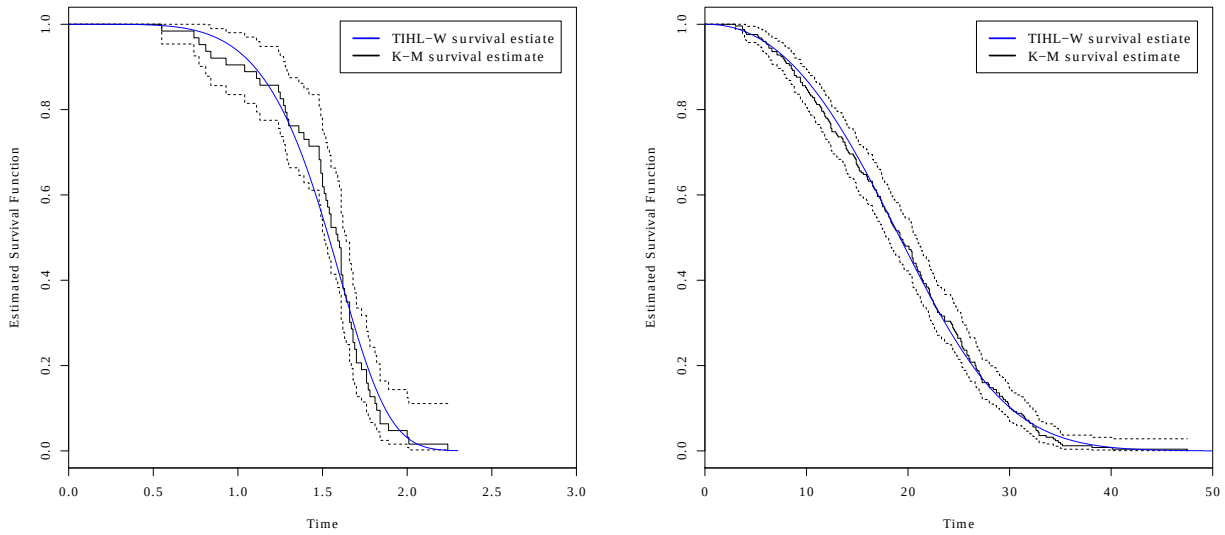
(a) (b)

Figure 4.8: Estimates of the density functions for the: (a) glass fibres data and (b) percentage of body fat.



(a) (b)

Figure 4.9: Estimates of the distribution function and empirical distribution for the: (a) glass fibre data and (b) percentage of body fat data.



(a) (b)

Figure 4.10: Kaplan-Meier estimates (K-M estimates) compared with the TIHL-W survival estimates for the: (a) glass fibre data and (b) percentage of body fat data. The confidence intervals are considered 95%.

Table 4.4: Goodness-of-fit statistics for the fits to the glass fibre data denoted by (I) and percentage of body fat data denoted by (II).

Data set	**Distribution**	A^*	W^*	AIC	CAIC	BIC	HQIC
I	Gamma (α, β)	3.1174	0.5684	51.9031	52.1031	56.1893	53.5889
	TIHL-W (α, β, λ)	**1.0059**	**0.1821**	**33.6543**	**34.0610**	**40.0837**	**36.1830**
	Kw-W (a, b, c, β)	1.7572	0.3210	89.8182	90.5078	98.3907	93.1898
	Kw-WP (a, b, c, λ, β)	1.7877	0.3263	118.3556	119.4082	129.0713	122.5701
II	Gamma (α, β)	2.0318	0.3362	1840.1960	1840.2440	1847.2390	1843.0300
	TIHL-W (α, β, λ)	**0.3149**	**0.0437**	**1758.1510**	**1758.2480**	**1768.7150**	**1762.4020**
	Kw-W (a, b, c, β)	0.7447	0.1201	1780.6800	1780.8430	1794.7660	1786.3490
	Kw-WP (a, b, c, λ, β)	2.4070	0.3983	1864.2450	1864.4910	1881.8520	1871.3310

4.8 Concluding Remarks

In this chapter, we propose a new class of distributions called the *type I half-logistic* (TIHL) family. Some special cases are presented. We provide a mixture representation in terms of exponentiated distributions which is important to derive various of its structural properties in full generality. Some general mathematical properties such as the shapes, asymptotics, ordinary and incomplete moments, quantile and generating functions, entropies and order statistics are investigated. For each baseline distribution, our results can be easily adapted to obtain the main structural properties of the generated distribution. We provide an extension to the bivariate case. The estimation of the model parameters is approached by the method of maximum likelihood and bootstrap is considered to estimate the biases and standard errors of the maximum likelihood estimators. We perform two

applications by means of uncensored real data sets to demonstrate the the new family is a very competitive generator to other classes of distributions.

References

[1] Aarset, M V. How to identify bathtub hazard rate. *IEEE Transactions on Reliability*, v. 36, p. 106–108, 1987.

[2] Akaike, H. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, v. 19, p. 716–723, 1974.

[3] Bezanson, J, Karpinsk, S, Shah, V B and Edelman, A. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012.

[4] Broyden, C G, Fletcher, R , Goldfarb, R, D. DF Shanno J. *Inst. Math. Appl*, v. 6, p. 76, 1970.

[5] Chen, G, and Balakrishnan, N. A general purpose approximate goodness-of-fit test. *Journal of Quality Technology*. v. 27. p. 154–161, 1995.

[6] Cordeiro, Gauss M, Ortega, Edwin MM, and Nadarajah, Saralees. The Kumaraswamy Weibull distribution with application to failure data. *Journal of the Franklin Institute*, v. 347, p. 1399–1429, 2010.

[7] Doornik, J A and Ooms, M. Introduction to Ox: An Object-Oriented Matrix Language, *Timberlake Consultants Ltd*, 2007.

[8] Efron, B and Tibshirani, Robert. An introduction to the bootstrap. v. 57, Ed. 1, CRC Press, New York, 1993.

[9] Gupta, R D and Kundu, D. *Exponentiated exponential family: an alternative to gamma and Weibull distributions*. v 43, p. 117–130, 2001.

[10] Hannan, E J and Quinn, B G. The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, p. 190–195, 1979.

[11] Kirkpatrick, S and Vecchi, M P. Optimization by simmulated annealing. *Science*, v. 220, p. 671–680, 1983.

[12] Mudholkar, G S and Hutson, A D. The exponentiated Weibull family: some properties and a flood data application. *Communications in Statistics–Theory and Methods*, v. 25, p. 3059–3083, 1996.

[13] Nadarajah, S and Kotz, S. The exponentiated type distributions. *Acta Applicandae Mathematica*, v. 92, p. 97–111, 2006.

[14] R Core Team. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing*, `https://www.R-project.org/`, Austria, 2016.

[15] Rényi, A. On measures of entropy and information. *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, p. 547–561, 1961.

[16] Schwarz, Gideon. Estimating the dimension of a model. *The annals of statistics*, v. 6, 461–464, 1978.

[17] Silverman, B W. Density estimation for statistics and data analysis. Ed. *CRC Press*, v. 26, 1986.

[18] Shannon, C E. Bell system tech. j. 27 (1948) 379; ce shannon. *Bell System Tech*, v. 27, p. 623, 1948.

[19] Smith, R L and J C Naylor. A comparison of maximum likelihood and Bayesian estimators for the three-parameter Weibull distribution. *Applied Statistics*, p. 358–369, 1987.

CHAPTER 5

AdequacyModel: An R Package for modeling probability distributions and general optimization

Abstract: Statistical analysis of lifetime data is an important topic in biomedical science, reliability, engineering, social sciences and several other areas. An important step in the treatment of lifetime data is the proposal of more flexible models, which provide a good representation for both nature of data and the shape of its empirical distribution. For some of these new models, however, it is very difficult to obtain the maximum likelihood estimates, since the corresponding likelihood functions have nearly flat regions, which make many derivative-based optimization methods well-established in the literature unsuitable for obtaining these estimates. In such cases, the use of meta-heuristic optimization algorithms usually provides good solutions to this class of problems. On this subject, this chapter introduces the `AdequacyModel` package for the `R` statistical computing environment, which is available on the Comprehensive `R` Archive Network (CRAN). The main application concerns to a new robust optimization package with two major contributions. The first one refers to the assessment of the adequacy of probabilistic models through a combination of several statistics, which measure the relative quality of statistical models for a given data set. The second one provides a general optimization method based on meta-heuristic functions for maximizing or minimizing an arbitrary objective function. It is important to emphasize that the proposed package can be used not only in statistics but in physics and mathematics as demonstrate in several examples. The `AdequacyModel` package has been cited very frequently by papers related to new lifetime distributions. This package is in version 2.0.0 and has been continuously updated.

Keywords: `AdequacyModel`. goodness-of-fit. lifetime. maximum likelihood. optimization. `R`.

5.1 Introduction

Probability distributions for lifetime data analysis are continually evolving due to increasing amount of available information from real phenomena and the development of statistical software packages to analyze these data. In this context, new probabilistic models for parametric inference and applications are proposed in parallel with computer based tools, which allow for using more complex distributions with a larger number of parameters to better study sizeable masses of data.

In the last two decades, several methods have been proposed to generate lifetime distributions in the literature. In addition to generalizing the traditional models, the relevance of these new distributions relies on the fact that, according to the problem, each of them can provide better fit to a given data set. New distributions often result from a modification of a baseline random variable

Z by (i) linear transformation, (ii) power transformation (e.g. the Weibull distribution is obtained from the exponential distribution), (iii) non-linear transformation (e.g. the lognormal distribution comes from the normal distribution), (iv) log transformation (e.g. the log Weibull distribution, also known as the type 1 extreme value distribution), and (v) inverse transformation (e.g. the inverse Weibull and inverse gamma models).

Furthermore, several other methods have been proposed in order to generate more flexible models, which furnish a good representation for both nature of data and the shape of its empirical distribution. Some well-known techniques for generating continuous univariate distributions include the power transformation (exponentiated class of distributions), the Marshall-Olkin family, the beta and Kumaraswamy generalized families and the compounding models.

Next, we provide a brief summary of the above methods, which are related to the statistical package studied in this chapter. The subtitles are followed by reference papers. In Section 5.1.2, we remark about heuristics algorithms for NP-complete problems, which include the swarm intelligence paradigm. In Section 5.2, the **P**article **S**warm **O**ptimization (PSO) method is implemented in the `AdequacyModel` package for the `R` statistical environment. Further, we present the `pso` function with several examples. We compare the results obtained by the `pso` function with those obtained by traditional global search approaches, such as the quasi-Newton BFGS [see Broyden et al. (1970)], Nelder-Mead [see Nelder and Mead (1965)] and simulated-annealing [see Bélisle (p. 890, 1992)] methods. For comparison purposes, we use several functions, which are well-known by their optimization difficulties. Section 5.3 provides several adequacy measures, which give the relative quality of the competing statistical models for a given data set. Finally, concluding remarks and the current package version are addressed in Section 5.4.

5.1.1 A short review on recent families of distributions

Marshall-Olkin family of distributions. Paper by Marshall and Olkin (1997).

Marshall and Olkin (1997) introduced an interesting method of adding a new parameter to an existing distribution. The resulting distribution, known as the Marshall-Olkin ($\mathcal{MO}$) extended distribution, includes the parent distribution as a special case and gives more flexibility to model several types of data. Let $\bar{G}(\cdot) = 1 - G(\cdot)$ denote the survival function of a continuous random variable X, which depends on a parameter vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)^\top$ of dimension q. Then, the corresponding $\mathcal{MOG}$ distribution has survival function given by

$$\bar{F}(x) = \frac{\alpha\,\bar{G}(x)}{1 - \bar{\alpha}\,\bar{G}(x)} = \frac{\alpha\,\bar{G}(x)}{G(x) + \alpha\,\bar{G}(x)}, \quad x \in \mathbb{R}, \quad \alpha > 0,$$

where $\bar{\alpha} = 1 - \alpha$. If $\alpha = 1$, we have $\bar{F}(\cdot) = \bar{G}(\cdot)$. For more details, the reader is referred to Marshall and Olkin (1997) and Lai (2013).

Exponentiated family of distributions. Paper by Gupta and Kundu (1999).

The exponentiation transform of cumulative distributions can furnish more flexible models. Such procedure generates the so-called exponentiated G ($\mathcal{EG}$) family. This approach consists to add a positive real parameter, say $\alpha > 0$, to a cumulative distribution function (cdf) $G(\cdot)$ by exponentiation, which yields a cdf $G(\cdot)^\alpha$ that usually provides interesting mathematical properties and better fits to data sets in different contexts. Indeed, although this transformation is simple, the generated

distribution from it is richer than the corresponding baseline $G(\cdot)$ and, therefore, it requires a special treatment. This technique of generating continuous univariate distributions has attracted a lot of attention in the last decade mainly after the work by Gupta and Kundu (1999), which proposed the exponentiated exponential distribution. Let $G(\cdot)$ and $g(\cdot)$ be the cdf and probability density function (pdf), respectively, of a known random variable Z (say, a baseline model). A random variable X is said to have the $\mathcal{EG}$ class if its cdf and pdf are given by

$$F(x) = G(x)^{\alpha} \qquad \text{and} \qquad f(x) = \alpha\, g(x)\, G(x)^{\alpha-1}, \qquad \alpha > 0,$$

respectively.

Beta family of distributions. Paper by Eugene, Lee, and Famoye (2002).

Starting from a baseline continuous cdf $G(\cdot)$, which depends on a parameter vector $\boldsymbol{\beta}$ of dimension q, Eugene et al. (2002) defined the beta generalized ($\mathcal{BG}$) family by the cdf

$$F(x) = \frac{1}{B(a,b)} \int_0^{G(x)} \omega^{a-1}(1-\omega)^{b-1}\mathrm{d}\omega, \quad x \in \mathbb{R},$$

where $a > 0$ and $b > 0$ are shape parameters, $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ is the beta function and $\Gamma(a) = \int_0^{\infty} \omega^{a-1}\mathrm{e}^{-\omega}\mathrm{d}\omega$ is the gamma function. The corresponding pdf of the $\mathcal{BG}$ family is given by

$$f(x) = \frac{1}{B(a,b)}\, g(x)\, G(x)^{a-1}\, [1-G(x)]^{b-1}, \quad x \in \mathbb{R}.$$

The $\mathcal{BG}$ family has received considerable attention over the last years, in particular after the works of Eugene et al. (2002) and Jones (2009). Since the publication of these seminal papers, many other generalizations have been proposed. Razzaghi (2009) used the beta normal ($\mathcal{BN}$) distribution in dose-response modeling and risk assessment for quantitative responses. Recently, Rego et al. (2012) provided a better foundation for the $\mathcal{BN}$ distribution and presented some properties and an analytical study of its bimodality. Another interesting application of the $\mathcal{BN}$ distribution can be found in SAR image processing; see Cintra et al. (2014), who introduced the beta generalized normal distribution defined by compounding the beta and generalized normal distributions.

Kumaraswamy family of distributions. Paper by Cordeiro and de Castro (2011).

Kumaraswamy (1980) proposed a distribution for double bounded random processes with hydrological applications. The Kumaraswamy ($\mathcal{K}w$) distribution received considerable interest in hydrology and related areas, see Fletcher and Ponnambalam (1996) and Seifi et al. (2000). Its cdf and pdf with two shape parameters $a > 0$ and $b > 0$ are given by

$$F(x) = 1 - (1-x^a)^b \qquad \text{and} \qquad f(x) = a\, b\, x^{a-1}(1-x^a)^{b-1}, \quad x \in (0,1),$$

respectively. Based on the $\mathcal{K}w$ distribution, Cordeiro and Castro (2011) defined the Kumaraswamy generalized ($\mathcal{K}w\mathcal{G}$) family by

$$F(x) = 1 - [1-G(x)^a]^b, \quad x > 0, \tag{5.1}$$

where $a > 0$ and $b > 0$ are two additional parameters whose role is to provide skewness and to vary tail weights. The pdf corresponding to (5.1) has a very simple form

$$f(x) = a\, b\, g(x)\, G(x)^{a-1}[1-G(x)^a]^{b-1}, \quad x > 0.$$

Compounding family of distributions. Paper by Adamidis and Loukas (1998).

The family of the compounding models is obtained by compounding absolutely continuous and discrete distributions. It was pionnered by Marshall and Olkin (1997) and after that extended by several authors, mainly after the work by Adamidis and Loukas (1998). The compounding lifetime distributions recently added to the statistical literature extended important and well-established lifetime models, based on the exponential, Weibull and gamma distributions. Further, they allow for greater flexibility of the tails and are motivated for industrial applications and biological studies. This type of compounding family arises by combining the power series and lifetime distributions. Let N be a discrete random variable having a zero truncated power series ($\mathcal{PS}$) probability mass function given by $P(N = n) = a_n\,\theta^n/A(\theta)$, where $n \in \mathbb{N}$ and N and T's are independent. The coefficients a_n's depend only on n, $A(\theta) = \sum_{n=1}^{\infty} a_n\,\theta^n$ (for $\theta > 0$) is such that $A(\theta)$ is finite. We define $X = \min\{T_1, \ldots, T_N\}$. Then, the conditional cumulative distribution of $X|N = n$ is $F_{X|N=n}(x) = 1 - [1 - G(x)]^n$ and the marginal cdf of X becomes

$$F(x) = 1 - \frac{1}{A(\theta)} A\left\{\theta\left[1 - G(x)\right]\right\}, \quad x > 0. \tag{5.2}$$

Table 5.1 lists some $\mathcal{PS}$ distributions (truncated at zero) such as the Poisson, logarithmic, geometric and binomial distributions.

Table 5.1: Useful quantities for some PS distributions.

Distribution	a_n	$A(\theta)$	θ
Poisson	$n!^{-1}$	$\mathrm{e}^{\theta} - 1$	$\theta \in (0, \infty)$
Logarithmic	n^{-1}	$-\log(1-\theta)$	$\theta \in (0, 1)$
Geometric	1	$\theta/(1-\theta)$	$\theta \in (0, 1)$
Binomial	$\binom{m}{n}$	$(\theta+1)^m - 1$	$\theta \in (0, 1)$

For some of these new models, however, it is very difficult to obtain the maximum likelihood estimates (MLEs), since the corresponding likelihood functions have nearly flat regions, which make many derivative-based optimization methods unsuitable for obtaining such estimates. In such cases, the use of meta-heuristic optimization algorithms usually provides solutions to this class of problems.

5.1.2 Heuristic algorithms for NP-complete problems

In the computational complexity theory, NP is an acronym for non-deterministic polynomial time, which denotes a set of problems that are not solvable in polynomial time by a non-deterministic Turing machine.

The great importance of such class of problems relies on the fact that it contains many search and optimization problems for which we would like to know if there is a solution. For example, note that the complexity class P (polynomial time) is a subset of NP, which can also contains as specific problems the so-called NP-complete problems, whose solutions are sufficient to deal with any other NP problem in polynomial time. Among the most important NP problems are the Hamiltonian-cycle

problem, which takes a graph G as input and asks whether there is a simple cycle in G that visits each vertex of G exactly once and then returns to its starting vertex; the circuit-sat problem, which takes as input a Boolean circuit with a single output node, and asks whether there is an assignment of values to the circuit's inputs so that its output value is '1'; and the vertex-cover, which is the decision problem that takes a graph G and an integer k as input, and asks whether there is a vertex cover for G containing at most k vertices.

The non-deterministic algorithm is a special type (and quite unrealistic) of an algorithm that "guess correctly" in all steps. Thus, the NP are problems whose solution can be found and verified in time polynomial by a non-deterministic algorithm. In Mathematics, the question whether P = NP or P $\neq$ NP is still an open problem. Without a solution, we think that in certain problems the solution is not currently achieved in polynomial time due to the fact that we do not have yet an algorithm that provide solutions in polynomial time or there is a class of problems whose solutions can not be obtained by a polynomial time model-based. This class of problems are usually called the NP-complete or NP-hard problems, whose solutions are more complex to achieve.

In Statistics, we are usually interested in the problem of search or optimizing a function, i.e. maximize or minimize an objective function, such as, for example, the likelihood function. Note that new families of distributions have been proposed with extra parameters and, in some of these new families, the corresponding likelihood functions have approximately flat regions. In this case, deterministic methods, especially those that make use of derivatives, are unable to find a satisfactory solution. So, heuristic methods for problems of search present several methods that provide reasonable solutions to the NP problems. Among these heuristic methods that look for solutions to problems of search, we mention the paradigm of swarm intelligence and the PSO method.

Swarm intelligence

According to Parsopoulos and Vrahatis (2002), swarm intelligence is an exciting new research field still in its infancy compared to other paradigms in artificial intelligence. It is a branch of artificial intelligence concerned to the study of collective behavior of decentralized and self-organized systems in a social structure. These kind of systems are composed by agents that interact in a small organization (swarm) wherein each individual is a particle.

The main idea behind swarm intelligence is that an isolated particle has a very limited action in search an ideal point for the solution of an NP-complete problem. However, the joint behavior of the particles in the search region shows evidence of artificial intelligence, i.e., the ability to take decisions to respond to changes. In this sense, the swarm intelligence concept arises directly from nature and is based on, for example, the self-organizing exploratory pattern of the schools of fish, flocks of birds and ant colonies. This collective behavior can not be described simply by aggregating the behavior of each element. Such situations have encouraged practitioners to obtain a satisfactory effect in the search for solutions to complex problems by studying methods that promote intelligent behavior through collaboration and competition among individuals.

Swarm-based algorithms have been widely developed in the last decade and the many successful applications in a variety of complex problems make it a very promising, efficient and robust optimization tool, although very simple to implement. The idea is modeling very simple local interactions among individuals from which complex problem-solving behaviors arise.

5.2 Conceptual design of the framework

Particle Swarm Optimization (PSO)

In computer science, the PSO is a computational method for optimization of parametric and multiparametric functions. The PSO algorithm is a meta-heuristic method, which has been providing good solutions for problems of global optimization functions with box-constrained. The use of meta-heuristic methods such as PSO has proved to be useful for maximizing complicated log-likelihood functions without the need for early kick functions as the BFGS, L-BFGS-B, Nelder-Mead and simulated annealing methods. As in most heuristic methods that are inspired by biological phenomena, the PSO method is inspired by the behavior of flying birds. The philosophical idea of the PSO algorithm is based on the collective behavior of birds (particle) in search of food (point of global optimal). This technique was first defined by Eberhart and Kennedy (1995) in a paper published in the Proceedings of the IEEE International Conference on Neural Networks IV. A modification of the PSO algorithm was proposed by Shi and Eberhart (1998). Further details on the philosophy of the PSO method are given in the book Swarm Intelligence [see Kennedy and Eberhart (2001)].

The PSO optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. The movement of the particles in the search space is randomized. Each iteration of the PSO algorithm, there is a leader particle, which is the particle that minimizes the objective function in the corresponding iteration. The remaining particles arranged in the search region will follow the leader particle randomly and sweep the area around this leading particle. In this local search process, another particle may become the new leader particle and the other particles will follow the new leader randomly. Each particle arranged in the search region has a velocity vector and position vector and its movement in the search region is given by changes in these vectors. The PSO algorithm is presented below, where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function to be minimized, S is the number of particles in the swarm (set of feasible points, i.e., search region), each particle having a vector position $x_i \in \mathbb{R}^n$ in the search-space and a vector velocity defined by $v_i \in \mathbb{R}^n$. Let p_i be the best known position of particle i and g the best position of all particles.

1. For each particle $i = 1, \ldots, S$ do:
   - Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
   - Initialize the particle's best known position to its initial position: $p_i \leftarrow\!\shortmid\, x_i$.
   - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrow\!\shortmid\, p_i$.
   - Initialize the particle's velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$.
2. Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
   - For each particle $i = 1, \ldots, S$ do:
     - Pick random numbers: $r_p, r_g \sim U(0, 1)$.
     - For each dimension $d = 1, \ldots, n$ do:
       - Update the particle's velocity: $v_{i,d} \leftarrow\!\shortmid\, \omega\, v_{i,d} + \varphi_p r_p(p_{i,d} - x_{i,d}) + \varphi_g r_g(g_d - x_{i,d})$.
     - Update the particle's position: $x_i \leftarrow\!\shortmid\, x_i + v_i$

- If $f(x_i) < f(p_i)$ do:
  - Update the particle's best known position: $p_i \leftarrow\!\shortmid x_i$
  - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrow\!\shortmid p_i$.

3. Now g holds the best found solution.

The parameter ω is called inertia coefficient and as the name implies controls the inertia of each particle arranged in the search region. The quantities ω_p and ω_g control the acceleration of each particle and are called acceleration coefficients. The PSO algorithm described above implemented in the programming language `R` is presented below.

This algorithm with few modifications will be implemented in the `AdequacyModel` package available on the website of `R`. The algorithm above is quite general and can be applied to maximize any function involving or not a database. Using the `pso` function, a given function is maximized taking into consideration vectors of restrictions delimiting the search-space. In fact, the `pso` function is constructed to minimize any function. However, to maximize f is equivalent to minimize $-f$. A brief description of the `pso` function arguments are listed below:

- `func`: objective function to be minimized;
- `S`: number of particles considered. By default, the number of particles is equal to 150;
- `lim_inf` e `lim_sup`: vectors that restrict the region-search inferiorly and superiorly, respectively.
- `e`: error considered. The algorithm stops if the variance in the last 10 iterations is less than or equal to `e`;
- `data`: by default `data = NULL`, but when the `func` is a log-likelihood function, `data` is a data vector;
- `N`: minimum number of iterations (default `N = 500`);
- `prop`: Proportion of last minimum value that is calculated variance used as a stopping criterion. That is, if the number of iterations is greater or equal to the minimum number of iterations N, calculate the variance of the last values of minimum obtained, where $0 \leq$ `prop` ≤ 1.

One advantage of the PSO method is that we do not need to concern ourselves with initial shots. Problems with initial shots are frequent in methods such as the BFGS when the objective function involves flat or nearly flat regions. To depend on the initial shots provided, we can obtain estimates totally different. In general, this does not occur with great frequency in methodologies of heuristic search, whose update steps embed randomness (generation of pseudo-random number). The example below shows clearly this problem and the use of the `pso` function, especially how to specify the objective function for the argument `func`.

In the `pso` function, it is used a minimization guests to `N`, i.e., the criterion of stopping will only be evaluated if the number of iterations of the PSO algorithm is greater or equal to `N`. The amount of minimum values considered in the calculation of the variance is given by the proportion of minimum values established by the argument prop which by default is prop = 0.2. That is, if the last 20% (`prop = 0.2`) of the minimum values has less variance than or equal to `e`, the algorithm will stop global search, indicating convergence according to the established criteria. This indicates that there was no significant improvements in this proportion of the last iterations. Thus, if the variance is less than or equal to $\varepsilon > 0$ assigned to the argument `e` of the `pso` function, the algorithm will stop the iterations and return the best point that minimizes the objective function.

Example

Initially, we will consider the case of a global search in a univariate function, where we are interested in estimate only a one-dimensional vector. Consider the objective function $f(\theta) = 6 + \theta^2 \sin(14\theta)$. This is a function with some local minima, such that the value of θ that minimizes globally f is equal to 2.3605 and $f(2.3605) = -11.5618$. Figure 5.1 displays the plot of $f(\theta)$, for $\theta \in [-2.5, 2.5]$, such that is highlighted the estimates of the global minimum for the BFGS, Nelder-Mead, simulated annealing and PSO methods.

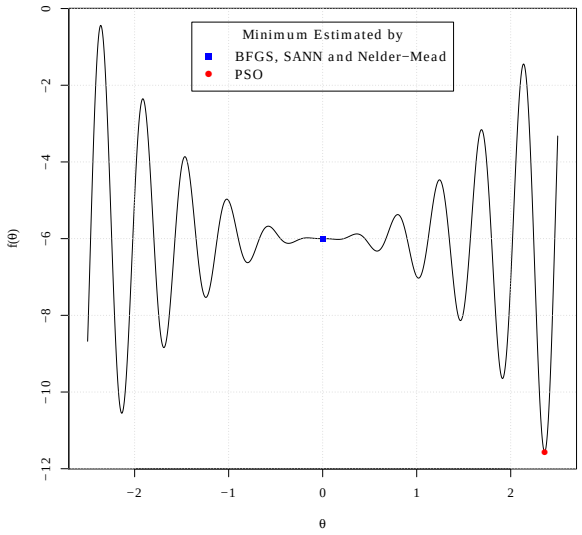


Figure 5.1: Function $f(\theta) = 6 + \theta^2 \sin(14\theta)$ with global minimum estimates.

It is noted that the global minimum estimates obtained by the BFGS, SANN and Nelder-Mead methods through the `optim` function (details about the `optim` function execute `?optim`) are heavily influenced by kickoff given zero. Note that there is $\varepsilon > 0$ such that f has derivative close to zero around $(-\varepsilon, \varepsilon)$. However, the global minimum estimate obtained by the `pso` function of the `AdequacyModel` package provided the true global minimum, i.e., the minimum is equal to that one obtained analytically.

Notes: [1] In all cases, the algorithms converged according to the criteria of convergences implemented by the `optim` function of the `stats` package nstalled by default in the installation of the `R` language and the `pso` function is obtained by installing the `AdequacyModel` package.

[2] For the BFGS, Nelder-Mead and simulated annealing methods were given the same initial kick zero. For the case of the simulated annealing and PSO methods, these methodologies involve randomization, and it is fixed a seed at the value 9, i.e., `set.seed(9)`.

[3] The global minimum values obtained by the BFGS, Nelder-Mead and simualted annealing methods are identical and influenced by the kickoff. Unlike these methodologies, the PSO method implemented by the `pso` function does not require initial kicks.

[4] These results can be replicated using the `AdequacyModel` package and the code below:

```
R> f <- function(x){
+   -(6 + x^2 * sin(14*x))
+ }

R> f_pso <- function(x,par){
+   theta = par[1]
+   -(6 + theta^2 * sin(14*theta))
+ }

R> set.seed(9)
R> result_pso_f = pso(func=f_pso,S=500,lim_inf=c(-2.5),lim_sup=c(2.5),
+                     e = 0.0001)
R> set.seed(9)
R> result_sann_f = optim(par=c(0),fn=f, lower=-2.5, upper = 2.5, method="SANN")
R> result_bfgs_f = optim(par=c(0),fn=f, lower=-2.5, upper = 2.5, method="BFGS")
R> result_nelder_f = optim(par=c(0),fn=f, lower=-2.5, upper = 2.5,
+                          method="Nelder-Mead")
```

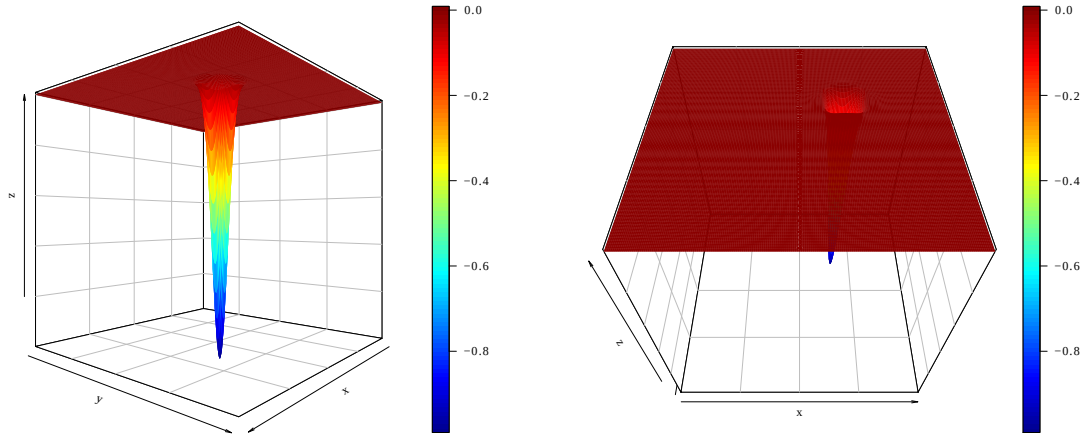
Note that the use of the `pso` function is rather simplistic. This function is implemented to be parsimonious in order to facilitate its use. The following example use `pso` function for multi-parameter optimizations.

Example

Consider the Easom function $f(x, y) = -\cos(x)\cos(y)\exp\{-[(x-\pi)^2 + (y-\pi)^2]\}$, and $-10 \leq x, y \leq 10$. Some plots are displayed at different angles in Figures 5.2(a) and 5.2(b). The Easom function is minimized at $x = y = \pi$, and $f(\pi, \pi) = -1$. The use of the `pso` function to minimize the above function is

```
R> easom <- function(x,par){
+   x1 = par[1]
+   x2 = par[2]
+   -cos(x1) * cos(x2) * exp(-((x1-pi)^2 + (x2-pi)^2))
+ }
R> set.seed(9)
R> results_pso = pso(func = easom, S = 500, lim_inf = c(-10,-10),
+                    lim_sup = c(10,10), e = 0.0001)
```

Before the execution of the `pso` function, we sent a seed, i.e., `set.seed(9)`, for which the same results can be replicated. The estimated minimum points by the `pso` function are $\hat{x} = 3.139752$ and $\hat{y} = 3.141564$, very close to $x = y = \pi$. The convergence of the methodology for very close values to the global optimum can be best observed in Easom function levels curves displayed in Figure 5.3.



(a) (b)

Figure 5.2: Easom function at two different angles.

We use the BFGS method through the `optim` function. We have, for example, for the initial kick $x = -9$ and $y = 9$, and there is convergence in the BFGS algorithm, where the kickoff is the minimum point itself ($\hat{x} = -9$ and $\hat{y} = 9$), which is quite different from the minimum true point $x = y = \pi$. This fact can be obtained from the code below.

```
R> easom1 <- function(x){
+   x1 = x[1]
+   x2 = x[2]
+   -cos(x1) * cos(x2) * exp(-((x1-pi)^2 + (x2-pi)^2))
+ }

R> result_bfgs_easom = optim(par = c(9,9), fn = easom1, method = "BFGS")
```

Notes: [1] Note that `result_bfgs_easom$convergence == 0` is equal to `TRUE`, which indicates convergence. For more details about the convergence criterion of the BFGS method implemented in the `optim` function execute `help(optim)`. So, this method is very sensitive to initial kicks. In addition, in the case of Easom function, convergence is hampered by the existence of infinite candidates to the point of minimum distributed on a flat region. The output stored in the object `result_bfgs_easom` is presented below:

```
R> result_bfgs_easom
$par
[1] -9  9

$value
[1] -1.283436e-30
```

```
$counts
function gradient
1        1

$convergence
[1] 0

$message
NULL
```

[2] Setting `result_nelder_easom=optim(par=c(-9,9),fn=easom1,method="Nelder-Mead")`, we also obtain a point distant from the minimum estimate of the true global minimum point, where $\hat{x} = -8.1$ and $\hat{y} = 9$ gives a minimum value approximately equal to zero. The results stored in `result_nelder_easom` are given below:

```
R> result_nelder_easom
$par
[1] -8.1  9.0

$value
[1] -3.609875e-71

$counts
function gradient
3       NA

$convergence
[1] 0

$message
NULL
```

[3] A similar fact is obtained using the simulated method in which the estimates can be obtained with the code that follows:

```
R> set.seed(9)
R> result_sann_easom = optim(par = c(-9,9), fn = easom1, method = "SANN")
```

In this case, it is noted that `result_sann_easom$convergence == 0` is `TRUE` (there is convergence). However, the estimated minimum point has coordinates distant from the coordinates of the true minimum point, where the estimated coordinates are $\hat{x} = 1.110688$ and $\hat{y} = 13.934928$ with seed fixed at 9, i.e. `set.seed(9)`.

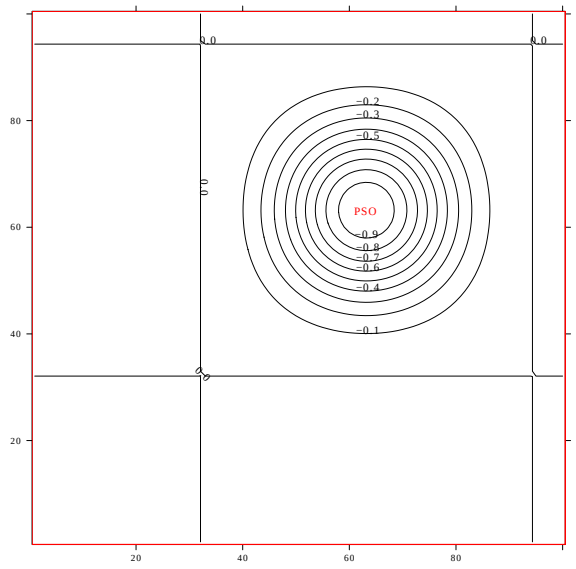


Figure 5.3: Curves of levels of the Easom function.

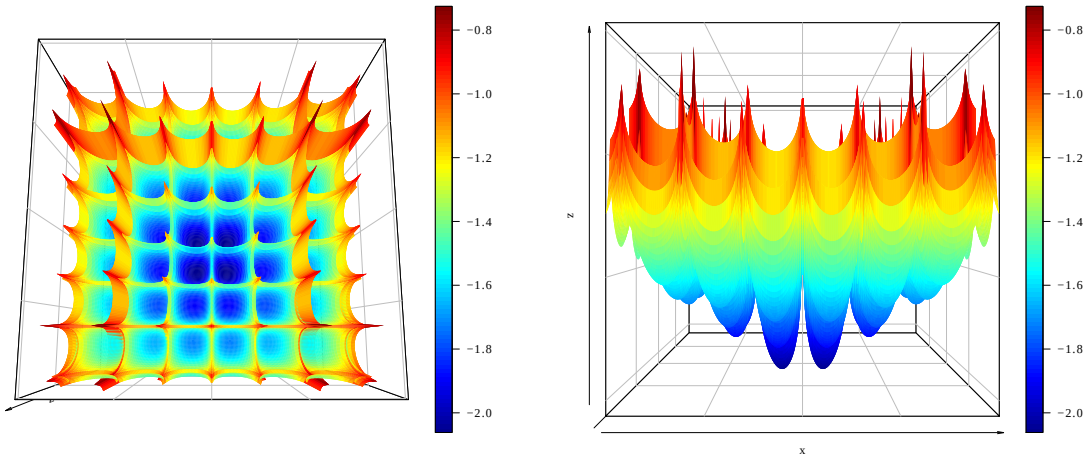
Example: Now, we consider the example of the use of the `pso` function to minimize the Cross-in-tray function. This is a difficult function to be minimized for different reasons from those presented in the previous example. The Cross-in-tray function has many local minima as can be seen in Figures 5.4(a) and 5.4(b). This fact can certainly hamper the convergence of various algorithms that lead to a global optimum. The Cross-in-tray function is defined by

$$f(x,y) = -0.0001\left(\left|\sin(x)\sin(y)\exp\left(\left|100 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right| + 1\right)^{0.1},$$

where $-10 \leq x,\, y \leq 10$ and

$$\text{Min} = \begin{cases} f(1.34941,\, -1.34941) &= -2.06261 \\ f(1.34941,\, 1.34941) &= -2.06261 \\ f(-1.34941,\, 1.34941) &= -2.06261 \\ f(-1.34941,\, -1.34941) &= -2.06261 \end{cases}.$$

Note that this function has four points of global minimum. Any estimate of the minimum points $(\hat{x},\, \hat{y})$ that applied in $f(\cdot)$ presents minimum value close to -2.0626 will be a good solution.



(a) (b)

Figure 5.4: Cross-in-tray function at two different angles.

By means of the `optim` function, we note the convergence of the BFGS, Nelder-Mead and simulated annealing methods with initial kicks at $x = 0$ and $y = 0$, where the estimated values of x and y are $\hat{x} = \hat{y} = 0$ in the three methodologies and $f(\hat{x}, \hat{y}) = -0.0001$.

The minimization of the Cross-in-tray function by using the PSO function achieves a satisfactory outcome as we can see in Figure 5.5. By the `pso` function, the estimated minimum point is $(1.3490, 1.3490)$ with a minimum value equal to $f(1.3490, 1.3490) = -2.0626$. These same results can be obtained with the script below:

```
R> cross <- function(x,par){
+   x1 = par[1]
+   x2 = par[2]
+   -0.0001 * (abs(sin(x1) * sin(x2) * exp(abs(100-sqrt(x1^2+x2^2)/pi)))
+   + 1)^0.1
+ }

R> set.seed(9)
R> result_pso_cross <- pso(func = cross, S = 500, lim_inf = c(-10,-10),
+                          lim_sup = c(10,10), e = 0.0001)
```

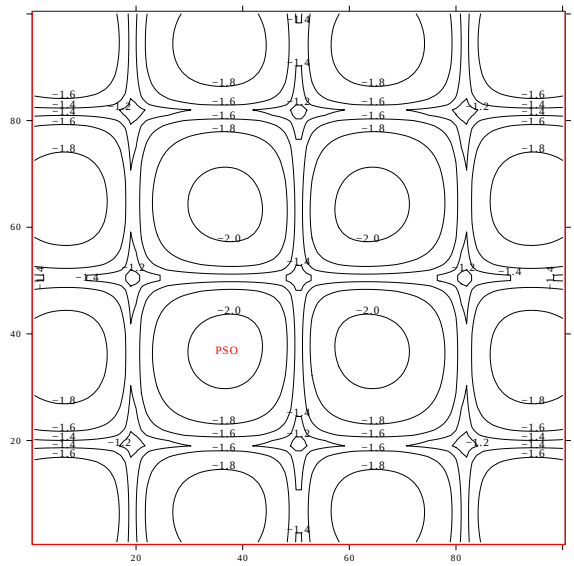


Figure 5.5: Curves of levels of Cross-in-tray function.

Note: The results of the optimization using the `optim` function and the Nelder-Mead, BFGS and simulated annealing methods can be obtained with the code below such that, for all these methodologies, the initial shot is given at the point $(0, 0)$.

```
R> cross1 <- function(x){
+   x1 = x[1]
+   x2 = x[2]
+   -0.0001 * (abs(sin(x1) * sin(x2) * exp(abs(100-sqrt(x1^2+x2^2)/pi)))
+   + 1)^0.1
+ }

R> result_bfgs_cross = optim(par = c(0,0), fn = cross1, lower = -10,
+                            upper = 10, method = "BFGS")
R> result_nelder_cross = optim(par = c(0,0), fn = cross1, lower = -10,
+                              upper = 10, method = "Nelder-Mead")
R> set.seed(9)
R> result_sann_cross = optim(par = c(0,0), fn = cross1, lower = -10,
                            upper = 10, method = "SANN")
```

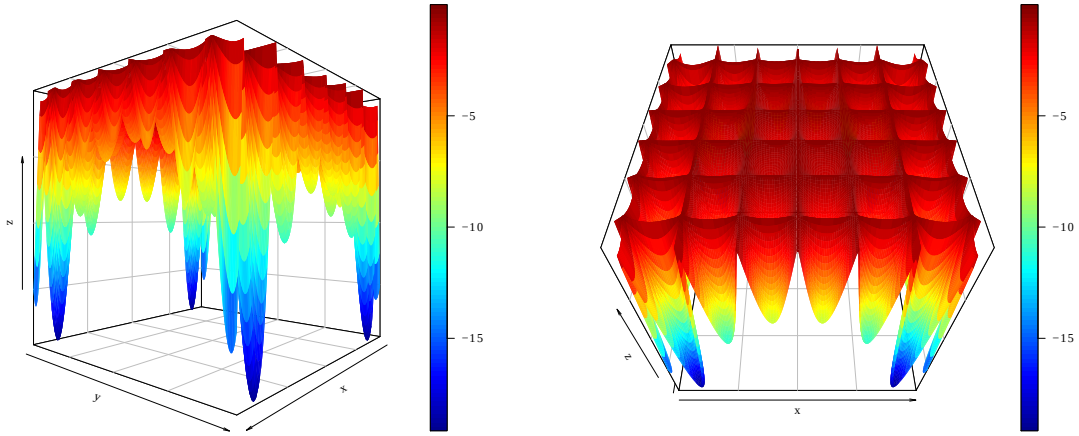
Example: Consider the case of the Hölder function, very peculiar and difficult to be optimized. It is defined by

$$f(x,y) = -\left|\sin(x)\cos(y)\exp\left(\left|1 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right|,$$

where

$$\text{Min} = \begin{cases} f(8.05502,\ 9.66459) & = & -19.2085 \\ f(-8.05502,\ 9.66459) & = & -19.2085 \\ f(8.05502,\ -9.66459) & = & -19.2085 \\ f(-8.05502,\ -9.66459) & = & -19.2085 \end{cases},$$

where $-10 \leq x,\ y \leq 10$. Figure 5.6 displays the plots of the Hölder function defined above.



(a) (b)

Figure 5.6: Hölder function at two different angles.

For the Hölder function, the results obtained from Nelder-Mead, BFGS and simulated annealing methods, thus as occurred in the previous examples, were not good. However, in all cases, there was a convergence following these methodologies implemented in the `optim` function. With initial kicks at the point $(0, 0)$, the convergence leads to this point, i.e., the three methodologies estimate the minimum point at $\hat{x} = 0$ and $\hat{y} = 0$.

For the simulated annealing method, the seed was set at 9, i.e., `set.seed(9)`. The details for the `optim` function can be obtained in the documentation of this function.

An interesting fact is that the `pso` function also failed to get good estimates for `S = 500`, i.e., when considering 500 particles for optimization. However, the problem is easily circumvented by raising the number of particles. Figure 5.7 displays plots of the levels of the Hölder function with the point of convergence of the PSO algorithm. This result was obtained by using the following script:

```
R> holder <- function(x,par){
+    x1 = par[1]
+    x2 = par[2]
+    -abs(sin(x1)*cos(x2) * exp(abs(1 - sqrt(x1^2+x2^2)/pi)))
+ }

R> result_pso_holder = pso(func = holder, S = 10000, lim_inf = c(-10,-10),
+                          lim_sup = c(10,10), e = 0.0001)
```

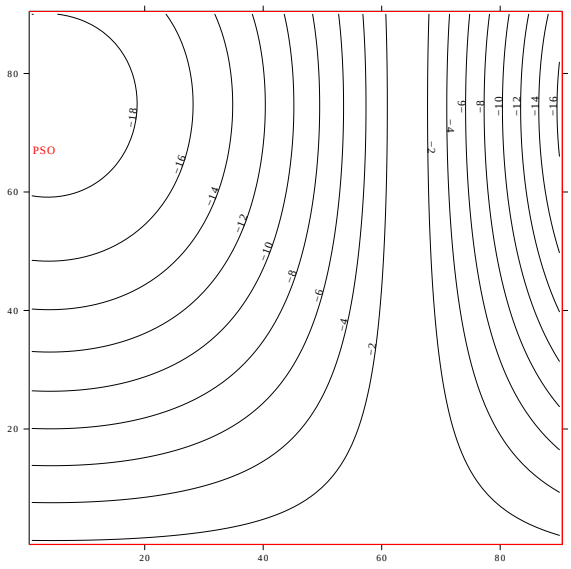


Figure 5.7: Curves of levels of Hölder function.

5.3 Fitting a distribution

In parametric inference, the problem of inferring about the unknown joint distribution, say F_θ, from a sample $X_1, \ldots, X_n$ reduces to the problem of inferring about the unknown parameter θ. So, suppose we have a parametric family of distributions $\mathcal{F} = \{F_\theta; \theta \in \Theta\}$, where Θ is the parameter space of θ. Suppose also that in $\mathcal{F}$ exists a F_θ such that F evaluated in $\hat{\theta}_n$ provides the best element to model $X_1, \ldots, X_n$, i.e., $\hat{\theta}_n$ is an estimator of θ.

In Statistics, we usually estimate θ by the maximum likelihood estimate (MLE) $\hat{\theta}_n$. However, not always the assumption that $\mathcal{F}$ is adequate, i.e., it can be that F evaluated at $\hat{\theta}_n$ does not provide a distribution that can model properly $X_1, \ldots, X_n$. Thus, we need to check the adequacy of the adjustment of the best element in $\mathcal{F}$ and decide if $F_{\hat{\theta}_n}$ is, or not, a good distribution for $X_1, \ldots, X_n$.

In this way, we need to verify if the best element of $\mathcal{F}$ can represent the joint distribution of $X_1, \ldots, X_n$, i.e., we need to process the best value $\theta \in \Theta$. One way to obtain an adequate distribution is using the likelihood ratio statistic. However, many times, this statistics is complicated and can even be inadequate as a statistical test, mainly in small samples, where the asymptotic χ^2 distribution can be poor.

Alternatives to the likelihood ratio test were proposed by Chen and Balakrishnan (1995) that are corrections to the Carmér-von Mises and Anderson Darling statistics proposed by von-Mises (1931), Cramér (1928) and Anderson and Darling (1952).

We use these statistics when we have a random sample $x_n = \{x_1, \ldots, x_n\}$ with empirical distribution function $F_n(x)$ and we want to test if the sample has a specified distribution. The Cramér-von Mises (A^*) and Anderson-Darling (W^*) are, respectively, given by

$$W^* = \left\{ n \int_{-\infty}^{+\infty} \{F_n(x) - F(x;\widehat{\theta}_n)\}^2 dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.5}{n}\right) = W^2 \left(1 + \frac{0.5}{n}\right), \tag{5.3}$$

$$\begin{aligned} A^* &= \left\{ n \int_{-\infty}^{+\infty} \frac{\{F_n(x) - F(x;\widehat{\theta}_n)\}^2}{\{F(x;\widehat{\theta})(1 - F(x;\widehat{\theta}_n))\}} dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right) \\ &= A^2 \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right), \end{aligned} \tag{5.4}$$

where $F_n(x)$ is the empirical distribution function, $F(x;\widehat{\theta}_n)$ is the postulated cdf evaluated at the MLE $\widehat{\theta}_n$ of θ, and W^2 and A^2 are the Cramér-von Mises and Anderson-Darling statistics, respectively. For more details about the W^2 and A^2 statistics, see Craér (1928), von-Mises (1931) and Anderson and Darling (1952). This statistic is given by the difference between $F_n(x)$ and $F(x;\widehat{\theta}_n)$. Thus, as lower are them more evidence we have that $F(x;\widehat{\theta}_n)$ generate the sample. The null hypothesis tested using the statistics (5.3) and (5.4) is that the random sample $x_1, \ldots, x_n$ has distribution $F(x;\theta)$. According to Chen and Balakrishnan (1995, p. 155), the W^2 and A^2 statistics can be readily calculated as

$$W^2 = \sum_{i=1}^{n} [u_i - \{(2i-1)/(2n)\}]^2 + 1/(12n) \tag{5.5}$$

and

$$A^2 = -n - n^{-1} \sum_{i=1}^{n} \{(2i-1)\log(u_i) + (2n+1-2i)\log(1-u_i)\}, \tag{5.6}$$

where $u_i = \Phi((y_i - \overline{y})/s_y)$, $v_i = F(x_i;\widehat{\theta}_n)$, $y_i = \Phi^{-1}(v_i)$ (Φ is the standard normal cdf) and s_y is the sample standard deviation of the y_i's, for $i = 1, 2, \ldots, n$. The algorithm below can be adpoted to obtain W^* and A^*:

1. Estimate θ by $\widehat{\theta}_n$ (consistently), order the sample values in crescent values to calculate $v_i = F(x_i;\widehat{\theta}_n)$;
2. Calculate $y_i = \Phi^{-1}(v_i)$, where Φ^{-1} is the standard normal quantile function;
3. Calculate $u_i = \Phi\{(y_i - \overline{y})/s_y\}$, where $\overline{y} = n^{-1}\sum_{i=1}^{n} y_i$ and $s_y^2 = (n-1)^{-1}\sum_{i=1}^{n}(y_i - \overline{y})^2$;
4. Calculate W^2 e A^2 using equations (5.5) and (5.6), respectively;
5. Obtain $W^* = W^2(1 + 0.5/n)$ and $A^* = A^2(1 + 0.75/n + 2.25/n^2)$, where n is the sample size;
6. We reject $\mathcal{H}_0$ at the significance level α if the test statistics exceed the critical values presented by Chen and Balakrishnan (1995, p. 155).

What is commonly done in practice is to use W^* and A^* to compare two or more continuous distributions. The distribution that gives the lowest values of W^* or A^* is the best suited to explain the random sample.

The `goodness.fit` function provides some useful statistics to assess the quality of fit of probabilistic models, including the W^* and A^* statistics. The function can also determine other goodness-of-fit

statistics such as the AIC (Akaike Information Criterion), CAIC (Consistent Akaikes Information Criterion), BIC (Bayesian Information Criterion), HQIC (Hannan-Quinn Information Criterion) and KST (Kolmogorov-Smirnov Test). The general form for the function is given below with the descriptions of each one of its arguments.

```
goodness.fit(pdf, cdf, starts=NULL, data, method="PSO", domain=c(0,Inf), mle =
             NULL)},
```

where

- `pdf`: probability density function;
- `cdf`: cumulative distribution function;
- `starts`: initial parameters to maximize the likelihood function;
- `data`: data vector;
- `method`: method used for minimization of the -log-likelihood function. The methods supported are: PSO (default), BFGS, Nelder-Mead, SANN (simulated annealing), CG (conjugate gradients). We can also provide only the first letter of the methodology, i.e., P, B, N, S or C, respectively;
- `domain`: domain of the pdf. By default the domain of the pdf is the open interval $(0, \infty)$. This option must be a vector with two components;
- `mle`: vector with the MLEs. This option should be used if one already has knowledge of the MLEs. The default is NULL, i.e., user the function will try to obtain the MLEs;
- `...`: If `method = PSO`, then all the arguments of pso function could be passed to the `goodness.fit` function.

An important observation is that it is not necessary to define the likelihood function or log-likelihood. Just we need to define the pdf and cdf. The function will self-criticism to the arguments passed to the `goodness.fit`. For example, if supplied to the arguments `pdf` or `cdf` functions that do not be pdfs and cdfs, a notice will be given so that the user can check the arguments passed. We provide below two examples of the use of the `goodness.fit` function.

Example: Suppose the problem is that it has a data set of stress (until fracture) of carbon fibres (in Gba). The data were obtained by Nichols and Padgett (2006) are available for use in the AdequacyModel package and can be accessed with the command `data(carbone)`. Further, details regarding the set of data is obtained in the documentation of the package with the command `help(carbone)`. Suppose also that we are interested in obtaining the best model in $\mathcal{F} = \{F_\theta;\, \theta \in \Theta\}$ that can represent the distribution of $X_1, \ldots, X_n$, whose observations are in `carbone`. Here, we consider that $\mathcal{F}$ is the Exp-Weibull (exponentiated Weibull) distribution. Its cdf is given by

$$F(x, \alpha, \beta, a) = \left\{1 - \exp\left[-(\alpha x)^\beta\right]\right\}^a,$$

where α, β and c are positive parameters and $x > 0$. Thus, each element in $\mathcal{F}$ is of the form $F(x; \alpha, \beta, a)$.

We initially implement the pdf $f(x;\alpha,\beta,a)$ and the cdf $F(x;\alpha,\beta,a)$. They will serve as arguments for `pdf` and `cdf`, respectively. We present below the implementation of the functions that will be given to the `goodness.fit` function.

```
R> # Probability density function.

R> pdf_expweibull <- function(par, x) {
+   alpha = par[1]
+   beta = par[2]
+   a = par[3]
+   alpha * beta * a * exp(-(alpha * x) ^ beta) * (alpha * x) ^ (beta
+   - 1) * (1 - exp(-(alpha * x) ^ beta)) ^ (a - 1)
+ }

R> # Cumulative distribution function.

R> cdf_expweibull <- function(par, x) {
+   alpha = par[1]
+   beta = par[2]
+   a = par[3]
+   (1 - exp(-(alpha * x) ^ beta)) ^ a
+ }
```

```
R> data(carbone)
R> results = goodness.fit(pdf = pdf_expweibull, cdf = cdf_expweibull,
+  starts = c(1, 1, 1), data = carbone, method = "BFGS", domain = c(0, Inf),
+           mle = NULL)
```

The object `results` feature all the goodness-of-fit statistics cited previously as well as the MLEs in case of `mle = NULL` (default). The error of the MLEs if the argument `method` receives `PSO`, `BFGS`, `Nelder-Mead`, `SANN` and `CG`. Thus,

- `R> results$W` provides the statistic W^*;
- `R> results$A` provides the statistic A^*;
- `R> results$KS` provides the statistic of Kolmogorov-Smirnov;
- `R> results$mle` provides a vector with the MLEs of the model parameters given as arguments for the `pdf`;
- `R> results$AIC`: provides the AIC statistic;
- `R> results$CAIC`: provides the CAIC statistic;
- `R> results$BIC`: provides the BIC statistic;
- `R> results$HQIC`: provides the HQIC statistic;

- `R> result$KS`: returns an object of class `htest` with information on the Kolmogorov-Smirnov test;
- `R> results$Erro`: provides the standard erros the MLEs of the parameters, which index the model parameters given as arguments for the `pdf` and `cdf`;
- `R> results$value`: displays the minimum value of the function -log(likelihood);
- `R> result$Convergence`: provides information on the convergence of the method passed as an argument for `method`. If `result$Convergence} == 0` for `TRUE`, there was convergence.

In case of `method = "PSO"` (default), the errors will not be provided. The researcher may obtain such errors through bootstrap. For details of how to obtain the estimates of the standard errors of the MLEs of the model parameters, see Davison and Hinkley (1997). Just below follow the results stored in the object `results` (output of the `goodness.fit` function) and a plot with the fitted Exp-Weibull density.

```
R> results
$W
[1] 0.07047089

$A
[1] 0.4133608

$KS

One-sample Kolmogorov-Smirnov test

   data:  data
   D = 0.064568, p-value = 0.7987
   alternative hypothesis: two-sided


$mle
[1] 0.3731249 2.4058010 1.3198053

$AIC
[1] 288.6641

$‘CAIC ‘
[1] 288.9141

$BIC
[1] 296.4796

$HQIC
[1] 291.8272
```

```
$Erro
[1] 0.06265212 0.60467076 0.59835491

$Value
[1] 141.332

$Convergence
[1] 0
```

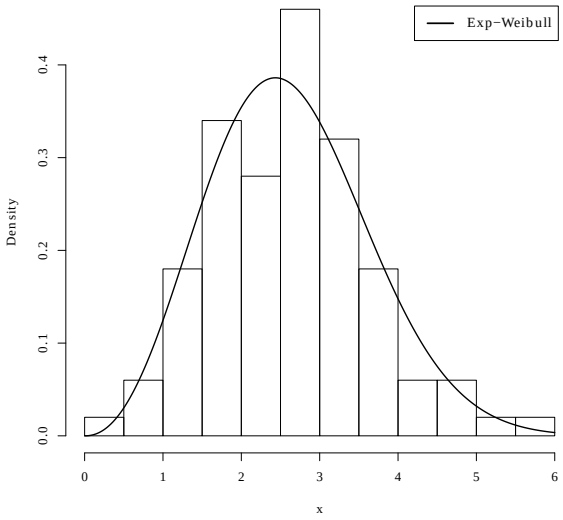


Figure 5.8: Fitted Exp-Weibull density to stress data (until fracture) of carbon fibers in Gba.

Notes: [1] The Kolmogorov-Smirnov statistic may return `NA` with a certain frequency. The return `NA` informs that this statistic is not reliable for the current data set. More details about this issue can be obtained with `help(ks.test)`. In situations where `results$Convergence==0` is `TRUE`, there was convergence for the method passed as an argument to the `method` that minimizes the log-likelihood function multiplied by -1, that is, it minimizes -log(likelihood).

[2] The convergence criterion as well as other details about possible values returned by `results$Convergence` can be obtained with `help(optim)` if the argument `method` of the `goodness.fit` function receives the strings `"BFGS"`, `"Nelder-Mead"`, `"SANN"` or `"CG"` (or such those initials letters `"B"`, `"N"`, `"S"` or `"C"`). For the PSO methodology of minimization of the -log(likelihood) function (default `method = "PSO"`), the convergence criterion is displayed as discussed in Section 5.2, which normally is satisfied.

[3] The code for obtaining Figure 5.8 is presented below:

```
R> pdf(file = "plot_adjustment.pdf", width = 9, height = 9, paper = "special",
+      family = "Bookman", pointsize = 14)
+     x = seq(0, 6, length.out = 250)
+     hist(carbone, probability = TRUE, xlab = "x", main = "")
+     lines(x, pdf_expweibull(par = results$mle, x), lwd = 2)
+     legend( "topright", legend = c(expression(paste("Exp-Weibull"))),
+     lwd = c(2.5), inset = 0.03, lty = c(1), cex = 1.1, col = c("black"))
+ dev.off()
```

TTT plot

Several aspects of an absolutely continuous distribution can be seen more clearly from the hazard rate function (hrf) than from either the distribution or density functions. The hrf is an important quantity characterizing life phenomena. Let X be a random variable with the pdf $f(x)$ and the cdf $F(x)$. The hrf of X is defined by

$$h(x) = \frac{f(x)}{1 - F(x)},$$

where $1 - F(x)$ is the survival function.

The hrf may be increase, decrease, constant, upside-down bathtub, bathtub-shaped or indicate a more complicated process. In many applications there is a qualitative information about the hazard rate shape, which can help in selecting a specified model. In this context, a device called total time on test (TTT) or its scaled TTT transform proposed by Aarset (1987) may be used for obtaining the empirical behavior of the hrf.

The scaled TTT transform if defined by $(0 < u < 1)$

$$\phi_X(u) = \frac{H_X^{-1}(u)}{H_X^{-1}(1)},$$

where $H_X^{-1}(u) = \int_0^{Q(u)} [1 - F(x)]dx$ and $Q(u)$ is the quantile function of X. The quantity $\phi_X(\cdot)$ can be empirically approximated by

$$T(i/n) = \frac{\sum_{k=1}^{i} X_{k:n} + (n-i)X_{i:n}}{\sum_{k=1}^{n} X_k},$$

where $i = 1, \ldots, n$ and $X_{k:n}$, $k = 1, \ldots, n$, are the order statistics of the sample. Thus, the TTT plot is obtained by plotting $T(i/n)$ against i/n. We can detect the type of hazard rate that the data have. It is a straight diagonal for constant failure rates, it is convex for decreasing failure rates and concave for increasing failure rates. It is first convex and then concave if the failure rate is bathtub-shaped. It is first concave and then convex if the failure rate is upside-down bathtub. For more details, see Aarset (1987).

The computation of the TTT plot is proposed in the `AdequacyModel` package. The data set named `carbone` will now be used to illustrate the TTT plot function of this package. The real data sets correspond to a data set from Nichols and Padgett (2006) on breaking stress of carbon fibres (in Gba). In order to obtain the TTT curve, the `TTT` function has been developed. The following instructions illustrate these functions:

```
R> library(AdequacyModel)
R> data(carbone)
R> TTT(carbone, col = "red", lwd = 2.5, grid = TRUE, lty = 2)
```

The TTT plot for the carbone data set Nichols and Padgett (2006) is displayed in Figure 5.9, which reveals increasing hrf. This plot indicates that distributions with increasing hrf seem to be appropriate for modeling the cabone data set, so that several distributions that have increasing hrf could be good candidates; see the theoretical plot in Figure 1 in Aarset (1987).

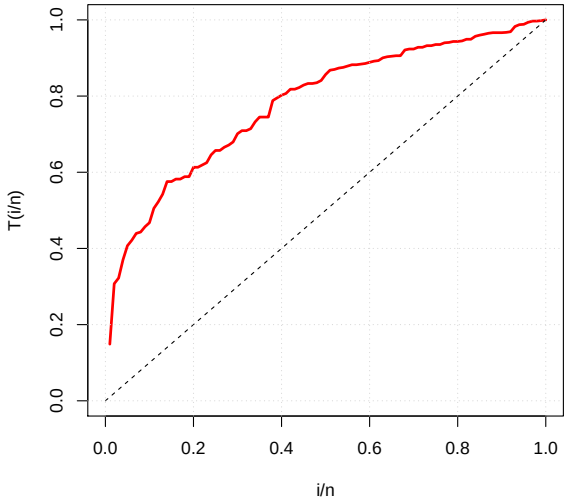


Figure 5.9: TTT-plot for carbon data.

5.4 Concluding remarks and the package current usage

It can be said that the `AdequacyModel` package provides a comprehensive and efficient method to perform a robust and general purpose optimization of an arbitrary objective function, mainly for situations where the objective function has approximately flat regions, which represents an important advantage over derivative-based optimization approaches. Further, it provides a conclusive toolbox for assessing the adequacy of probabilistic models for a given data set by combining several statistical measures. The quantities computed include (W^*, W^*, AIC, BIC, CAIC, among others). Due to the great interest in and active development of lifetime models, which in general need to maximize the likelihood function, it was deliberately constructed in an object-oriented and extensible fashion. Consequently, it is ready for the many extensions that are sure to come in near future. The source code is open and extensive documentation of the system is freely available. We emphasize that this package has been developed with the objective of helping `R` users to fit probabilistic models in a wide range of scientific studies. Indeed, this package is already used by a lot of practitioners and academics for model fits and goodness-of-fit statistics in biomedical and health sciences (Bourguignon et al., 2015; Cordeiro et al., 2015; Ramos et al., 2013); physics (Cordeiro et al., 2016); reliability (Ramos et al., 2013); financial and actuarial mathematics (Duarte et al., 2016). Figures 5.10 and 5.11 show,

respectively, the numbers of downloads of `AdequacyModel` package from 2013-12-20 to 2016-04-11 and the numbers of downloads of the package by countries in the RStudio repository.

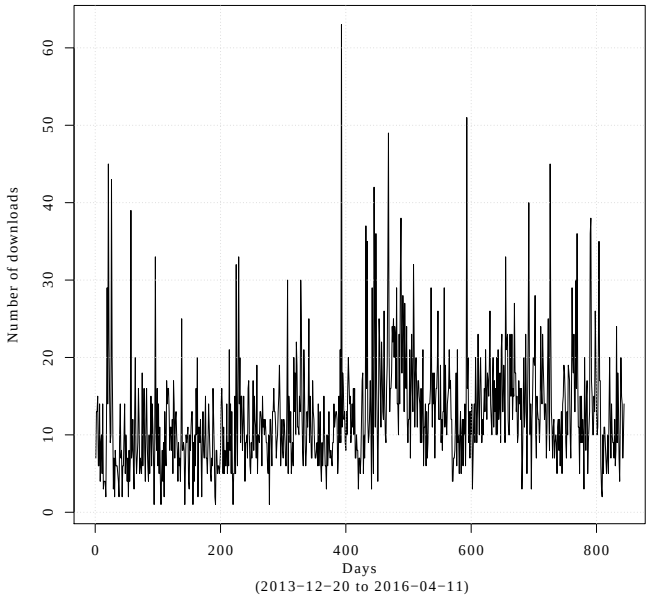


Figure 5.10: Numbers of downloads from `AdequacyModel` package by RStudio repository.

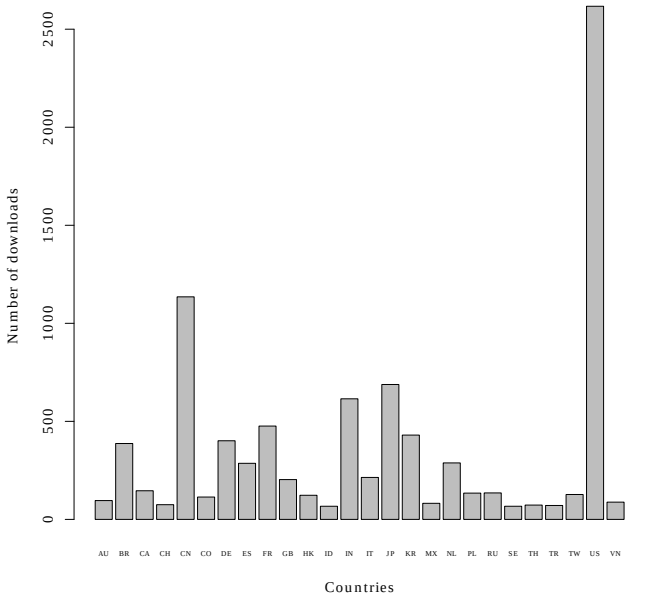


Figure 5.11: Numbers of downloads from `AdequacyModel` package by countries by RStudio repository.

References

[1] Aarset, M V. How to identify a bathtub hazard rate. *IEEE Transactions on Reliability*, v. 36, p. 106–108, 1987.

[2] Adamidis, K and Loukas, S. A lifetime distribution with decreasing failure rate. *Statistics & Probability Letters*, v. 39, p. 35–42, 1998.

[3] Anderson, T W and Darling, D A. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, p. 193–212, 1952.

[4] Broyden, CG, Fletcher, R and Goldfarb, D. DF Shanno J. Inst. *Math. Appl*, v. 6, p. 76, 1970.

[5] Bourguignon, M, Ghosh, I and Cordeiro, G M. General results for the transmuted family of distributions and new models. *Journal of Probability and Statistics*, v. 36, 2015.

[6] Chen, G and Balakrishnan, N. A general purpose approximate goodness-of-fit test. *Journal of Quality Technology*, v. 27, p. 154–161, 1995.

[7] Cintra, R J and Rêgo, L C and Cordeiro, G M and Nascimento, A D C. Beta generalized normal distribution with an application for SAR image processing. *Statistics*, v. 48, p. 279–294, 2014.

[8] Cordeiro, G M and de Castro, M. A New Family of Generalized Distributions. *Journal of Statistical Computation and Simulation*, v. 81, p. 883–898, 2011.

[9] Cordeiro, G M, Ortega, E M M and Ramires, Thiago G. A new generalized Weibull family of distributions: mathematical properties and applications. *Journal of Statistical Distributions and Applications*, v. 2, p. 1–25, 2015.

[10] Cramér, H. On the composition of elementary errors. *Mathematical deductions. Scandinavian Actuarial Journal*, v. 1928, p. 13–74, 1928.

[11] Davison, A C and Hinkley, D V. Bootstrap methods and their application. *Cambridge University Press*, Ed. 1, 1997.

[12] Duarte, J J, Freitas, W W L and Cordeiro, G M. The Extended Generalized Half-Normal Distribution. *Brazilian Journal of Probability and Statistics*, Accepted Manuscript, 2016.

[13] Eberhart, R C and Kennedy, J. A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human science*, v. 1, p. 39–43, 1995.

[14] Eugene, N, Lee, C and Famoye, F. Beta-normal Distribution and its Applications. *Communications in Statistics - Theory and Methods*, v. 31, p. 497–512, 2002.

[15] Fletcher, S G and Ponnambalam, K. A New Formulation for the Stochastic Control of Systems with Bounded State Variables: An Application to a Single Reservoir System. *Stochastic Hydrology and Hydraulics*, v. 10, p. 167–186, 1996.

[16] Gupta, R D and Kundu, D. Generalized Exponential Distributions. *Generalized Exponential Distributions*, v. 41, p. 173–188, 1999.

[17] Jones, M C. Kumaraswamy's Distribution: A Beta-type Distribution with some Tractability Advantages. *Statistical Methodology*, v. 6, p. 70–81, 2009.

[18] Kennedy, J, Kennedy, J F and Eberhart, R C. Swarm intelligence, Ed. Morgan Kaufmann, 2001.

[19] Kumaraswamy, P. A Generalized Probability Density Function for Double-bounded Random Processes. *Journal of Hydrology*, v. 46, p. 79–88, 1980.

[20] Lai, C D. Constructions and applications of lifetime distributions. *Applied Stochastic Models in Business and Industry*, v. 29, p. 127–140, 2013.

[21] Marshall, A W and Olkin, I. A new Method for Adding a Parameter to a Family of Distributions with Application to the Exponential and Weibull Families. *Biometrika*, v. 84, p. 641–652, 1997.

[22] Nelder, J A and Mead, Roger. A simplex method for function minimization. *The Computer Journal*, v. 7, p. 308–313, 1965.

[23] Nichols, M D and Padgett, W J. A bootstrap control chart for Weibull percentiles. *Quality and reliability engineering international*, v. 22, p. 141–151, 2006.

[24] Parsopoulos, K E and Vrahatis, M N. Recent approaches to global optimization problems through particle swarm optimization. *Natural computing*, v. 1, p. 235–306, 2002.

[25] Ramos, M W A, Cordeiro, G M, Marinho, P R D, Dias, C R B and Hamedani, G G. The Zografos-Balakrishnan Log-Logistic Distribution: Properties and Applications. *Journal of Statistical Theory and Applications*, v. 12, p. 225–244, 2013.

[26] Razzaghi, M. Beta-normal distribution in dose-response modeling and risk assessment for quantitative responses. *Environmental and Ecological Statistics*, v. 16, p. 25–26, 2009.

[27] Rêgo, L C and Cintra, R J and Cordeiro, G M. On Some Properties of the Beta Normal Distribution. *Communications in Statistics - Theory and Methods*, v. 41, p. 3722–3738, 2012.

[28] Seifi, A, Ponnambalam, K and Vlach, J. Maximization of Manufacturing Yield of Systems with Arbitrary Distributions of Component Values. *Annals of Operations Research*, v. 99, p. 373–383, 2000.

[29] . Shi, Y and Eberhart, R. A modified particle swarm optimizer. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, p. 69–73, 1998.

[30] von-Mises, R. Wahrscheinlichkeitsrechnung und ihre Anwendung in der Statistik und theoretischen Physik. Ed. *Deuticke*, 1931.