



Faculty of Sciences
Department of Mathematics & Statistical Sciences

Some Generalizations of the Type II Half Logistic Topp-Leone Distributions

by

Katlego Keganne

Reg. No: 14000820

MSc (Statistics), BIUST, Botswana

A Dissertation/Thesis Submitted to the Faculty of Sciences in Partial Fulfillment of the Requirements for the Award of the Degree of Master of Science in Statistics of BIUST

Supervisor(s):

Prof. Boikanyo Makubate
Department of Mathematics & Statistical Sciences,
Faculty of Sciences, BIUST
E-mail: makubateb@biust.ac.bw

Dr. Lesego Gabaitiri
Department of Mathematics & Statistical Sciences,
Faculty of Sciences, BIUST
E-mail: gabaitiril@biust.ac.bw

January, 2023

Contents

Contents	iii
List of Figures	iii
List of Tables	iv
1 Literature Review	1
1.1 Introduction	1
1.2 Some Useful Classical Distributions	2
1.2.1 Power Series Distribution	3
1.2.2 Uniform Distribution	3
1.2.3 Exponential Distribution	3
1.2.4 Weibull Distribution	4
1.2.5 Log Logistic Distribution	4
1.3 Some Important Generators	4
1.3.1 Beta-G Distribution and Exponentiated Generalized-G Distribution	4
1.3.2 Compounding Method	5
1.4 Some Characteristics of Distributions	5
1.4.1 Survival, Hazard, Reverse Hazard, and Quantile Functions	5
1.4.2 Skewness and Kurtosis	6
1.4.3 Moments and Moment Generating Functions	6
1.4.4 Inequality Curves	7
1.4.5 Mean Deviations	7
1.4.6 Entropy and Order Statistics	8
1.5 Maximum Likelihood Estimation	9
1.6 Goodness of Fit Statistics	9
1.6.1 Akaike Information Criterion (AIC)	9
1.6.2 Corrected Akaike Information Criterion (CAIC)	10
1.6.3 Bayesian Information Criterion (BIC)	10
1.6.4 Kolmogorov-Smirnov Test (K-S)	10
1.6.5 Anderson-Darling (A^*) and Cramér-von-Mises (W^*)	11
1.7 Aim and Objectives of the Study	11
1.8 Outline of the Thesis	11
2 Type II Half Logistic Topp-Leone-G Power Series Distribution	12
2.1 Introduction	12
2.2 Type II Half Logistic Topp-Leone-G Power Series Distribution	12
2.2.1 Quantile Function	14
2.3 Some Statistical Properties of the TIIHLTL-GPS Class of distributions	19

2.3.1	Moments, Conditional Moments, and Generating Function	19
2.3.2	Mean Deviations	20
2.3.3	Lorenz and Bonferroni Curves	20
2.4	Estimation	21
2.5	Some Special Cases of the TIIHLTL-GPS Family of Distributions	22
2.5.1	Type II Half Logistic Topp-Leone Weibull Poisson (TIIHLTL-WP) Distribution	22
2.5.2	Type II Half Logistic Topp-Leone Weibull Geometric (TIIHLTL-WG) Distribution	24
2.5.3	Type II Half Logistic Topp-Leone Weibull Logarithmic (TIIHLTL-WL) Distribution	26
2.5.4	Type II Half Logistic Topp-Leone Log-logistic Poisson (TIIHLTL-LLoP) Distribution	28
2.5.5	Type II Half Logistic Topp-Leone Log-logistic Geometric (TIIHLTL-LLoGG) Distribution	30
2.5.6	Type II Half Logistic Topp-Leone Log-logistic Logarithmic (TIIHLTL-LLoGL) Distribution	32
2.6	Simulation Study	34
2.7	Applications	36
2.7.1	Fatigue Life of Bearings Data	37
2.7.2	Window Strength Data	38
2.8	Concluding Remarks	39
3	Beta Type II Half Logistic Topp-Leone-G Distribution	40
3.1	Introduction	40
3.2	Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) Distribution	40
3.2.1	Quantile Function	41
3.3	Some Statistical Properties of the BTIIHLTL-G Family of Distributions	45
3.3.1	Moments, Conditional Moments, and Generating Functions	45
3.3.2	Mean Deviations	46
3.3.3	Lorenz and Bonferroni Curves	47
3.4	Estimation	47
3.5	Some Special Cases of the BTIIHLTL-G Distribution	48
3.5.1	Beta Type II Half Logistic Topp-Leone Uniform (BTIIHLTL-U) Distribution	49
3.5.2	Beta Type II Half Log Logistic Topp-Leone Exponential (BTIIHLTL-E) Distribution	50
3.5.3	Beta Type II Half Logistic Topp-Leone Log Logistic (BTIIHLTL-LLoG) Distribution	52
3.6	Simulation Study	55
3.7	Applications	55
3.7.1	Yarn at 2.3% Strain Level Data Set	57
3.7.2	Monthly Actual Taxes Revenue Data Set	57
3.8	Concluding Remarks	59

4	Type II Half Logistic Topp-Leone Exponentiated Generalized-G Distribution	60
4.1	Introduction	60
4.2	Type II Half Logistic Topp-Leone Exponentiated Generalized-G (TIIHLTLEG-G) Family of Distributions	60
4.2.1	Quantile Function	61
4.3	Some Statistical Properties of the TIIHLTLEG-G Family of Distributions	65
4.3.1	Moments, Conditional Moments, and Generating Functions	65
4.3.2	Mean Deviations	65
4.3.3	Lorenz and Bonferroni Curves	66
4.4	Estimation	66
4.5	Some Special Cases of TIIHLTLEG-G Distribution	68
4.5.1	Type II Half Logistic Topp-Leone Exponentiated Generalized Exponential (TIIHLTLEG-E) Distribution	68
4.5.2	Type II Half Log Logistic Topp-Leone Exponentiated Generalized Log Logistic (TIIHLTLEG-LLoG) Distribution	70
4.5.3	Type II Half Logistic Topp-Leone Exponentiated Generalized Weibull (TIIHLTLEG-W) Distribution	73
4.6	Simulation Study	75
4.7	Applications	75
4.7.1	Filure Times of Air Conditioning System Data Set	77
4.7.2	Survival Times of Guinea Pigs Data Set	79
4.8	Concluding Remarks	81
5	Conclusion and Discussion	82
5.1	Discussion and Conclusion	82

List of Figures

2.1	Pdf and hrf plots for the TIIHLTL-WP distribution	22
2.2	Skewness and Kurtosis plots for the TIIHLTL-WP distribution	24
2.3	Pdf and hrf plots for the TIIHLTL-WG distribution	25
2.4	Skewness and Kurtosis plots for the TIIHLTL-WG distribution	26
2.5	Pdf and hrf plots for the TIIHLTL-WL distribution	27
2.6	Skewness and Kurtosis plots for the TIIHLTL-WL distribution	28
2.7	Pdf and hrf plots for the TIIHLTL-LLoP distribution	29
2.8	Skewness and Kurtosis plots for the TIIHLTL-LLoGP distribution	30
2.9	Pdf and hrf plots for the TIIHLTL-LLoGG distribution	31
2.10	Skewness and Kurtosis plots for the TIIHLTL-LLoGG distribution	32
2.11	Pdfs and hrfs plots for the TIIHLTL-LLoGL distribution	33
2.12	Skewness and Kurtosis plots for the TIIHLTL-LLoGL distribution	34
2.13	Fitted Densities and Probability Plots for the Fatigue Life of Bearings Data . . .	37
2.14	Fitted Densities and Probability Plots for the Window Strength Data	38
3.1	Pdf and hrf plots for the BTIIHLTL-U distribution	49
3.2	Skewness and Kurtosis plots for the BTIIHLTL-U distribution	51
3.3	Pdf and hrf plots for the BTIIHLTL-E distribution	51
3.4	Skewness and Kurtosis plots for the BTIIHLTL-E distribution	53
3.5	Pdf and hrf plots for the BTIIHLTL-LLoG distribution	53
3.6	Skewness and Kurtosis plots for the BTIIHLTL-LLoG distribution	55
3.7	Fitted Densities and Probability Plots for the Yarn Data	58
3.8	Fitted Densities and Probability Plots for the monthly actual taxes revenue Data	59
4.1	Pdf and hrf plots for the TIIHLTLEG-E distribution	69
4.2	Skewness and Kurtosis plots for the TIIHLTLEG-E distribution	70
4.3	Pdf and hrf plots for the TIIHLTLEG-LLoG distribution	71
4.4	Skewness and Kurtosis plots for the TIIHLTLEG-LLoG distribution	72
4.5	Pdf and hrf plots for the TIIHLTLEG-W distribution	73
4.6	Skewness and Kurtosis plots for the TIIHLTLEG-W distribution	75
4.7	Fitted Densities and Probability Plots for the Failure Times of Air Conditioning System of an Airplane Data	78
4.8	Fitted Densities and Probability Plots for the Survival Times of Guinea Pigs Data	80

List of Tables

1.1	Equations of Some Power Series Distributions	3
2.1	Special families of the TIIHLTL-GPS	13
2.2	Generated Random Variables of the TIIHLTL-WP Distribution	23
2.3	Table of Moments for Selected Parameters of the TIIHLTL-WP Distribution . .	23
2.4	Generated Random Variables of the TIIHLTL-WG Distribution	25
2.5	Table of Moments for Selected Parameters of the TIIHLTL-WG Distribution . .	26
2.6	Generated Random Variables of the TIIHLTL-WL Distribution	27
2.7	Table of Moments for Selected Parameters of the TIIHLTL-WL Distribution . .	28
2.8	Generated Random Variables of the TIIHLTL-LLoP Distribution	29
2.9	Table of Moments for Selected Parameters of the TIIHLTL-LLoP Distribution .	30
2.10	Generated Random Variables of the TIIHLTL-LLoGG Distribution	31
2.11	Table of Moments for Selected Parameters of the TIIHLTL-LLoGG Distribution	32
2.12	Random Generated Variables of the TIIHLTL-LLoGL Distribution	33
2.13	Table of Moments for Selected Parameters of the TIIHLTL-LLoGL Distribution .	34
2.14	Monte Carlo Simulation Results for TIIHLTL-LLoP Distribution: Mean, RMSE and Average Bias	35
2.15	Parameter estimates and goodness-of-fit statistics for various models fitted for fatigue life of bearings data set	37
2.16	Parameter estimates and goodness-of-fit statistics for various models fitted for window strength data set	38
3.1	Random Generated Variables of the BTIIHLTL-U Distribution	50
3.2	Table of Moments for Selected Parameters of the BTIIHLTL-U Distribution . . .	50
3.3	Generated Random Variables of the BTIIHLTL-E Distribution	52
3.4	Table of Moments for Selected Parameters of the BTIIHLTL-E Distribution . . .	52
3.5	Generated Random Variables of the BTIIHLTL-LLoG Distribution	54
3.6	Table of Moments for Selected Parameters of the BTIIHLTL-LLoG Distribution	54
3.7	Monte Carlo Simulation Results for BTIIHLTL-E Distribution: Mean, RMSE and Average Bias	56
3.8	Yarn at 2.3% Strain Level Data	57
3.9	Parameter estimates and goodness-of-fit statistics for various models fitted for Yarn at 2.3% Level Strain Data	57
3.10	Monthly actual taxes revenue	58
3.11	Parameter estimates and goodness-of-fit statistics for various models fitted for Monthly Actual Taxes Revenue Data	58
4.1	Generated Random Variables of the TIIHLTLEG-E Distribution	69
4.2	Table of Moments for Selected Parameters of the TIIHLTLEG-E Distribution . .	70

4.3	Generated Random Variables of the TIIHLTLEG-LLoG Distribution	72
4.4	Table of Moments for Selected Parameters of the TIIHLTLEG-LLoG Distribution	72
4.5	Generated Random Variables of the TIIHLTLEG-W Distribution	74
4.6	Table of Moments for Selected Parameters of the TIIHLTLEG-W Distribution .	74
4.7	Monte Carlo Simulation Results for TIIHLTLEG-E Distribution: Mean, RMSE and Average Bias	76
4.8	Filure Times of Air Conditioning System Data	77
4.9	Parameter estimates and goodness-of-fit statistics for various models fitted for Failure Times of the Air Conditioning System of an Airplane Data	78
4.10	Survival Times of Guinea Pigs Data	79
4.11	Parameter estimates and goodness-of-fit statistics for various models fitted for Survival Times of Guinea Pigs Injected with Different Doses of Tubercle Bacilli Data.	80



DECLARATION REGARDING THE WORK AND COPYRIGHT

Candidate Katlego Keganne

Student ID: 14000820

Thesis Titled: Some Generalizations of the Type II Half Logistic Topp-Leone Distributions

I **Katlego Keganne**, certify that the Thesis is all my own original work and that I have not obtained a degree in this University or elsewhere on the basis of any of this work.

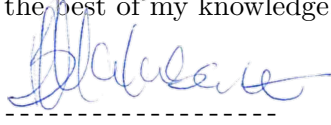
This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright and Neighbouring Rights Act, Act. No. 8 of 2000 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Postgraduate School, on behalf of both the author and the BIUST.

Signed: - 

Date:- 03/04/2023

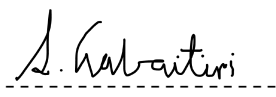
Primary Supervisors Prof. Boikanyo Makubate and Dr. Lesego Gabaitiri

I **Prof. Boikanyo Makubate**, hereby confirm that I have inspected the above titled thesis and, to the best of my knowledge, it is based on the original work of the candidate.

Signed: 

Date: April 19, 2023

I **Dr. Lesego Gabaitiri**, hereby confirm that I have inspected the above titled thesis and, to the best of my knowledge, it is based on the original work of the candidate.

Signed: 

Date: April 19, 2023

Abstract

We present three new extensions of the Type II Half Logistic Topp-Leone-G (TIIHLTL-G) distribution: the Type II Half Logistic Topp-Leone-G Power Series (TIIHLTL-GPS) distribution, the Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) distribution and the Type II Half Logistic Topp-Leone Exponentiated Generalized-G (TIIHLTLEG-G) distribution. We are motivated by the frequent failure of classical statistical distributions in offering a good fit to actual data which is particularly due to their lack of flexibility. Statistical properties of the new distributions are explored, methods of finding estimators such as the Maximum Likelihood were used and we also assessed the performance of the distributions by carrying out a simulation study for the special cases of each distribution. Finally, we applied our distributions to real data sets to show that they are useful when applied to real data. We conclude that the new distributions offer a good fit in comparison to some existing methods and improve the flexibility of some classical distributions.

Keywords: Generalized Power Series Family of Distributions; Beta G Family of Distributions; Exponentiated Generalized G Family of Distributions; Type II Half Logistic Topp-Leone-G; Simulation Studies.

Dedications

In loving memory of Beauty Portia Keganne, my late mother. Thank you for inspiring me to pursue higher education.



Acknowledgements

I would like to thank the seniors in my department who shared resources and valuable information with me towards writing and completing this dissertation.



1. Literature Review

1.1 Introduction

Lifetime distributions have a practical significance in probability and statistics as well as in other related areas. The Half Logistic distribution by Balakrishnan [16] with Topp-Leone distribution by Nadarajah and Kotz [72] and Topp and Leone [95] are particularly useful lifetime models. All these have been used for data analysis in many fields including finance, reliability and economics. There are several ways of generalizing any continuous distribution, $G(x)$, and they include the Exponentiated Generalized-G (EGG) as discussed by Aryal and Yousof [13], Cordeiro et al. [36], T-X method by Alzaatreh et al. [9], Aljarrah et al. [3], Marshall-Olkin-G generator by Marshall and Olkin [64], Yousof et al. [97], competing risks by Allignol et al. [5], Fine and Lindqvist [40] and compounding method by Leahu et al. [56], Louzade et al. [58], Tahir and Cordeiro [94]. The special cases of T-X method include Kumaraswamy-G by Cordeiro and de Castro [33], Nadarajah et al. [74], beta-G by Eugene et al. [39], Zografos and Balakrishnan [99], McDonald-G by Alexander et al. [1], Cordeiro et al. [35], gamma-G by Pinho et al. [79], Ristic and Balakrishnan [82], Zografos and Balakrishnan [99] and Topp-Leone generator by Sangsanit & Bodhisuwan [83].

For $0 < t < 1$, the cumulative distribution function (cdf) and probability density function (pdf) of the Topp-Leone(TL) distribution with shape parameter $b > 0$ is given by

$$R_{TL}(t; b) = [1 - (1 - t)^2]^b \quad \text{and} \quad r_{TL}(t; b) = 2b(1 - t)[1 - (1 - t)^2]^{b-1}, \quad (1.1)$$

respectively. For $y > 0$ and $\beta > 0$, the cdf and pdf of the Half Logistic (HL) distribution is given by

$$F_{HL}(y; \beta) = \frac{1 - e^{-\beta y}}{1 + e^{-\beta y}} \quad \text{and} \quad f_{HL}(y; \beta) = \frac{2\beta e^{-\beta y}}{[1 + e^{-\beta y}]^2}, \quad (1.2)$$

respectively.

Using the T-X method, Al-Shomrani et al. [7] obtained the Topp-Leone family of distributions called the Topp-Leone-G (TL-G) family of distribution, with cdf and pdf given by

$$R_{TL-G}(x; b, \xi) = [1 - (1 - G(x; \xi))^2]^b, \quad (1.3)$$

and

$$r_{TL-G}(x; b, \xi) = 2bg(x; \xi)(1 - G(x; \xi))[1 - (1 - G(x; \xi))^2]^{b-1}, \quad (1.4)$$

respectively, where $G(x; b, \xi)$ is any continuous cdf and $g(x; b, \xi) = \frac{dG(x; b, \xi)}{dx}$. Topp-Leone-G family of distributions include the Topp-Leone generalized exponential by Sangsanit and Bud-

hisuwan [83] founded in the T-X method, Topp-Leone-Marshall-Olkin-G family by Chipepa et al. [30] used the T-X method, Type II power Topp-Leone generated family by Bantan et al. [17], also obtained through the T-X method, the Exponentiated Odd Weibull-Topp-Leone-G by Chamunorwa et al. [27] via the Exponentiated Generalized-G method and the Type I Topp-Leone q-Exponential distribution by Sebastian et al. [84] based on Topp-Leone generator.

In their article, Hassan et al. [45] used the half logistic generator to derive the Type II Half Logistic-G (TIIHL-G) family of distributions. The cdf and the pdf of TIIHL-G distribution is given by

$$F_{TIIHL-G}(x; \xi) = 1 - \frac{1 - G(x; \xi)}{1 + G(x; \xi)} \quad \text{and} \quad f_{TIIHL-G}(x; \xi) = \frac{2g(x; \xi)}{(1 + G(x; \xi))^2}, \quad (1.5)$$

respectively. Thus, for any baseline cdf, $G(x; \xi)$, the cdf and pdf of the Type II Half Logistic Topp-Leone-G (TIIHLTL-G) distribution is given by

$$F_{TIIHLTL-G}(x; b, \xi) = 1 - \frac{1 - \left(1 - \bar{G}^2(x; \xi)\right)^b}{1 + \left(1 - \bar{G}^2(x; \xi)\right)^b}, \quad (1.6)$$

and

$$f_{TIIHLTL-G}(x; b, \xi) = \frac{4bg(x; \xi)\bar{G}(x; \xi)\left(1 - \bar{G}^2(x; \xi)\right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi)\right)^b\right)^2}, \quad (1.7)$$

where $b > 0$ is a shape parameter and ξ is a vector of parameters. The corresponding TIIHLTL-G survival function is given by

$$S_{TIIHLTL-G}(x; b, \xi) = \frac{1 - \left(1 - \bar{G}^2(x; \xi)\right)^b}{1 + \left(1 - \bar{G}^2(x; \xi)\right)^b}. \quad (1.8)$$

The purpose of the study is to propose new distributions that extend the TIIHLTL-G distribution and improve flexibility of classical distributions. We are motivated by the frequent failure of classical statistical distributions in offering a good fit to actual data. The Topp-Leone distribution has been gaining attention due to its bathtub shaped hazard function and the Half Logistic distribution for its popularity in modelling survival data.

1.2 Some Useful Classical Distributions

This section contains some important distributions that were used throughout the thesis.

1.2.1 Power Series Distribution

Let $(a_1, a_2, a_3, \dots)$ be a sequence of non negative numbers. The power series is defined by

$$C(\theta) = \sum_{n=1}^{\infty} a_n \theta^n, \quad (1.9)$$

for $\theta > 0$. If N is a random variable contained in the set of natural numbers, then N has a power series distribution associated with $C(\theta)$ with probability mass function given by

$$f_{\theta}(n) = \frac{a_n \theta^n}{C(\theta)}, \quad (1.10)$$

where $n \in \mathbb{N}$. The associated cdf of the power series is given by

$$F_{\theta}(n) = \frac{C_n(\theta)}{C(\theta)}, \quad (1.11)$$

where

$$C_n(\theta) = \sum_{k=1}^n a_k \theta^k.$$

Special power series distributions include the Logarithmic distribution, Geometric distribution, Binomial distribution and Poisson distribution, Johnson et al. [51]. See Table 1.1 for some important equations of the distributions

Distribution	$C(\theta)$	$C'(\theta)$	$C''(\theta)$	$C(\theta)^{-1}$	a_n	Parameter Space
Poisson	$e^{\theta} - 1$	e^{θ}	e^{θ}	$\log(1 + \theta)$	$(n!)^{-1}$	$(0, \infty)$
Geometric	$\theta(1 - \theta)^{-1}$	$(1 - \theta)^{-2}$	$2(1 - \theta)^{-3}$	$\theta(1 + \theta)^{-1}$	1	$(0, 1)$
Logarithmic	$-\log(1 - \theta)$	$(1 - \theta)^{-1}$	$(1 - \theta)^{-2}$	$1 - e^{-\theta}$	n^{-1}	$(0, 1)$
Binomial	$(1 + \theta)^n - 1$	$n(1 + \theta)^{n-1}$	$n(n-1)(1 + \theta)^{n-2}$	$(\theta - 1)^{1/n} - 1$	$\binom{n}{k}$	$(0, 1)$

Table 1.1: Equations of Some Power Series Distributions

1.2.2 Uniform Distribution

For $x > 0$, the cdf and pdf of a uniform distribution with a scale parameter $\theta > 0$ is given by

$$F_u(x; \theta) = \frac{x}{\theta} \quad \text{and} \quad f_u(x; \theta) = \frac{1}{\theta}, \quad (1.12)$$

respectively for $0 < x < \theta$.

1.2.3 Exponential Distribution

For $x > 0$, the cdf and pdf of an exponential distribution with scale parameter $\lambda > 0$ is given by

$$F_E(x; \lambda) = 1 - e^{-\lambda x} \quad \text{and} \quad f_E(x; \lambda) = \lambda e^{-\lambda x}, \quad (1.13)$$

respectively.

1.2.4 Weibull Distribution

Weibull distributions are used widely in reliability and survival analysis. The cdf and pdf of the Weibull distribution with scale parameter α and shape parameter λ is given by

$$F_w(x; \alpha, \lambda) = 1 - e^{-\lambda x^\alpha} \quad \text{and} \quad f_w(x; \alpha, \lambda) = \alpha \lambda x^{\alpha-1} e^{-\lambda x^\alpha}, \quad (1.14)$$

respectively for $x > 0$, $\alpha > 0$, $\lambda > 0$.

1.2.5 Log Logistic Distribution

For $x > 0$, the cdf and pdf of the Log Logistic distribution with shape parameter $d > 0$ is given by

$$F_{LLoG}(x; d) = 1 - (1 + x^d)^{-1} \quad \text{and} \quad f_{LLoG}(x; d) = dx^{d-1}(1 + x^d)^{-2}, \quad (1.15)$$

respectively.

1.3 Some Important Generators

1.3.1 Beta-G Distribution and Exponentiated Generalized-G Distribution

Let $G(x; \xi)$ be the cdf of a random variable X , where ξ is a vector of parameters. The cdf of a beta-G family of distribution by Eugene et al. [39] is given by

$$F(x; \alpha, \beta, \xi) = \frac{1}{B(\alpha, \beta)} \int_0^{G(x; \xi)} t^{\alpha-1} (1-t)^{\beta-1} dt \quad (1.16)$$

$$= \frac{1}{B(\alpha, \beta)} \sum_{j=1}^{\infty} \binom{\beta-1}{j} \frac{(-1)^j}{\alpha+j} [G(x; \xi)]^{\alpha+j} \quad (1.17)$$

The corresponding pdf is given by

$$f(x; \alpha, \beta, \xi) = \frac{g(x; \xi)}{B(\alpha, \beta)} [G(x; \xi)]^{\alpha-1} [\bar{G}(x; \xi)]^{\beta-1} \quad (1.18)$$

$$= \frac{g(x; \xi)}{B(\alpha, \beta)} \sum_{j=1}^{\infty} \binom{\beta-1}{j} \frac{(-1)^j}{\alpha+j} [G(x; \xi)]^{\alpha+j-1} \quad (1.19)$$

where $\alpha > 0$, $\beta > 0$, and $B_y(\alpha, \beta) = \int_0^y t^{\alpha-1} (1-t)^{\beta-1} dt$ is the incomplete beta function, $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ is the beta function and $\Gamma(\cdot)$ is the gamma function.

The cdf and pdf of the exponentiated generalized-G (EGG) family of distribution by Cordeiro et al. [36] are given by

$$F_{EGG}(x; \alpha, \beta, \xi) = [1 - \bar{G}^\alpha(x; \xi)]^\beta, \quad (1.20)$$

$$f_{EGG}(x; \alpha, \beta, \xi) = \alpha \beta g(x; \xi) \bar{G}^{\alpha-1}(x; \xi) [1 - \bar{G}^\alpha(x; \xi)]^{\beta-1}. \quad (1.21)$$

respectively, where $\alpha, \beta > 0$ are two additional shape parameters and ξ is a vector of parameters. Some beta generated distributions include the Beta normal distribution by Eugene et al. [39], Beta Log-Logistic Weibull distribution by Makubate et al. [62], the Beta Weibull-G family of distributions by Makubate et al. [63], the Beta Odd Lindley-G family of distribution by Chipepa et al. [31], Beta generated distribution by Algeria et al. [2], Beta extended-G family by Cordeiro et al. [34], Beta extended half normal distribution by Cordeiro et al. [37], Beta Marshall-Olkin family of distributions by Alizadeh et al. [6] and generalized beta-exponential weibull distribution by Badmus et al. [14]. There are also known works of EGG distributions and these include exponentiated Weibull distribution by Mudholkar and Srivastava [70], exponentiated exponential distribution by Kundu and Gupta [55], exponentiated-type distributions by Nadarajah and Kotz [73] and the exponentiated log normal distributions by Shirke and Kakade [87].

1.3.2 Compounding Method

The probability mass function of a generalized power series (GPS) class of distributions by Momeni [69] is given by

$$P(N = m) = \frac{a(m)\theta^m}{g(\theta)} \quad (1.22)$$

where $\theta > 0$, $m \in \mathbb{N}$, $a(m) > 0$ and $g(\theta) = \sum_m a(m)\theta^m$. The corresponding pdf of N is given by

$$P_N(s) = \frac{g(\theta^s)}{g(s)} \quad (1.23)$$

for $s \in [0, 1]$.

1.4 Some Characteristics of Distributions

There are three (3) components the statisticians are looking at whenever they want to describe a viable and standard statistical distribution, these are: the shape, central tendency and variability of the distribution. A more specific description of characterization of a statistical distribution is given by Kotz [53]. In his paper titled the "Characteristics of Statistical Distributions: A Supplement to recent Surveys", Kotz states that characterization of statistical distributions can be approached from three perspectives, which are description of theorems pertaining to specific distributions, classification looking at probabilistic statistical methods and classification from mathematical techniques used in proofs. In this section we will be discussing the characteristics of distributions, mainly looking at those that are to be discussed in the chapters that cover the new distributions that we are introducing.

1.4.1 Survival, Hazard, Reverse Hazard, and Quantile Functions

The survival, hazard and reverse hazard functions are probabilistic characterization of distributions. These functions are used for analysis and modeling in survival analysis. According to Kartsonaki [52], survival analysis is the analysis of time to event data. The data describes the length of time from a time of origin to end point of interest. The hazard function models data by performing an analysis of the likelihood of survival to a certain point based on survival to

an earlier time. This function is useful when modeling censored data. Reverse hazard function describes the probability of an immediate past failure, given that the item has already failed at time y , as opposed to the immediate future failure given that the item has not yet failed at time y , (Gurler) [43].

For a continuous random variable Y , the survival function gives the likelihood that an event of interest will occur beyond any specified time y . The survival function is given by the following equation

$$S(y) = \bar{F}(y) = 1 - F(y) = \int_t^{\infty} f(y)dy, \quad (1.24)$$

The hazard function and the reverse hazard functions are given by

$$h(y) = \frac{f(y)}{S(y)} \quad \text{and} \quad \tau(y) = \frac{f(y)}{F(y)}, \quad (1.25)$$

respectively, where $F(y)$ is the cdf and $f(y)$ is the pdf. The quantile function specifies how much of your data lies below a given value. It is the generalized inverse of a given cdf, $F(y)$.

1.4.2 Skewness and Kurtosis

Skewness is a characterization of statistical distributions that measures asymmetry. It has 3 categories, namely positive also known as right, negative also known as left and zero. That is to mean that a distribution can either be left skewed, right skewed or have a skewness value of zero. There are various ways that can be used to measure skewness and these include the pearson mode technique, the Bowley technique, Kelly's measure technique and the momental technique. On the other hand kurtosis is a characterization technique that measures whether data is heavy tailed or light tailed. Three (3) categories of kurtosis displayed by a set of data are meoskurtic, leptokurtic and platykurtic. Meoskurtic distributions have a kurtosis value equal to three, indicating that the distribution is similar to the normal distribution. Leptokurtic distributions have a kurtosis value that is greater than three which is indication for distributions with long tails and platykurtic distributions have short tails with a kurtosis value less than three. Suppose that X is a real valued random variable, the mathematical formula of skewness and kurtosis is given respectively by

$$skew(X) = E \left[\left(\frac{X - \mu}{\sigma} \right)^3 \right], \quad (1.26)$$

$$kurt(X) = E \left[\left(\frac{X - \mu}{\sigma} \right)^4 \right], \quad (1.27)$$

where $E(X) = \mu$ and $\sigma^2 = \text{Var}(X)$.

1.4.3 Moments and Moment Generating Functions

The moments of a function are a set of statistical parameters that give specific quantitative measures of the shape of a statistical distribution. The measure is the r^{th} power of the deviation

taken from an arbitrary point of a statistical distribution, that is the arithmetic means of the 1^{st} , 2^{nd} , 3^{rd} and so forth. The first 4 moments describe the mean, variance, skewness and kurtosis of a distribution. Mathematically the r^{th} moment of a continuous random variable X is given by

$$E(X^r) = \int_{-\infty}^{\infty} x^r f(x) dx, \quad (1.28)$$

where, $f(x)$ is the given pdf. Moment generating functions (MGF) are used to obtain the moments. Mathematically, a moment generating function of a random variable X is given by

$$M_x(t) = E(e^{tx}). \quad (1.29)$$

A random variable X is said to have an MGF if

- $M_x(t)$ exists for X
- The $M_x(t)$ has a finite value for all t belonging to the set $[-a, a]$, where a is any positive real number.

1.4.4 Inequality Curves

A study by Arcani and Porro [11] states that inequality curves are used mainly in socio-economic sciences. They give a graphical presentation of the distribution of affluence. These curves can also be applied in defining orderings of models, helping statisticians to understand how distribution of parameters influence inequality. Lorenz [59] and Bonferroni [20] curves are two popular curves that have been studied in literature.

The Bonferroni and Lorenz curves are given by

$$B(p) = \frac{1}{p\mu} \int_0^q x f(x) dx = \frac{1}{p\mu} [\mu - T(q)], \quad (1.30)$$

$$L(p) = \mu^{-1} \int_0^q x f(x) dx = \frac{1}{\mu} [\mu - T(q)], \quad (1.31)$$

respectively, where μ is the mean distribution, $T(q) = \int_q^\infty x f(x) dx$, $q = F^{-1}(p)$ and $0 \leq p \leq 1$.

1.4.5 Mean Deviations

Mean deviations are a measure of central tendency that compute the average deviation from the average value in a given data set. It provides information about how far data values are spread out from the mean of a given data set. The mean deviation from the mean and the median deviation from the mean is given by

$$D(\mu) = \int_0^\infty |x - \mu| f(x; \xi) dx \quad (1.32)$$

$$D(M) = \int_0^\infty |x - M| f(x; \xi) dx \quad (1.33)$$

respectively.

Note that equation (1.32) and (1.33) can be written as

$$\int_0^\infty |x - \mu| f(x; \xi) dx = 2\mu F(x; \xi) - 2\mu + 2T(\mu) \quad (1.34)$$

and

$$\int_0^\infty |x - M| f(x; \xi) dx = -\mu + 2T(M) \quad (1.35)$$

respectively.

1.4.6 Entropy and Order Statistics

The general definition of entropy is a measure of the randomness of a statistical system. The concept of entropy started as an abstract mathematical property in thermodynamics and developed with the advent of statistical mechanics, (Natal et al.) [75]. According to Aristov et al. [12] entropy is widely applied in thermodynamics, statistics, communications as well as biological systems. From Clausius Entropy, Boltzmann-Gibbs Entropy to Shannon Entropy [86], this concept has been studied and further developed over different time periods. In fact many researchers across different fields are now studying Rényi entropy such as Baez [15], Bashkirov [19], Buryak and Mishura [22], Chehade and Vershynina [28], Leone et al. [57] and Yu and Deng [98]. Rényi entropy is a generalization of the Shannon entropy and is given by the following equation

$$I_R(v) = \frac{1}{1-v} \log \left(\int_0^\infty [f(x)]^v dx \right), v \neq 1, v > 0, \quad (1.36)$$

Order statistics have a wide range of applications which includes insurance policies and modeling auctions. To explain this concept, suppose we have a set of random variables $X_1, X_2, X_3, \dots, X_n$ which are independent and identically distributed. The i^{th} order statistic for this set of random variables is defined as the i^{th} smallest value of the sample. If the random variables are arranged in ascending order, then the i^{th} order statistic is shown as

$$X_1 \leq X_2 \leq \dots \leq X_i \leq \dots \leq X_n$$

The 1st order statistic $X_{(1)}$ is the set of the minimum values from the sample and the n^{th} order statistic $X_{(n)}$ is the set of the maximum values from the sample. The pdf of the i^{th} order statistic is given by

$$f_{i:n}(x) = \frac{n!}{(i-1)!(n-i)!} f(x) \sum_{k=0}^{n-i} (-1)^k \binom{n-i}{k} [F(x)]^{k+i-1}, \quad (1.37)$$

where $X_1, X_2, \dots, X_n$ is a random sample of size n from a given distribution. In particular, the pdf of the maximum and minimum is given by

$$f_{n:n}(x) = n f(x) F(x)^{n-1}, \quad (1.38)$$

and

$$f_{1:n}(x) = n f(x) [1 - F(x)]^{n-1}, \quad (1.39)$$

respectively.

1.5 Maximum Likelihood Estimation

In order to draw statistical insights about a population of interest, say for a particular study, often a sample is drawn then estimates and hypothesis tests are carried out to infer properties of the population. This is achieved by inferential statistical analysis, a method of estimation or obtaining estimates. There are two types of statistical estimates, namely the point estimate and the interval estimate. Methods of estimation include but are not limited to Maximum Likelihood Estimation, Method of Moments, Weighted Least Squares Estimation, Bayes Estimation and Least Squares Estimation. In this thesis, our preferred method of obtaining estimates is the Maximum Likelihood Estimation (MLE). This is a probabilistic optimization method for statistical density estimation. It involves obtaining a likelihood function and maximizing it in order to obtain a probability density and its estimates that explain a given data of interest. This method is also popular in machine learning, deep learning as well as satellite imaging, (Van der Laan and Rose) [96].

To explain this method, let us consider $X_1, X_2, \dots, X_n$ to be observations from n independent and identically distributed random variables. Consider also f_θ to be a pdf that depends on one or more unknown parameters $(\theta_1, \theta_2, \theta_3, \dots, \theta_m)$. The goal of MLE is to maximize the likelihood which is denoted by

$$L(\theta_1, \theta_2, \theta_3, \dots, \theta_m) = \prod_{i=1}^n (f(x_i; \theta_1, \theta_2, \theta_3, \dots, \theta_m)), \quad (1.40)$$

Maximizing the likelihood function is the same as maximizing the log function

$$\ell(\theta) = \sum_{i=1}^n \log(f(x_i; \theta_1, \theta_2, \theta_3, \dots, \theta_m)). \quad (1.41)$$

1.6 Goodness of Fit Statistics

The Goodness of Fit of a statistical model is a description of how well it fits into a set of observations by summarizing the discrepancy between the observed values and the values expected of the model, (Maydeu-Olivares and Forero) [65]. Model selection criterion include but are not limited to the Akaike Information Criterion (AIC), Corrected Akaike Information Criterion (CAIC), Bayesian Information Criterion (BIC), Kolmogorov-Smirnov test (K-S), Anderson-Darling test and the Cramér-von-Mises statistic.

1.6.1 Akaike Information Criterion (AIC)

This model selection criterion was introduced by Hirotugu Akaike [24], where he proposed a framework wherein both model and selection could be simultaneously achieved. The model to

be preferred is the one with the minimum AIC value. The AIC is given by

$$AIC = -2K \log(L) + 2K, \quad (1.42)$$

where,

K = number of estimated parameters in the model

L = maximum value of the likelihood function for the model.

1.6.2 Corrected Akaike Information Criterion (CAIC)

In their paper titled, "Regression and time series model selection in small sample", Harvich and Tsai [44] mention that a bias correction to the AIC is particularly useful when the sample is small. Furthermore, the CAIC also denoted in some books and papers as AICc, is asymptotically efficient if the true model is infinite dimensional. The CAIC is defined by

$$CAIC = AIC + \frac{2K(K+1)}{n-K-1}, \quad (1.43)$$

where,

K = number of estimated parameters in the model

L = maximum value of the likelihood function for the model.

1.6.3 Bayesian Information Criterion (BIC)

This is a popular model selection criterion also known as the Schwarz information criterion, (Cavanaugh and Neath) [23]. It is often compared to the AIC and just like it the model to be preferred is the one with the minimum BIC value. The formula of BIC is given by

$$BIC = 2 \times LL + \log(N) \times K, \quad (1.44)$$

where,

LL = log likelihood of the model

N = the number of values in a data set

K = the number of parameters in the model.

1.6.4 Kolmogorov-Smirnov Test (K-S)

This test measure the largest difference between the empirical distribution function (EDF) and a function $F(x)$ which is the true distribution. More information on the Kolmogorov-Sminorv test can be found in Stephens [93]. The Kolmogorov-Smirnov test is given by

$$K-S = \sup_x |F_n(x) - F_0(x)|, \quad (1.45)$$

where, $F_n(x)$ is the EDF and $F_0(x)$ is the hypothesized distribution.

1.6.5 Anderson-Darling (A*) and Cramér-von-Mises (W*)

These criteria are refinements of the K-S test. The A^* test and W^* test are given respectively by the following formulas

$$A^* = \sum_{i=1}^n \frac{2i-1}{n} [\ln F(Y_i) + \ln(1 - F(Y_{n+1-i}))], \quad (1.46)$$

$$W^* = \frac{1}{12n} + \sum_{i=1}^n \left[\frac{2i-1}{2n} - F(x_{(i)}) \right]^2, \quad (1.47)$$

where,

F = is the cumulative distribution of specified distribution

n = the sample size

$i = i^{th}$ sample calculated when data is sorted in ascending order.

Other model selection tests include the Shapiro-Wilk Test, the Hosmer-Lemeshow Criterion, Bayes Factor, Focused Information Criterion which were not used in this thesis.

1.7 Aim and Objectives of the Study

The aim is to generate three new distributions by extending the Type II Half Logistic Topp-Leone-G distribution and derive their properties. The specific objectives are to:

- develop three new families of distributions of the Type II Half Logistic Topp-Leone-G.
- explore mathematical and statistical characterizations of the new distributions.
- quantify parameter estimates of the new proposed distributions using the MLE method.
- illustrate the performance of the proposed models by means of simulation studies and applications to real data set.

1.8 Outline of the Thesis

In Chapter 2, we present the Type II Half Logistic Topp-Leone-G Power Series (TIIHLTL-GPS) distribution. An outline of the statistical properties of the TIIHLTL-GPS is presented, estimation statistics given and an assessment of the performance of the new distribution is carried out. The applicability of the new distribution is also illustrated. Chapter 3 introduces the Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) distribution, the statistical characterizations including special cases and a discussion on the maximum likelihood of the new distribution. A simulation study and application to real data sets of the BTIIHLTL-G are also discussed. A third extension of the TIIHLTL-G distribution is presented in Chapter 4, namely the Type II Half Logistic Topp-Leone Exponentiated Generalized-G distribution (TIIHLTLEG-G). We also present a characterization of the distribution in its various ways, give a discussion on the distribution's estimation statistics, perform a goodness-of-fit assessment of the new distribution and illustrate its applicability to real life data.

2. Type II Half Logistic Topp-Leone-G Power Series Distribution

2.1 Introduction

In this chapter, we present the new life-time distribution by compounding, (see Marshall and Olkin) [64] the TIIHLTL-G distribution defined in equation (1.7) and the power series distribution defined in equation (1.10). The general class is called the Type II Half Logistic Topp-Leone-G Power Series (TIIHLTL-GPS). Some statistical properties of the TIIHLTL-GPS including cdf, pdf, hazard function, reverse hazard function, moments, mean deviation, median deviation, Bonferroni curve and Lorenz curve are established. The pdf of TIIHLTL-GPS is also presented as an infinite linear combination of the exponentiated-G (Exp-G) distribution, (see Barreto-Souza and Simas [18]). Hence statistical properties of the TIIHLTL-GPS are derived directly from those of the Exp-G distribution. Some special models of the TIIHLTL-GPS are presented. The pdf function of the new distribution shows negative skewed, positive skewed and symmetrical shapes while the hazard rate function shows increasing, decreasing, bathtub and upside-down bathtub shapes. The maximum likelihood method is used to estimate model parameters. Simulation results are presented to determine the performance of the maximum likelihood estimators. Finally, an application of the new model to two real data is conducted.

2.2 Type II Half Logistic Topp-Leone-G Power Series Distribution

Let N be a random variable with a power series pmf defined in equation 1.10. Given N , let $X_1, \dots, X_N$ be independent and identically distributed random variables from TIIHLTL-G distribution, for $X = \min(X_1, X_2, \dots, X_N)$ the conditional cdf of X given $N = n$ is given by

$$F_{X|N=n}(x) = 1 - [S_{TIIHLTL-G}(x; b, \xi)]^n = 1 - \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^n,$$

where $S_{TIIHLTL-G}(x; b, \xi)$ is the survival function of the TIIHLTL-G distribution defined in equation (1.8). The joint cdf of X and N is given by

$$F(X \leq x, N = n) = \frac{a_n \theta^n}{C(\theta)} \left[1 - \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^n \right],$$

The cdf of the TIIHLTL-GPS is given by the marginal cdf of X, which is

$$F_{TIIHLTL-GPS}(x; \theta, b, \xi) = \sum_{n=1}^{\infty} \frac{a_n \theta^n}{C(\theta)} \left[1 - \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^n \right], \quad (2.1)$$

$$= \sum_{n=1}^{\infty} \frac{a_n \theta^n}{C(\theta)} - \sum_{n=1}^{\infty} \frac{a_n \theta^n}{C(\theta)} \left[\left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^n \right], \quad (2.2)$$

$$= 1 - \sum_{n=0}^{\infty} \frac{a_{n+1} \theta^{n+1} \left[\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right]^{n+1}}{C(\theta)}, \quad (2.3)$$

The corresponding pdf of the TIIHLTL-GPS is given by

$$f_{TIIHLTL-GPS}(x; \theta, b, \xi) = \frac{4b\theta g(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi) \right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi) \right)^b \right)^2} \times \frac{C' \left(\theta \left[\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right] \right)}{C(\theta)}. \quad (2.4)$$

where, $C(\theta) = \sum_{n=1}^{\infty} a_n \theta^n$, $C'(\theta) = \sum_{n=1}^{\infty} a_n n \theta^{n-1}$, $b > 0$, $\theta \in \mathbb{N}$, $n \in \mathbb{N}$ and ξ is vector of parameters.

Some special families of the TIIHLTL-GPS family of distributions are presented in Table 2.1.

Distribution	a_n	$C(\theta)$	cdf
TIIHLTL-G Poisson	$(n!)^{-1}$	$e^\theta - 1$	$1 - \frac{\exp \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right) - 1}{e^\theta - 1}$
TIIHLTL-G Geometric	1	$\theta(1 - \theta)^{-1}$	$1 - \frac{(1 - \theta) \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)}{\left(1 - \theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right)}$
TIIHLTL-G Logarithmic	n^{-1}	$-\log(1 - \theta)$	$1 - \frac{\log \left(1 - \theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right)}{\log(1 - \theta)}$
TIIHLTL-G Binomial	$\binom{m}{n}$	$(1 + \theta)^m - 1$	$1 - \frac{\left(1 + \theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right)^m - 1}{(1 + \theta)^m - 1}$

Table 2.1: Special families of the TIIHLTL-GPS

The survival function, hazard rate function (hrf) and reverse hazard rate function (rhrf) of the

TIHHLTL-GPS distribution is given by

$$S_{TIHHLTL-GPS}(x; \theta, b, \xi) = \frac{C\left(\theta \left[\frac{1-(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]\right)}{C(\theta)},$$

$$h_{TIHHLTL-GPS}(x; \theta, b, \xi) = \frac{4b\theta g(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi)\right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi)\right)^b\right)^2} \frac{C'\left(\theta \left[\frac{1-(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]\right)}{C\left(\theta \left[\frac{1-(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]\right)},$$

and

$$\tau_{TIHHLTL-GPS}(x; \theta, b, \xi) = \frac{4b\theta g(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi)\right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi)\right)^b\right)^2} \frac{C'\left(\theta \left[\frac{1-(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]\right)}{C(\theta) - C\left(\theta \left[\frac{1-(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]\right)}.$$

respectively.

2.2.1 Quantile Function

By inverting TIHHLTL-GPS distribution, $F_{TIHHLTL-GPS}(x; \theta, b, \xi) = u$, $0 \leq u \leq 1$, we obtain the quantile function.

$$1 - \frac{C(\theta S_{TIHHLTL-G}(x; b, \xi))}{C(\theta)} = u,$$

so that

$$C(\theta S_{TIHHLTL-G}(x; b, \xi)) = C(\theta)(1 - u).$$

This is equivalent to

$$C^{-1}(C(\theta)(1 - u)) = \theta S_{TIHHLTL-G}(x; b, \xi),$$

$$C^{-1}(C(\theta)(1 - u)) - \theta S_{TIHHLTL-G}(x; b, \xi) = 0, \quad (2.5)$$

Random TIHHLTL-GPS variables can be generated based on the non-linear equation (2.5) which is solved using numerical methods.

Corollary 0.1. The pdf of TIHHLTL-GPS given in equation (2.4) can be expressed as

$$\begin{aligned} f_{TIHHLTL-GPS}(x; \theta, b, \xi) &= \sum_{n=0}^{\infty} \frac{(n+1)a_{n+1}\theta^{n+1}}{C(\theta)} \left[\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right]^n \\ &\quad \times \frac{4bg(x; \xi) \bar{G}(x; \xi) (1 - \bar{G}^2(x; \xi))^{b-1}}{[1 + (1 - \bar{G}^2(x; \xi))^b]^2} \\ &= \sum_{l=0}^{\infty} \eta_l g_l(x; \xi), \end{aligned}$$

where,

$$g_l(x; \xi) = (l+1)g(x; \xi)G^l(x; \xi) \quad (2.6)$$

is an Exponentiated-G (Exp-G) distribution with power parameter $l+1$ and

$$\begin{aligned} \eta_l = & \sum_{i,j,k,n=0}^{\infty} \frac{(n+1)a_{n+1}\theta^{n+1}}{C(\theta)} (-1)^{j+k+l} \frac{4b}{l+1} \\ & \times \binom{n+i+1}{i} \binom{n}{j} \binom{b(i+j+1)-1}{k} \binom{2k+1}{l}. \end{aligned} \quad (2.7)$$

Proof. Using the binomial expansions,

$$\begin{aligned} [1 + (1 - \bar{G}^2(x; \xi)^b)]^{-(n+2)} &= \sum_{i=0}^{\infty} \binom{-(n+2)}{i} [1 - \bar{G}^2(x; \xi)]^{bi} \\ &= \sum_{i=0}^{\infty} \binom{n+i+1}{i} [1 - \bar{G}^2(x; \xi)]^{bi}, \end{aligned}$$

$$[1 - (1 - \bar{G}^2(x; \xi)^b)]^n = \sum_{j=0}^{\infty} (-1)^j \binom{n}{j} [1 - \bar{G}^2(x; \xi)]^{bj},$$

$$[1 - \bar{G}^2(x; \xi)]^{b(i+j+1)-1} = \sum_{k=0}^{\infty} (-1)^k \binom{b(i+j+1)-1}{k} [\bar{G}^2(x; \xi)]^k$$

and

$$[\bar{G}(x; \xi)]^{2k+1} = [1 - G(x; \xi)]^{2k+1} = \sum_{l=0}^{\infty} (-1)^l \binom{2k+1}{l} G^l(x; \xi)$$

□

We get,

$$\begin{aligned} f_{TIIHLTL-GPS}(x; \theta, b, \xi) &= \sum_{i,j,k,l,n=0}^{\infty} \frac{(n+1)a_{n+1}\theta^{n+1}}{C(\theta)} (-1)^{j+k+l} \frac{4b}{l+1} \\ &\times \binom{n+i+1}{i} \binom{n}{j} \binom{b(i+j+1)-1}{k} \binom{2k+1}{l} \\ &\times (l+1)g(x; \xi)G^l(x; \xi) \\ &= \sum_{l=0}^{\infty} \eta_l g_l(x; \xi). \end{aligned} \quad (2.8)$$

Therefore, the TIIHLTL-GPS pdf can be expressed as a linear combination (infinite) of the Exp-G distributions. We can then derive the mathematical and statistical characterizations of the TIIHLTL-GPS distribution directly from those of the Exp-G distribution. More information regarding statistical properties of the Exp-G distribution can be found in Barreto-Souza and Simas [18] and Aryal and Yousof [13].

Corollary 0.2. Using equation (1.37), the pdf of the i^{th} order statistic of the TIHLTL-GPS is given by

$$\begin{aligned}
f(x)[F(x)]^{k+i-1} &= \sum_{n=0}^{\infty} \frac{(n+1)a_{n+1}\theta^{n+1}}{C(\theta)} \left[\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right]^n \\
&\quad \times \frac{4bg(x; \xi)\bar{G}(x; \xi)(1 - \bar{G}^2(x; \xi))^{b-1}}{[1 + (1 - \bar{G}^2(x; \xi))^b]^2} \\
&\quad \times \left[1 - \frac{C\left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)\right)}{C(\theta)} \right]^{k+i-1} \\
&= \sum_{r=0}^{\infty} U_r g_r(x; \xi).
\end{aligned}$$

where $g_r(x; \xi) = (r+1)g(x; \xi)G(x; \xi)^r$ is the Exp-G distribution with power parameter $r+1$ and

$$\begin{aligned}
U_r &= \frac{m!}{(i-1)!(m-i)!} \sum_{j,k,l,n,p,q=0}^{\infty} \binom{m-i}{k} \binom{k+i-1}{p} \binom{w+n}{j} \\
&\quad \times \binom{w+n+l+1}{l} \binom{b(j+l+1)-1}{q} \binom{2q+1}{r} (-1)^{p+j+r+q+k} \\
&\quad \times \frac{4(n+1)a_{n+1}\theta^{n+1+w}}{(r+1)C^{p+1}(\theta)} b_{w,p}.
\end{aligned}$$

Proof. Using binomial expansion

$$\left[1 - \frac{C\left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)\right)}{C(\theta)} \right]^{k+i-1} = \sum_{p=0}^{\infty} \binom{k+i-1}{p} (-1)^p \left[\frac{C\left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)\right)}{C(\theta)} \right]^p$$

and using the power series raised to a positive integer result by Gradshteyn and Ryzhik [42] we get

$$\left[\frac{C\left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)\right)}{C(\theta)} \right]^p = \sum_{w=0}^{\infty} \frac{b_{w,p}\theta^w}{C^p(\theta)} \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^w,$$

where $b_{w,p} = (wd_0)^{-1} \sum_{i=1}^w [p(l+1) - w] d_l b_{w-l,p}$ and $b_{0,p} = d_0^p$. Hence

$$\begin{aligned}
f(x)[F(x)]^{k+i-1} &= \sum_{w,n,p=0}^{\infty} \binom{k+i-1}{p} (-1)^p \frac{(n+1)a_{n+1}\theta^{n+1+w}}{C^{p+1}(\theta)} b_{w,p} \\
&\quad \times \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^{w+n} \\
&\quad \times \frac{4bg(x; \xi)\bar{G}(x; \xi)(1 - \bar{G}^2(x; \xi))^{b-1}}{[1 + (1 - \bar{G}^2(x; \xi))^b]^2}.
\end{aligned}$$

Applying the generalised binomial expansions

$$\begin{aligned}
(1 - (1 - \bar{G}^2(x; \xi))^b)^{w+n} &= \sum_{j=0}^{\infty} \binom{w+n}{j} (-1)^j (1 - \bar{G}^2(x; \xi))^{bj}, \\
(1 + (1 - \bar{G}^2(x; \xi))^b)^{-(w+n+2)} &= \sum_{l=0}^{\infty} \binom{w+n+2+l-1}{l} (1 - \bar{G}^2(x; \xi))^{bl}, \\
(1 - \bar{G}^2(x; \xi))^{b(j+l+1)-1} &= \sum_{q=0}^{\infty} \binom{b(j+l+1)-1}{q} (-1)^q \bar{G}^2(x; \xi)^{2q}
\end{aligned}$$

and

$$(1 - G(x; \xi))^{2q+1} = \sum_{r=0}^{\infty} \binom{2q+1}{r} (-1)^r G(x; \xi)^r,$$

□

yields

$$\begin{aligned}
f_{i:m} &= \frac{m!}{(i-1)!(m-i)!} \sum_{j,k,l,n,p,q,r=0}^{\infty} \binom{m-i}{k} \binom{k+i-1}{p} \binom{w+n}{j} \\
&\times \binom{w+n+l+1}{l} \binom{b(j+l+1)-1}{q} \binom{2q+1}{r} (-1)^{p+j+r+q+k} \\
&\times \frac{4(n+1)a_{n+1}\theta^{n+1+w}}{(r+1)C^{p+1}(\theta)} b_{w,p}(r+1)g(x; \xi)G(x; \xi)^r \\
&= \sum_{r=0}^{\infty} U_r g_r(x; \xi),
\end{aligned}$$

It then follows that the distribution of the i^{th} order statistic from the TIIHLTL-GPS class of distributions can be written as a linear combination of the Exp-G distribution with power parameter $(r+1)$.

Corollary 0.3. The Rényi Entropy of the TIIHLTL-GPS is given by

$$\begin{aligned}
I_{R(v)} &= \frac{1}{1-v} \log \left(\int_0^{\infty} (f_{TIIHLTL-GPS}(x; \theta, b, \xi))^v dx \right) \\
&= \frac{1}{1-v} \log \left(\sum_{q=0}^{\infty} z_q^* e^{1-v(I_{REG})} \right),
\end{aligned}$$

for $v > 0$, $v \neq 1$ and where

$$I_{REG} = \frac{1}{1-v} \log \left(\int_0^{\infty} \left(\left[\frac{q}{v} + 1 \right] g(x; \xi) G_{\frac{q}{v}}^{\frac{q}{v}}(x; \xi) \right)^v dx \right)$$

is the Rényi entropy of Exp-G distribution with power parameter $\frac{q}{v}$ and

$$z_q^* = \sum_{k,m,n,p,q=0}^{\infty} \frac{(4b)^v \theta^{k+v} (-1)^{m+p+q} d_{k,v}}{C^v(\theta)} \binom{k}{m} \binom{k+n-2}{n} \\ \times \binom{b(m+n+v)-v}{q} \binom{2p+v}{q}.$$

Proof. Note that $(f_{TIIHLTL-GPS}(x; \theta, b, \xi))^v$ can be written as

$$(f_{TIIHLTL-GPS}(x; \theta, b, \xi))^v = \frac{(4b\theta)^v g(x; \xi)^v \bar{G}(x; \xi)^v [1 - \bar{G}^2(x; \xi)]^{v(b-1)}}{(1 + (1 - \bar{G}^2(x; \xi))^b)^{2v}} \\ \times \left[\frac{C' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right)}{C(\theta)} \right]^v.$$

Applying the power series raised to a positive integer (see Gradshetyn and Ryzhik [42] for details), we get

$$\left[\frac{C' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right) \right)}{C(\theta)} \right]^v = \sum_{k=0}^{\infty} \frac{d_{k,v} \theta^k}{C^k(\theta)} \left(\frac{1 - (1 - \bar{G}^2(x; \xi))^b}{1 + (1 - \bar{G}^2(x; \xi))^b} \right)^k,$$

where $d_{k,v} = kb_0^{-1} \sum_{l=1}^k [v(l+1) - k] b_l d_{k-l,v}$ and $d_{0,v} = b_0^v$. Using the generalized binomial expansions

$$(1 - (1 - \bar{G}^2(x; \xi))^b)^k = \sum_{m=0}^{\infty} (-1)^m \binom{k}{m} (1 - \bar{G}^2(x; \xi))^{bm},$$

$$(1 + (1 - \bar{G}^2(x; \xi))^b)^{-(k+2v)} = \sum_{n=0}^{\infty} \binom{k+2v+n-1}{n} (1 - \bar{G}^2(x; \xi))^{bn},$$

$$(1 - \bar{G}^2(x; \xi))^{b(m+n+v)-v} = \sum_{p=0}^{\infty} (-1)^p \binom{b(m+n+v)-v}{p} \bar{G}^{2p}(x; \xi)$$

and

$$\bar{G}(x; \xi)^{2p+v} = [1 - G(x; \xi)]^{2p+v} = \sum_{q=0}^{\infty} (-1)^q \binom{2p+v}{q} G(x; \xi)^q,$$

□

we get

$$f(x; \theta, b, \xi)^v = \sum_{k,m,n,p,q=0}^{\infty} \frac{(4b)^v \theta^{k+v} (-1)^{m+p+q} d_{k,v}}{C^v(\theta)} \binom{k}{m} \binom{k+2v+n-1}{n} \\ \times \binom{b(m+n+v)-v}{q} \binom{2p+v}{q} g^v(x; \xi) G^q(x; \xi).$$

Therefore, Rényi entropy of the TIIHLTL-GPS class of distributions can be written as

$$\begin{aligned}
I_R(v) &= \frac{1}{1-v} \left(\log \sum_{k,m,n,p,q=0}^{\infty} \frac{(4b)^v \theta^{k+v} (-1)^{m+p+q} d_{k,v}}{C^v(\theta)} \binom{k}{m} \binom{k+2v+n-1}{n} \right. \\
&\quad \times \left. \binom{b(m+n+v)-v}{q} \binom{2p+v}{q} \int_0^{\infty} g^v(x; \xi) G^q(x; \xi) dx \right) \\
&= \frac{1}{1-v} \left(\log \sum_{k,m,n,p,q=0}^{\infty} \frac{(4b)^v \theta^{k+v} (-1)^{m+p+q} d_{k,v}}{C^v(\theta)} \binom{k}{m} \binom{k+2v+n-1}{n} \right. \\
&\quad \times \left. \binom{b(m+n+v)-v}{q} \binom{2p+v}{q} \text{frac1} \left[\frac{q}{v} + 1 \right] \int_0^{\infty} \left(\left[\frac{q}{v} + 1 \right] g(x; \xi) G^{\frac{q}{v}}(x; \xi) \right)^v dx \right) \\
&= \frac{1}{1-v} \log \left(\sum_{q=0}^{\infty} z_q^* e^{1-v(I_{REG})} \right),
\end{aligned}$$

2.3 Some Statistical Properties of the TIIHLTL-GPS Class of distributions

In this section, we present some of the properties of the TIIHLTL-GPS class of distributions.

2.3.1 Moments, Conditional Moments, and Generating Function

If X follows the TIIHLTL-GPS distribution and $Y \sim \text{Exp-G}(l)$ as in equation (2.6), then the r^{th} moment of the TIIHLTL-GPS class of distributions can be obtained as

$$\begin{aligned}
E(X^r) &= \int_0^{\infty} x^r \cdot f(x; \theta, b, \xi) dx \\
&= \sum_{l=0}^{\infty} \eta_l E(Y^r),
\end{aligned}$$

where $E(Y^r)$ is the r^{th} moment of the Exp-G distribution with power parameter $l+1$ and η_l is given by equation (2.7). The moment generating function (mgf) of X is given by

$$M_X(t) = \sum_{l=0}^{\infty} \eta_l M_Y(t),$$

where $M_Y(t)$ is the mgf of the Exp-G distribution with power parameter $l+1$ and η_l is given by equation (2.7).

The conditional r^{th} moment of the TIIHLTL-GPS class of distributions is given by

$$\begin{aligned}
E(X^r | X > a) &= \frac{1}{\bar{F}(a; b, \theta, \xi)} \int_a^{\infty} x^r \cdot f(x; \theta, b, \xi) dx \\
&= \frac{1}{\bar{F}(a; b, \theta, \xi)} \sum_{l=0}^{\infty} \eta_l \cdot E[Y^r I_{(Y^r > a)}],
\end{aligned}$$

where

$$E[Y^r I_{(Y^r > t)}] = \int_t^\infty y^r \cdot g_l(x; \xi) dy.$$

2.3.2 Mean Deviations

Using equations (1.32) and (1.33), $D(\mu)$ and $D(M)$ of the TIIHLTL-GPS are given by

$$D(\mu) = \int_0^\infty |x - \mu| f_{TIIHLTL-GPS}(x; \theta, b, \xi) dx = 2\mu F_{TIIHLTL-GPS}(\mu) - 2\mu + 2T(\mu),$$

and

$$D(M) = \int_0^\infty |x - M| f_{TIIHLTL-GPS}(x; \theta, b, \xi) dx = -\mu + 2T(M),$$

respectively. Note that,

$$T(\mu) = \int_\mu^\infty x \cdot f(x; \theta, b, \xi) dx = \sum_{l=0}^\infty \eta_l \int_\mu^\infty y \cdot g_l(x; \xi) dy,$$

and

$$T(M) = \int_M^\infty x \cdot f(x; \theta, b, \xi) dx = \sum_{l=0}^\infty \eta_l \int_M^\infty y \cdot g_l(x; \xi) dy.$$

where $g_l(x; \xi)$ is given in equation (2.6) and η_l is given by equation (2.7), μ and M are the mean and median respectively of $F_{TIIHLTL-GPS}(x; \theta, b, \xi)$ given in equation (2.1). Therefore,

$$D(\mu) = 2\mu F_{TIIHLTL-GPS}(\mu) - 2\mu + 2 \sum_{l=0}^\infty \eta_l \int_\mu^\infty y \cdot g_l(x; \xi) dy,$$

and

$$D(M) = -\mu + 2 \sum_{l=0}^\infty \eta_l \int_M^\infty y \cdot g_l(x; \xi) dy.$$

2.3.3 Lorenz and Bonferroni Curves

Using equations (1.30) and (1.31), for $0 < p < 1$ the Bonferroni and Lorenz curves of the TIIHLTL-GPS are given by

$$B(p) = \frac{1}{p\mu} \int_0^q x \cdot f_{TIIHLTL-GPS}(x; \theta, b, \xi) dx = \frac{1}{p\mu} \sum_{l=0}^\infty \eta_l \int_0^q x \cdot g_l(x; \xi) dx$$

and

$$L(p) = \frac{1}{\mu} \int_0^q x \cdot f_{TIIHLTL-GPS}(x; \theta, b, \xi) dx = \frac{1}{\mu} \sum_{l=0}^\infty \eta_l \int_0^q x \cdot g_l(x; \xi) dx,$$

respectively, where $\int_0^q x \cdot g_l(x; \xi) dx$ is the equation of the first moment of the Exp-G distribution with parameter $g_l(x; \xi)$ given in equation (2.6) and η_l given in equation (2.7), with $q = F^{-1}(p)$.

2.4 Estimation

The maximum likelihood estimation (MLE) method is used to estimate the model parameters. Let $X_1, \dots, X_n \sim \text{TIIHLTL-GPS}(\theta, b, \xi)$ and $\Delta = (\theta, b, \xi)^T$ be the vector of model parameters. The log-likelihood $\ell = \ell(\Delta)$ based on a random sample of size n from the TIIHLTL-GPS class of distributions is given by

$$\begin{aligned}\ell(\Delta) &= n \ln(4b\theta) + \sum_{i=1}^n \ln[g(x_i; \xi)] + \sum_{i=1}^n \ln[\bar{G}(x_i; \xi)] \\ &+ (b-1) \sum_{i=1}^n \ln[1 - \bar{G}^2(x_i; \xi)] - 2 \sum_{i=1}^n \ln[1 + (1 - \bar{G}^2(x_i; \xi))^b] \\ &+ \sum_{i=1}^n \ln \left[C' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right) \right) \right] - n \ln[C(\theta)].\end{aligned}$$

The score vector $U = (\frac{\partial \ell}{\partial \theta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})$ has elements given by

$$\frac{\partial \ell}{\partial \theta} = \frac{n}{\theta} - \frac{nC'(\theta)}{C(\theta)} + \sum_{i=1}^n \frac{C'' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right) \right) \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right)}{C' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right) \right)},$$

$$\frac{\partial \ell}{\partial b} = \frac{n}{b} + \sum_{i=1}^n \ln[1 - \bar{G}^2(x_i; \xi)] - 2 \sum_{i=1}^n \frac{(1 - \bar{G}^2(x_i; \xi))^b \ln[1 - \bar{G}^2(x_i; \xi)]}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]},$$

and

$$\begin{aligned}\frac{\partial \ell}{\partial \xi_k} &= \sum_{i=1}^n \frac{\frac{\partial g(x_i; \xi)}{\partial \xi_k}}{g(x_i; \xi)} + \sum_{i=1}^n \frac{\frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{\bar{G}(x_i; \xi)} + (b-1) \sum_{i=1}^n \frac{\frac{\partial [1 - \bar{G}^2(x_i; \xi)]}{\partial \xi_k}}{[1 - \bar{G}^2(x_i; \xi)]} \\ &+ 2 \sum_{i=1}^n \frac{2b(1 - \bar{G}^2(x_i; \xi))^{b-1} \bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]} \\ &+ \sum_{i=1}^n \frac{C'' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right) \right) 4b\theta(1 - \bar{G}^2(x_i; \xi))^{b-1} \bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{C' \left(\theta \left(\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right) \right) (1 + (1 - \bar{G}^2(x_i; \xi))^b)^2}.\end{aligned}$$

The MLE of the parameters, denoted by $\hat{\Delta}$ are obtained by solving the nonlinear equation $(\frac{\partial \ell}{\partial \theta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})^T = \mathbf{0}$. We use the Newton-Raphson numerical method to solve the nonlinear equations. Observed Fisher information matrix evaluated at $\hat{\Delta}$ is used to construct confidence intervals for the model parameters.

2.5 Some Special Cases of the TIIHLTL-GPS Family of Distributions

In this section, we present some special cases of the TIIHLTL-GPS class of distributions. We considered Weibull and Log-logistic distributions defined in equations (1.14) and (1.15) respectively.

2.5.1 Type II Half Logistic Topp-Leone Weibull Poisson (TIIHLTL-WP) Distribution

The cdf and pdf of the TIIHLTL-WP distribution are given by

$$F_{TIIHLTL-WP}(\theta, b, \beta) = 1 - \frac{\exp\left(\theta\left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right) - 1}{e^\theta - 1},$$

and

$$f_{TIIHLTL-WP}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1}e^{-2x^\beta}(1-e^{-2x^\beta})^{b-1}}{(1+(1-e^{-2x^\beta})^b)^2} \frac{\exp\left(\theta\left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right)}{e^\theta - 1},$$

respectively, for $\theta, b, \beta > 0$. The hrf is given by

$$h_{TIIHLTL-WP}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1}e^{-2x^\beta}(1-e^{-2x^\beta})^{b-1}}{(1+(1-e^{-2x^\beta})^b)^2} \frac{\exp\left(\theta\left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right)}{\exp\left(\theta\left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right) - 1}.$$

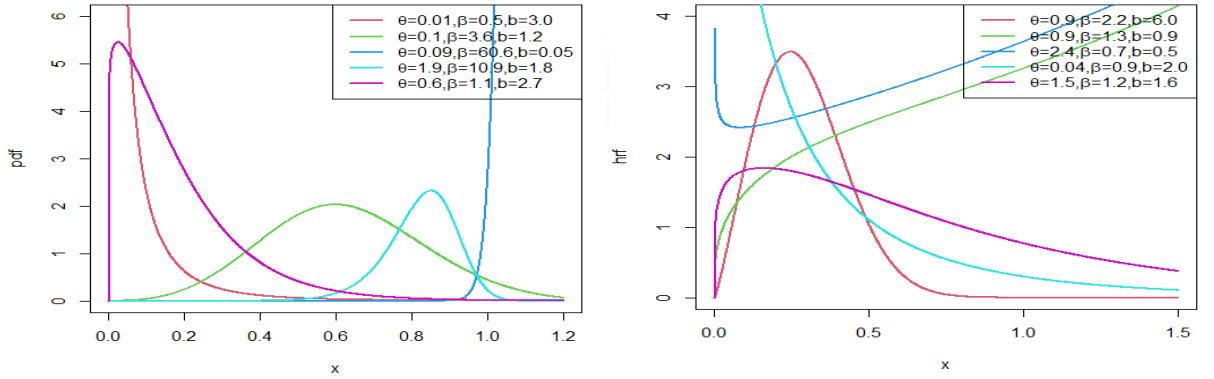


Figure 2.1: Pdf and hrf plots for the TIIHLTL-WP distribution

Figure 2.1 shows the plots of the pdfs and hrfs for the TIIHLTL-WP distribution. The pdf plots display right skewed, left skewed, reverse-J, J and almost symmetric shapes. The hrf plots display decreasing, increasing, bathtub and upside-down bathtub shapes.

	(θ, b, β)				
u	(0.1,1.4,0.8)	(0.5,1.4,1.1)	(0.9,0.7,1.8)	(0.3,1.5,0.9)	(0.6,1.2,1.4)
0.1	0.04251962	0.07918648	0.02937870	0.06485196	0.08631685
0.2	0.08420894	0.13235801	0.05778450	0.11512484	0.14037823
0.3	0.13003290	0.18470739	0.09066492	0.16633827	0.19329770
0.4	0.18237812	0.24077148	0.13091879	0.22208676	0.25003529
0.5	0.24441494	0.30431863	0.18266522	0.28584059	0.31478825
0.6	0.32079846	0.38034836	0.25305717	0.36226933	0.39309705
0.7	0.42018017	0.47745332	0.35668535	0.45952531	0.49467071
0.8	0.56123239	0.61399011	0.52952090	0.59500803	0.64077573
0.9	0.80096424	0.84631923	0.90042725	0.82121568	0.89810620

Table 2.2: Generated Random Variables of the TIIHLTL-WP Distribution

Table 2.2 lists the generated random variables of the TIIHLTL-WP distribution via equation (2.5) for selected values of the parameters $(\theta, b, \beta,)$.

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-WP distribution are given in Table 2.3.

	(θ, b, β)				
	(1.3,0.6,0.8)	(1,0.8,1.2)	(1.2,0.3,1.5)	(1.8,2.5,0.7)	(0.2,2.1,1)
$E(X)$	0.06498772	0.11698046	0.026878358	0.13104252	0.38947609
$E(X^2)$	0.03116979	0.04979469	0.010138589	0.08541981	0.22602463
$E(X^3)$	0.02001688	0.02942310	0.006036321	0.06154514	0.15014255
$E(X^4)$	0.01464438	0.02039422	0.004267908	0.04736469	0.10897247
$E(X^5)$	0.01151703	0.01546354	0.003293846	0.03816130	0.08406596
SD	0.16415353	0.19002702	0.097036814	0.26124255	0.27264080
CV	4.245200537	6.161805879	3.234838929	3.778637441	4.974970997
CS	2.52591630	1.62443390	3.610221114	1.99357091	0.70001935
CK	14.01627397	7.78647969	41.294263082	4.94251550	2.12653456

Table 2.3: Table of Moments for Selected Parameters of the TIIHLTL-WP Distribution

Figure 2.2 shows skewness and kurtosis plots for the TIIHLTL-WP distribution. When we fix the parameter b , skewness and kurtosis increases.

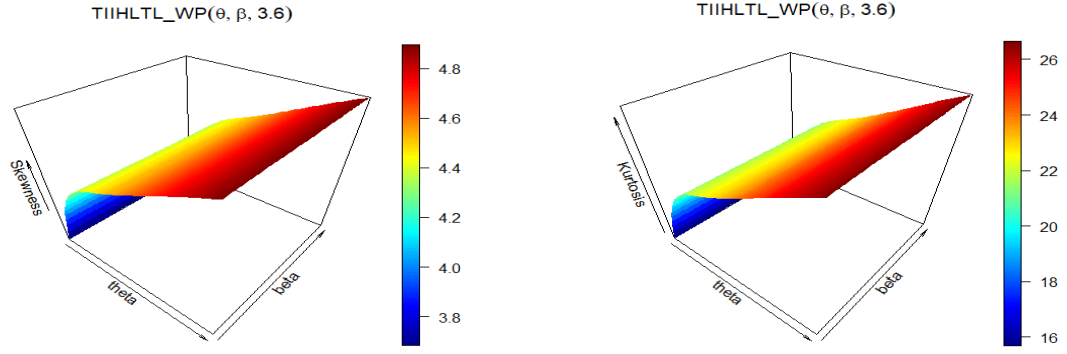


Figure 2.2: Skewness and Kurtosis plots for the TIIHLTL-WP distribution

2.5.2 Type II Half Logistic Topp-Leone Weibull Geometric (TIIHLTL-WG) Distribution

The cdf and pdf of the TIIHLTL-WG distribution are given by

$$F_{TIIHLTL-WG}(\theta, b, \beta) = 1 - \frac{(1 - \theta) \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right)}{1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right)},$$

and

$$f_{TIIHLTL-WG}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1} e^{-2x^\beta} (1 - e^{-2x^\beta})^{b-1}}{(1 + (1 - e^{-2x^\beta})^b)^2} \frac{(1 - \theta)}{\left(1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right) \right)^2},$$

respectively, for $\theta, b, \beta > 0$. The hrf is given by

$$h_{TIIHLTL-WG}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1} e^{-2x^\beta} (1 - e^{-2x^\beta})^{b-1}}{(1 + (1 - e^{-2x^\beta})^b)^2} \frac{(1 - \theta)}{\left(1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right) \right)^2} \times \left(\frac{(1 - \theta) \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right)}{1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right)} \right)^{-1}.$$

Figure 2.3 shows the plots of the pdfs and hrfs for the TIIHLTL-WG distribution. The pdf plots display right skewed, left skewed, reverse-J, J and almost symmetric shapes. The hrf plots display increasing, decreasing, bathtub and upside-down bathtub shapes.

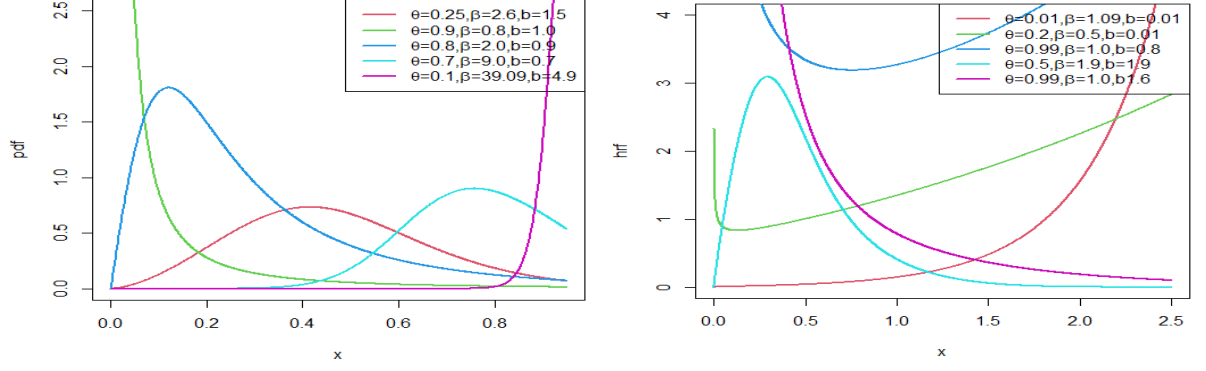


Figure 2.3: Pdf and hrf plots for the TIIHLTL-WG distribution

	(θ, b, β)				
u	(1.2,1.2,1.7)	(0.9,1.0,0.8)	(0.9,0.7,1.8)	(0.7,0.7,0.9)	(0.5,0.6,0.9)
0.1	0.1244841	0.006629968	0.04792498	0.002633445	0.001299862
0.2	0.1855164	0.017914689	0.08847316	0.008954011	0.005366049
0.3	0.2415661	0.034314233	0.13106757	0.019615787	0.013161102
0.4	0.2994603	0.057967388	0.17875643	0.036406973	0.026547041
0.5	0.3640854	0.092894700	0.23488751	0.062703342	0.048904661
0.6	0.4416853	0.147225095	0.30469848	0.105153421	0.086709587
0.7	0.5433568	0.239492823	0.39813879	0.178703094	0.154614842
0.8	0.6953140	0.422872494	0.53873096	0.324952741	0.293310390
0.9	0.6953140	0.938030313	0.80873060	0.723821311	0.678712597

Table 2.4: Generated Random Variables of the TIIHLTL-WG Distribution

Table 2.4 lists the generated random variables of the TIIHLTL-WG distribution generated via equation (2.5) for selected values of the parameters (θ, b, β) .

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-WG distribution are given in Table 2.5.

	(θ, β, b)				
	(1.3,0.6,0.8)	(1,0.8,1.2)	(1.2,0.3,1.5)	(1.8,2.5,0.7)	(0.2,2.1,1.4)
$E(X)$	0.0784	0.1874	0.0899	0.2414	0.3748
$E(X^2)$	0.0309	0.0809	0.0348	0.1167	0.2324
$E(X^3)$	0.0184	0.0481	0.0201	0.0727	0.1620
$E(X^4)$	0.0130	0.0336	0.0139	0.0519	0.1220
$E(X^5)$	0.0101	0.0256	0.0105	0.0401	0.0970
SD	0.1572	0.2139	0.1635	0.2417	0.3031
CV	2.0063	1.1413	1.8190	1.0013	0.8086
CS	3.1265	1.6192	2.7951	1.1574	0.2165
CK	13.5509	5.1766	11.3788	3.6037	1.8768

Table 2.5: Table of Moments for Selected Parameters of the TIIHLTL-WG Distribution

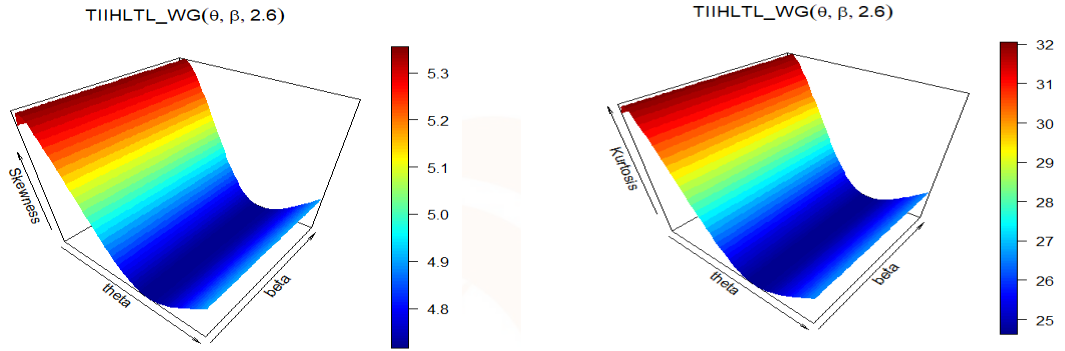


Figure 2.4: Skewness and Kurtosis plots for the TIIHLTL-WG distribution

Figure 2.4 shows skewness and kurtosis plots for the TIIHLTL-WG distribution. When we fix the parameter b , skewness and kurtosis decreases.

2.5.3 Type II Half Logistic Topp-Leone Weibull Logarithmic (TIIHLTL-WL) Distribution

The cdf and pdf of the TIIHLTL-WL distribution are given by

$$F_{TIIHLTL-WL}(\theta, b, \beta) = 1 - \frac{\log \left(1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right) \right)}{\log(1 - \theta)},$$

and

$$f_{TIIHLTL-WL}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1}e^{-2x^\beta}(1 - e^{-2x^\beta})^{b-1}}{(1 + (1 - e^{-2x^\beta})^b)^2} \frac{\left(1 - \theta \left(\frac{1 - (1 - e^{-2x^\beta})^b}{1 + (1 - e^{-2x^\beta})^b} \right) \right)^{-1}}{-\log(1 - \theta)},$$

respectively, for $\theta, b, \beta > 0$. The hrf is given by

$$h_{TIIHLTL-WL}(\theta, b, \beta) = \frac{4b\theta\beta x^{\beta-1}e^{-2x^\beta}(1 - e^{-2x^\beta})^{b-1} \left(1 - \theta \left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right)^{-1}}{(1 + (1 - e^{-2x^\beta})^b)^2} \frac{1}{-\log(1 - \theta)} \\ \times \left(\frac{\log \left(1 - \theta \left(\frac{1-(1-e^{-2x^\beta})^b}{1+(1-e^{-2x^\beta})^b}\right)\right)}{\log(1 - \theta)} \right)^{-1}.$$

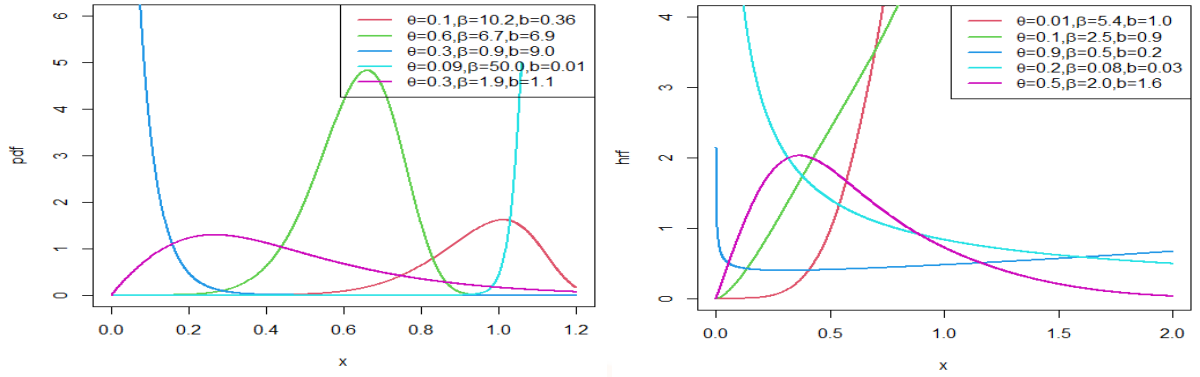


Figure 2.5: Pdf and hrf plots for the TIIHLTL-WL distribution

Figure 2.5 shows the plots of the pdfs and hrfs for the TIIHLTL-WL distribution. The pdf plots display left skewed, right skewed, reverse-J, J and symmetric shapes. The hrf plots display decreasing, increasing and upside-down bathtub shapes.

	$(\theta, b, \beta,)$				
u	(0.1,1.4,0.8)	(0.5,1.4,1.1)	(0.9,0.7,1.8)	(0.3,1.5,0.9)	(0.6,1.2,1.4)
0.1	0.04251962	0.07918648	0.02937870	0.06485196	0.08631685
0.2	0.08420894	0.13235801	0.05778450	0.11512484	0.14037823
0.3	0.13003290	0.18470739	0.09066492	0.16633827	0.19329770
0.4	0.18237812	0.24077148	0.13091879	0.22208676	0.25003529
0.5	0.24441494	0.30431863	0.18266522	0.28584059	0.31478825
0.6	0.32079846	0.38034836	0.25305717	0.36226933	0.39309705
0.7	0.42018017	0.47745332	0.35668535	0.45952531	0.49467071
0.8	0.56123239	0.61399011	0.52952090	0.59500803	0.64077573
0.9	0.80096424	0.84631923	0.90042725	0.82121568	0.89810620

Table 2.6: Generated Random Variables of the TIIHLTL-WL Distribution

Table 2.6 lists the generated random variables of the TIIHLTL-WL distribution generated via equation (2.5) for selected values of the parameters (θ, b, β) .

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-WL distribution are given in Table 2.7.

	(θ, b, β)				
	(1.3,0.6,0.8)	(1,0.8,1.2)	(1.2,0.3,1.5)	(1.8,2.5,0.7)	(0.2,2.1,1)
$E(X)$	0.06498772	0.11698046	0.026878358	0.13104252	0.38947609
$E(X^2)$	0.03116979	0.04979469	0.010138589	0.08541981	0.22602463
$E(X^3)$	0.02001688	0.02942310	0.006036321	0.06154514	0.15014255
$E(X^4)$	0.01464438	0.02039422	0.004267908	0.04736469	0.10897247
$E(X^5)$	0.01151703	0.01546354	0.003293846	0.03816130	0.08406596
SD	0.16415353	0.19002702	0.097036814	0.26124255	0.27264080
CV	4.245200537	6.161805879	3.234838929	3.778637441	4.974970997
CS	2.52591630	1.62443390	3.610221114	1.99357091	0.70001935
CK	14.01627397	7.78647969	41.294263082	4.94251550	2.12653456

Table 2.7: Table of Moments for Selected Parameters of the TIIHLTL-WL Distribution

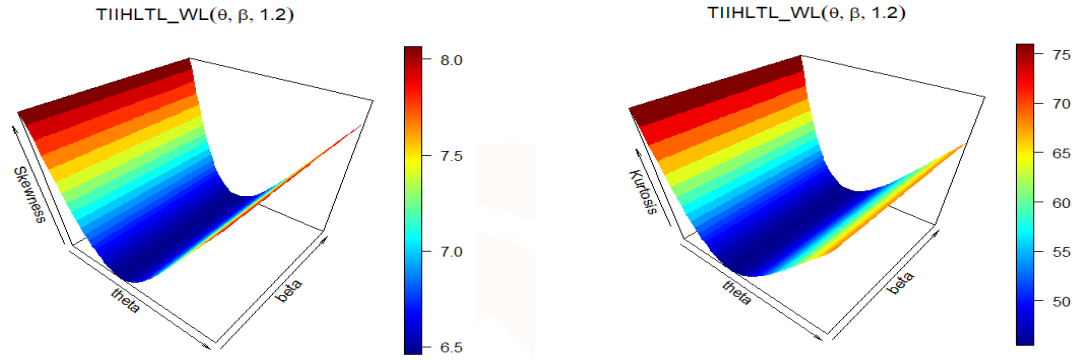


Figure 2.6: Skewness and Kurtosis plots for the TIIHLTL-WL distribution

Figure 2.6 shows skewness and kurtosis plots for the TIIHLTL-WL distribution. When we fix the parameters b , skewness and kurtosis decreases.

2.5.4 Type II Half Logistic Topp-Leone Log-logistic Poisson (TIIHLTL-LLoP) Distribution

The cdf and pdf of the TIIHLTL-LLoP distribution are given by

$$F_{TIIHLTL-LLoP}(\theta, b, c) = 1 - \frac{\exp\left(\theta\left(\frac{1-(1+(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right) - 1}{e^\theta - 1},$$

and

$$f_{TIIHLTL-LLoP}(\theta, b, c) = \frac{4b\theta c x^{c-1}(1+x^c)^{-3}(1-(1+x^c)^{-2})^{(b-1)} \exp\left(\theta\left(\frac{1-(1+(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right)}{(1+(1-(1+x^c)^{-2})^b)^2} \frac{1}{e^\theta - 1},$$

respectively, for $\theta, b, c > 0$. The hrf is given by

$$h_{TIIHLTL-LLoP}(\theta, b, c) = \frac{4b\theta c x^{c-1}(1+x^c)^{-3}(1-(1+x^c)^{-2})^{(b-1)}}{(1+(1-(1+x^c)^{-2})^b)^2} \frac{\exp\left(\theta\left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right)}{\exp\left(\theta\left(\frac{1-(1-(1+x^c)^{-1})^b}{1+(1-(1+x^c)^{-1})^b}\right)\right) - 1}.$$

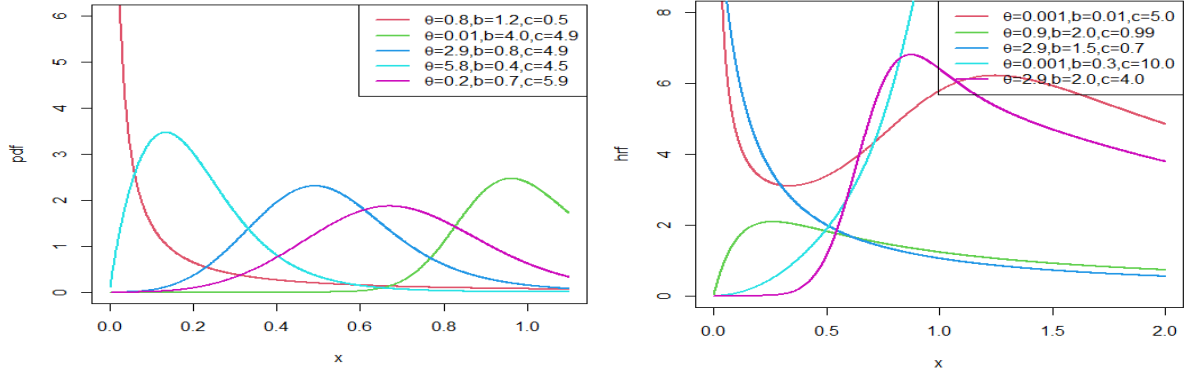


Figure 2.7: Pdf and hrf plots for the TIIHLTL-LLoP distribution

Figure 2.7 shows the plots of the pdfs and hrfs for the TIIHLTL-LLoP distribution. The pdf plots display left skewed, right skewed, symmetric and reverse-J shapes. The hrfs plots display increasing, decreasing, bathtub and bathtub followed by upside-down bathtub shapes.

	(θ, b, c)				
u	(1.2,1.2,1.7)	(0.9,1.0,0.8)	(0.9,0.7,1.8)	(0.7,0.7,0.9)	(0.5,0.6,0.9)
0.1	0.1244841	0.006629968	0.04792498	0.002633445	0.001299862
0.2	0.1855164	0.017914689	0.08847316	0.008954011	0.005366049
0.3	0.2415661	0.034314233	0.13106757	0.019615787	0.013161102
0.4	0.2994603	0.057967388	0.17875643	0.036406973	0.026547041
0.5	0.3640854	0.092894700	0.23488751	0.062703342	0.048904661
0.6	0.4416853	0.147225095	0.30469848	0.105153421	0.086709587
0.7	0.5433568	0.239492823	0.39813879	0.178703094	0.154614842
0.8	0.6953140	0.422872494	0.53873096	0.324952741	0.293310390
0.9	0.6953140	0.938030313	0.80873060	0.723821311	0.678712597

Table 2.8: Generated Random Variables of the TIIHLTL-LLoP Distribution

Table 2.8 lists the generated random variables of the TIIHLTL-LLoP distribution generated via equation (2.5) for selected values of the parameters (θ, b, c).

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-LLoP distribution are given in Table 2.9.

	(θ, b, c)				
	(1.3,0.6,0.8)	(1,0.8,1.2)	(1.2,0.3,1.5)	(1.8,2.5,0.7)	(0.2,2.1,1.4)
$E(X)$	0.0784	0.1874	0.0899	0.2414	0.3748
$E(X^2)$	0.0309	0.0809	0.0348	0.1167	0.2324
$E(X^3)$	0.0184	0.0481	0.0201	0.0727	0.1620
$E(X^4)$	0.0130	0.0336	0.0139	0.0519	0.1220
$E(X^5)$	0.0101	0.0256	0.0105	0.0401	0.0970
SD	0.1572	0.2139	0.1635	0.2417	0.3031
CV	2.0063	1.1413	1.8190	1.0013	0.8086
CS	3.1265	1.6192	2.7951	1.1574	0.2165
CK	13.5509	5.1766	11.3788	3.6037	1.8768

Table 2.9: Table of Moments for Selected Parameters of the TIIHLTL-LLoP Distribution

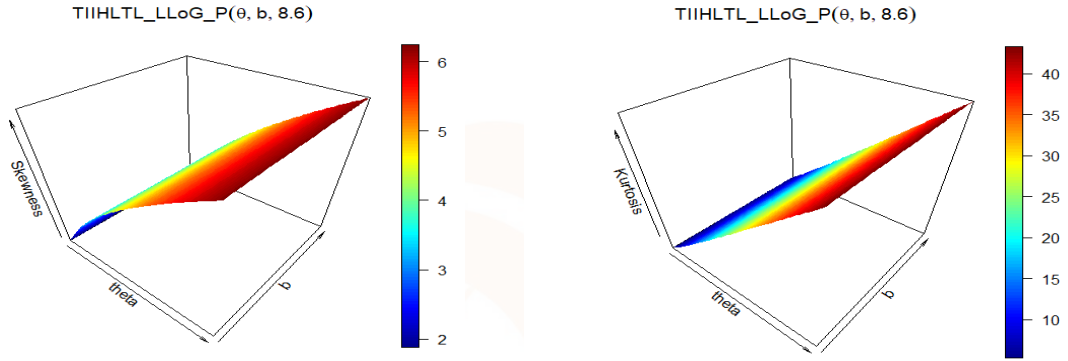


Figure 2.8: Skewness and Kurtosis plots for the TIIHLTL-LLoGP distribution

Figure 2.8 shows skewness and kurtosis plots for the TIIHLTL-LLoP distribution. When we fix the parameter c , skewness and kurtosis decreases.

2.5.5 Type II Half Logistic Topp-Leone Log-logistic Geometric (TIIHLTL-LLoGG) Distribution

The cdf and pdf of the TIIHLTL-LLoGG distribution are given by

$$F_{TIIHLTL-LLoGG}(\theta, b, c) = 1 - \frac{(1 - \theta) \left(\frac{1 - (1 - (1 + x^c)^{-2})^b}{1 + (1 - (1 + x^c)^{-2})^b} \right)}{1 - \theta \left(\frac{1 - (1 - (1 + x^c)^{-2})^b}{1 + (1 - (1 + x^c)^{-2})^b} \right)},$$

and

$$f_{TIIHLTL-LLoGG}(\theta, b, c) = \frac{4b\theta c x^{c-1} (1 + x^c)^{-3} (1 - (1 + x^c)^{-2})^{(b-1)}}{(1 + (1 - (1 + x^c)^{-2})^b)^2} \frac{(1 - \theta)}{\left(1 - \theta \left(\frac{1 - (1 - (1 + x^c)^{-2})^b}{1 + (1 - (1 + x^c)^{-2})^b} \right) \right)^2},$$

respectively, for $\theta, b, c > 0$. The hrf is given by

$$h_{TIIHLTL-LLoGG}(\theta, b, c) = \frac{4b\theta cx^{c-1}(1+x^c)^{-3}(1-(1+x^c)^{-2})^{(b-1)}}{(1+(1-(1+x^c)^{-2})^b)^2} \frac{(1-\theta)}{\left(1-\theta\left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right)^2} \\ \times \left(\frac{(1-\theta)\left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)}{1-\theta\left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)}\right)^{-1}.$$

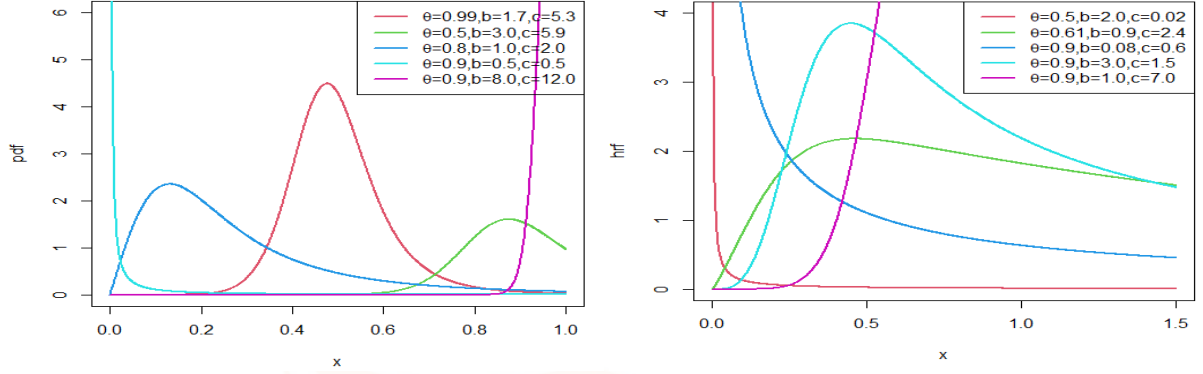


Figure 2.9: Pdf and hrf plots for the TIIHLTL-LLoGG distribution

Figure 2.9 shows the pdfs and hrfs of the TIIHLTL-LLoGG distribution. The pdf plots display shapes that include symmetric, J, reverse-J, left skewed and right skewed. Furthermore, the hrf plots display decreasing, increasing and upside-down bathtub shapes.

	(θ, b, c)				
u	(0.6,1.4,0.8)	(0.9,1.4,1.9)	(0.9,0.7,1.8)	(0.7,1.5,0.9)	(0.6,1.2,1.4)
0.1	0.01466076	0.09932004	0.01099237	0.02328787	0.06382537
0.2	0.03106558	0.13541186	0.08847316	0.04356436	0.10400362
0.3	0.05144253	0.16666041	0.03174055	0.06647969	0.14399771
0.4	0.07805015	0.19782128	0.04472373	0.09442607	0.18803332
0.5	0.11476371	0.23177930	0.06106914	0.13078253	0.24008712
0.6	0.16899450	0.27188091	0.08300493	0.18168485	0.30616229
0.7	0.25764638	0.32397211	0.11521575	0.26063490	0.39792729
0.8	0.42899647	0.40192802	0.16970675	0.40509822	0.54431992
0.9	0.90373394	0.55675133	0.29298301	0.77944299	0.85405962

Table 2.10: Generated Random Variables of the TIIHLTL-LLoGG Distribution

Table 2.10 lists the generated random variables of the TIIHLTL-LLoGG distribution generated via equation (2.5) for selected values of the parameters (θ, b, c).

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-LLoGG distribution are given in Table 2.11.

	(θ, b, c)				
	(0.8,0.5,0.7)	(0.2,0.3,0.8)	(0.5,0.8,0.6)	(0.5,0.5,0.7)	(0.8,0.8,0.4)
$E(X)$	0.019594161	0.010601420	0.035641643	0.026238036	0.015663231
$E(X^2)$	0.007303034	0.004379601	0.014563258	0.010517972	0.006317505
$E(X^3)$	0.004364504	0.002674558	0.008912333	0.006390038	0.003918095
$E(X^4)$	0.003095226	0.001910522	0.006388786	0.004561188	0.002835180
$E(X^5)$	0.002393805	0.001482224	0.004971557	0.003539441	0.002220235
SD	0.083181142	0.065323893	0.115294974	0.099144023	0.077924120
CV	4.245200537	6.161805879	3.234838929	3.778637441	4.974970997
CS	6.863584123	9.103652285	4.858196057	5.744507586	7.669411030
CK	57.850470518	98.852894967	29.565845605	40.701422333	70.483647471

Table 2.11: Table of Moments for Selected Parameters of the TIIHLTL-LLoGG Distribution

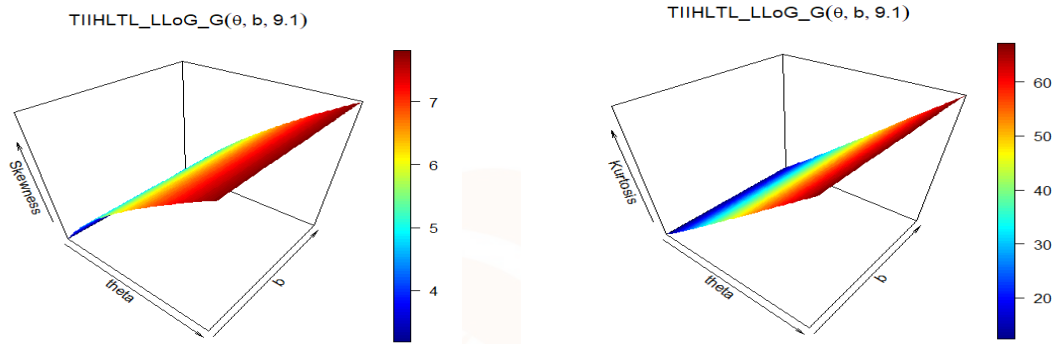


Figure 2.10: Skewness and Kurtosis plots for the TIIHLTL-LLoGG distribution

Figure 2.10 displays skewness and kurtosis plots for the TIIHLTL-LLoGG distribution. When we fix the parameter c , skewness and kurtosis increases.

2.5.6 Type II Half Logistic Topp-Leone Log-logistic Logarithmic (TIIHLTL-LLoGL) Distribution

The cdf and pdf of the TIIHLTL-LLoGL distribution are given by

$$F_{TIIHLTL-LLoGL}(\theta, b, c) = 1 - \frac{\log \left(1 - \theta \left(\frac{1 - (1 + x^c)^{-2} b}{1 + (1 - (1 + x^c)^{-2} b)} \right) \right)}{\log(1 - \theta)},$$

and

$$f_{TIIHLTL-LLoGL}(\theta, b, c) = \frac{4b\theta c x^{c-1} (1 + x^c)^{-3} (1 - (1 + x^c)^{-2})^{(b-1)} \left(1 - \theta \left(\frac{1 - (1 + x^c)^{-2} b}{1 + (1 - (1 + x^c)^{-2} b)} \right) \right)^{-1}}{(1 + (1 - (1 + x^c)^{-2} b)^2)^2 - \log(1 - \theta)},$$

respectively, for $\theta, b, c > 0$. The hrf is given by

$$h_{TIIHLTL-LLoGL}(\theta, b, c) = \frac{4b\theta cx^{c-1}(1+x^c)^{-3}(1-(1+x^c)^{-2})^{(b-1)} \left(1 - \theta \left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right)^{-1}}{(1+(1-(1+x^c)^{-2})^b)^2} \times \left(\frac{\log \left(1 - \theta \left(\frac{1-(1-(1+x^c)^{-2})^b}{1+(1-(1+x^c)^{-2})^b}\right)\right)}{\log(1-\theta)}\right)^{-1}.$$

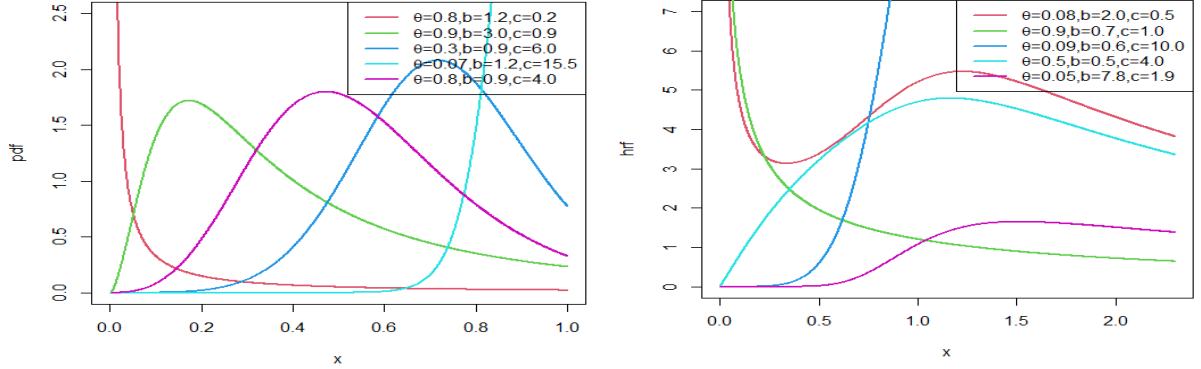


Figure 2.11: Pdfs and hrfs plots for the TIIHLTL-LLoGL distribution

Figure 2.11 shows the pdfs and hrfs of the TIIHLTL-LLoGL distribution. The pdf plots display shapes that include almost symmetric, J, reverse-J and left, or right skewed. Furthermore, the hrf plots display decreasing, increasing, upside-down bathtub and bathtub followed by upside-down bathtub shapes.

	(θ, b, c)				
u	(0.9,1.4,0.5)	(0.9,1.4,1.9)	(0.9,0.7,1.8)	(0.7,1,0.9)	(0.9,1.2,0.9)
0.1	0.0006617744	0.1445318	0.02378002	0.008991981	0.009449063
0.2	0.0022295749	0.2010741	0.04618381	0.022210603	0.021033445
0.3	0.0053540405	0.2525025	0.07205201	0.040458515	0.036275224
0.4	0.0110674490	0.3058754	0.10359391	0.065885373	0.057220173
0.5	0.0218897467	0.3657632	0.14363131	0.102440411	0.087197184
0.6	0.0433847547	0.4379296	0.19689667	0.157858798	0.132686015
0.7	0.0910129166	0.5321908	0.27246329	0.249126121	0.208009050
0.8	0.2200629401	0.6714008	0.39177602	0.423348797	0.353086880
0.9	0.7754731858	0.9352756	0.62982678	0.882432882	0.741407858

Table 2.12: Random Generated Variables of the TIIHLTL-LLoGL Distribution

Table 2.12 lists the generated random variables of the TIIHLTL-LLoGL distribution via equation (2.5) for selected values of the parameters (θ, b, c) .

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTL-LLoGL distribution are given in Table 2.13.

	(θ, b, c)				
	(0.8,0.5,0.7)	(0.2,0.3,0.8)	(0.5,0.8,0.6)	(0.5,0.5,0.7)	(0.8,0.8,0.4)
$E(X)$	0.050521036	0.05786678	0.08936005	0.068356131	0.040268250
$E(X^2)$	0.020447421	0.02415359	0.03830602	0.028516326	0.017230206
$E(X^3)$	0.012465001	0.01479343	0.02377160	0.017515295	0.010837506
$E(X^4)$	0.008911225	0.01058072	0.01714338	0.012560344	0.007888939
$E(X^5)$	0.006920906	0.00821431	0.01338391	0.009770864	0.006197926
SD	0.133772367	0.14423948	0.17412871	0.154414265	0.124934679
CV	2.647854801	2.49261281	1.94861925	2.258967300	3.102560450
CS	4.020207131	3.66152546	2.82772985	3.342450978	4.557083545
CK	20.878066146	17.57695491	11.19317374	14.960111079	25.871358579

Table 2.13: Table of Moments for Selected Parameters of the TIIHLTL-LLoGL Distribution

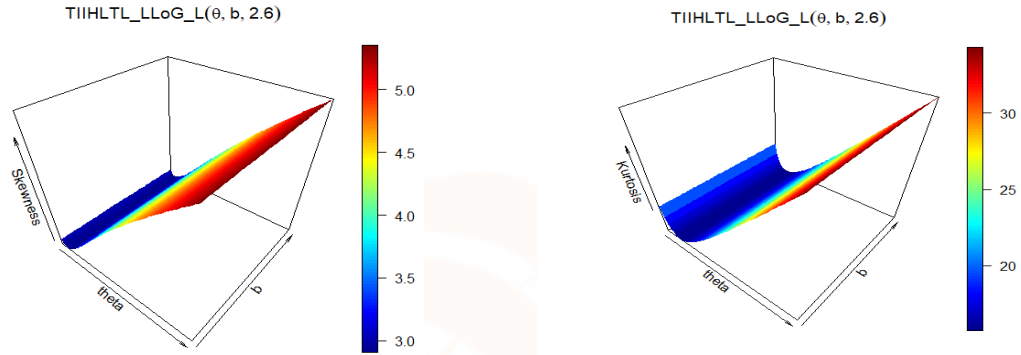


Figure 2.12: Skewness and Kurtosis plots for the TIIHLTL-LLoGL distribution

Figure 2.12 displays skewness and kurtosis plots for the TIIHLTL-LLoGL distribution. When we fix the parameter c , skewness and kurtosis increases.

2.6 Simulation Study

In this section, a simulation study was conducted to examine consistency of the maximum likelihood estimators. We considered simulations from the TIIHLTL-LLoP distribution for the sample sizes $n=25, 50, 100, 200$, and 400 , for $N=1000$ for each sample. The simulation results are shown in Table 2.14. The mean, root mean square error (RMSE), and average bias for the estimated parameter, say $\hat{\Delta}$ are calculated using the formulae

$$Bias(\hat{\Delta}) = \frac{1}{N} \sum_{i=1}^N (\hat{\Delta}_i - \Delta), \quad \text{and} \quad RMSE(\hat{\Delta}) = \sqrt{\frac{\sum_{i=1}^N (\hat{\Delta}_i - \Delta)^2}{N}},$$

respectively. If the model performs better, we expect the mean to approximate the true parameter values, the RMSE, and bias to decay toward zero for an increase in sample size. Table 2.14 shows that the mean figures indicate an approximation to the true parameter values with the RMSE and bias values decreasing towards zero for all the parameter values.

	$\theta = 1.1, b = 1.1, c = 1.3$			$\theta = 1.0, b = 1.0, c = 1.7$		
samplesize	Mean	RMSE	Bias	Mean	RMSE	Bias
θ	25	1.9227	1.9469	0.8227	1.8861	2.0490
	50	1.6248	1.1756	0.5248	1.5506	1.1879
	100	1.4016	1.0084	0.3016	1.2853	0.9126
	200	1.1910	0.6234	0.0910	1.0797	0.5675
	400	1.1365	0.4008	0.0365	1.0328	0.3614
b	25	1.4310	0.9824	0.3310	1.3391	0.9708
	50	1.3517	0.7595	0.2517	1.2400	0.7371
	100	1.2631	0.6653	0.1631	1.1383	0.5842
	200	1.1593	0.3957	0.0593	1.0455	0.3312
	400	1.1270	0.2329	0.0270	1.0210	0.1859
c	25	1.5611	1.1663	0.2611	2.0612	1.5693
	50	1.3711	0.6150	0.0711	1.8203	0.9176
	100	1.3245	0.3413	0.0245	1.7423	0.4598
	200	1.3256	0.2553	0.0256	1.7382	0.3312
	400	1.3081	0.1589	0.0081	1.7120	0.2021
	$\theta = 0.9, b = 0.9, c = 0.9$			$\theta = 1.3, b = 1.1, c = 1.1$		
θ	25	1.7585	1.6530	0.8585	2.0451	1.6957
	50	1.4703	1.1620	0.5703	1.8029	1.2254
	100	1.2365	0.9008	0.3365	1.5832	1.0119
	200	1.0255	0.5051	0.1255	1.3862	0.6361
	400	0.9847	0.3437	0.0847	1.3432	0.4298
b	25	1.2197	0.8941	0.3197	1.3905	0.9428
	50	1.1299	0.6741	0.2299	1.3438	0.7871
	100	1.0473	0.5619	0.1473	1.2519	0.6663
	200	0.9551	0.2679	0.0551	1.1574	0.4099
	400	0.9397	0.1653	0.0397	1.1313	0.2668
c	25	1.1054	0.8543	0.2054	1.3854	1.1195
	50	0.9487	0.4252	0.0487	1.1881	0.6075
	100	0.9178	0.2510	0.0178	1.1387	0.3275
	200	0.9154	0.1810	0.0154	1.1278	0.2340
	400	0.8987	0.1072	-0.0013	1.1074	0.1436

Table 2.14: Monte Carlo Simulation Results for TIIHLTL-LLoP Distribution: Mean, RMSE and Average Bias

2.7 Applications

In this section, two real data examples are presented to illustrate how useful the TIHRTL-GPS distribution is in real life. The new proposed model is compared to several non-nested models. The maximum likelihood estimation technique, via the R software, was used to estimate parameters. Goodness-of-fit statistics were used to determine how the model will perform, and they include; -2loglikelihood (-2 log L), Akaike Information Criterion (AIC), Consistent Akaike Information Criterion (CAIC), Bayesian Information Criterion (BIC), Cramer-von Mises (W^*) and Andersen-Darling (A^*) (see Chen and Balakrishnan [29] for details), Kolmogorov-Smirnov (K-S) statistic and its p-value. The model with the smallest values of these statistics and a bigger p-value for the K-S statistic is considered as the best model.

The model parameter estimates (standard errors in parenthesis) and the goodness-of-fit-statistics for the two data sets considered are shown in Tables 2.15 and 2.16. Plots of the fitted densities, the histogram of the data and probability plots (as described by Chambers et al. [26]) are provided to show how well our model fits the selected data sets.

The TIHRTL-LLoP distribution was compared to the following non-nested models: Weibull poisson (WP) distribution by Percontini et al. [80], Power Lindley logarithmic (PLL) distribution by Alkarni [4], Gompertz poisson (GoP) distribution by Jafari and Tahmasebi [48], Type II Half logistic Weibull (TIHLW) distribution by Soliman et al. [92] and Topp-Leone-Marshall-Olkin-Log logistic (TL-Mo-LLo) distribution by Chipepa et al. [30]. The pdfs of the non-nested distributions are given by;

$$f_{WP}(x; \theta, \alpha, \beta) = \frac{\theta \alpha \beta^\alpha}{e^\theta - 1} e^{-(\beta x)^\alpha} e^{\theta e^{-(\beta x)^\alpha}},$$

for $\theta, \alpha, \beta > 0$,

$$f_{PLL}(x; \theta, \alpha, \beta) = \frac{\theta \alpha \beta^2}{(\beta + 1) \log(1 - \theta)} \frac{(1 + x^\alpha) x^{\alpha-1} e^{-\beta x^\alpha}}{\left[\theta \left(\frac{\beta+1+\beta x^\alpha}{\beta+1} \right) e^{-\beta x^\alpha} \right]},$$

for $\theta, \alpha, \beta > 0$,

$$f_{GoP}(x; \theta, \beta, \gamma) = \frac{\theta \beta e^{\gamma x} e^{-\frac{\beta}{\gamma}(e^{\gamma x}-1)} e^{\theta e^{-\frac{\beta}{\gamma}(e^{\gamma x}-1)}}}{e^\theta - 1},$$

for $\theta, \beta, \gamma > 0$,

$$f_{TIHLW}(x; \lambda, \delta, \gamma) = \frac{2\lambda \delta \gamma x^{\gamma-1} e^{-\delta x^\gamma} [1 - e^{-\delta x^\gamma}]^{\lambda-1}}{[1 + [1 - e^{-\delta x^\gamma}]^\lambda]^2},$$

for $\lambda, \delta, \gamma > 0$ and

$$f_{TL-Mo-LLo}(b, \delta, c) = \frac{2b\delta^2 c x^{c-1} (1 + x^c)^{-3}}{[1 - \delta(1 + x^c)^{-1}]^3} \left[1 - \frac{\delta^2 (1 + x^c)^{-2}}{[1 - \delta(1 + x^c)^{-1}]^2} \right],$$

for $b, \delta, c > 0$.

2.7.1 Fatigue Life of Bearings Data

The first data considered was taken from McCool [66] where the fatigue life in hours of 10 bearings of a certain type was studied. The data set consists of 10 observations which have been ordered as following: 152.7, 172.0, 172.5, 173.3, 193.0, 204.7, 216.5, 234.9, 262.6, 422.6.

Model	Estimates			Statistics							
	θ	b	c	$-2 \log L$	AIC	CAIC	BIC	W^*	A^*	KS	P-value
TIHLTL-LLoGP	2.0498 (1.9170)	6.7579×10^7 (9.5171×10^{-9})	1.6478 (0.03089)	106.2	112.2	116.2	113.1	0.0305	0.2199	0.1630	0.9158
WP	θ 3.8881 (0.0790)	α 3.9120 (0.8254)	β 0.0031 (0.0003)	112.4	118.4	122.4	119.3	0.1225	0.8311	0.1919	0.7900
PLL	θ 7.6192×10^{-4} (1.0184×10^{-11})	α 2.1425 (1.0555×10^{-10})	β 1.6687×10^{-5} (2.5090×10^{-6})	113.2	119.2	123.2	120.1	0.1393	0.9284	0.2047	0.7243
GoP	θ 4.0100 (1.2126×10^{-6})	α 1.6327×10^{-4} (1.2125×10^{-4})	γ 0.0129 (3.2712×10^{-3})	116.7	122.7	126.7	123.6	0.1989	1.2490	0.2654	0.4099
TIHLW	λ 2.3092×10^4 (5.7303×10^{-6})	δ 0.9321 (0.4934)	γ 0.4465 (0.1020)	106.6	112.6	116.6	113.5	0.0368	0.2775	0.1791	0.8514
TL-Mo-LLo	b 1.1095×10^5 (2.6847×10^{-8})	δ 3.2119×10^3 (1.8480×10^{-6})	c 2.6525 (0.0307)	106.3	112.3	116.3	113.2	0.0339	0.2524	0.1788	0.8526

Table 2.15: Parameter estimates and goodness-of-fit statistics for various models fitted for fatigue life of bearings data set

The estimated variance-covariance matrix for the TIHLTL-LLoGP model on fatigue life of bearings data is

$$\begin{bmatrix} 3.6748 & 1.824 \times 10^{-8} & -0.0459 \\ 1.824 \times 10^{-8} & 39.0575 \times 10^{-17} & -2.2863 \times 10^{-10} \\ -0.0459 & -2.2863 \times 10^{-10} & 9.5398 \times 10^{-4} \end{bmatrix}$$

and the confidence intervals (95%) for parameters are $\theta \in [2.0498 \pm 3.7573]$, $b \in [6.7579 \times 10^7 \pm 1.8653 \times 10^{-8}]$ and $c \in [1.6478 \pm 0.0605]$.

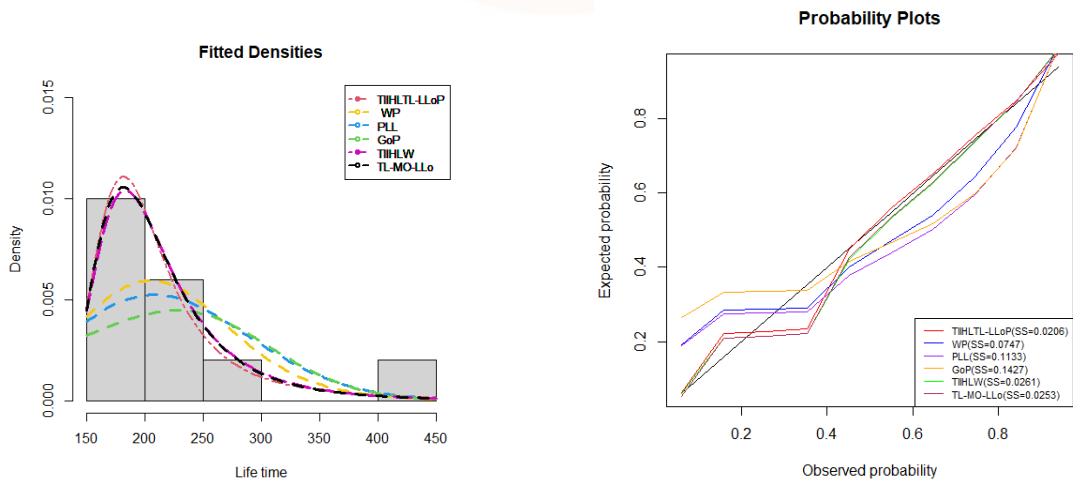


Figure 2.13: Fitted Densities and Probability Plots for the Fatigue Life of Bearings Data

From the results shown in Table 2.15, we determine that the TIHLTL-LLoP model outperforms the other models as it has the least value of goodness of fit statistics with a larger value of the

K-S p-value. Also by observing plots of the fitted densities and probability in Figure 2.13, we arrive at the same conclusion.

2.7.2 Window Strength Data

The second data set considered was reported by Fuller et al. [41], where they were studying the design of glass aircraft windows primarily looking at the strength of the glasses. The data consists of 31 observations which are: 18.83, 20.8, 21.657, 23.03, 23.23, 24.05, 24.321, 25.5, 25.52, 25.8, 26.69, 26.77, 26.78, 27.05, 27.67, 29.9, 31.11, 33.2, 33.73, 33.76, 33.89, 34.76, 35.75, 35.91, 36.98, 37.08, 37.09, 39.58, 44.045, 45.29, 45.381.

Model	Estimates			Statistics							
	θ	b	c	$-2\log L$	AIC	CAIC	BIC	W*	A*	KS	P-value
TIHLLTL-LLoGP	24.4693 (15.6359)	463.2658 (1.1052)	0.6594 (0.0281)	208.5	214.5	215.4	218.8	0.0805	0.4356	0.1356	0.5721
WP	θ 2.1095 (2.1071)	α 5.4073 (0.7995)	β 0.0264 (0.0031)	209.9	215.9	216.7	220.2	0.0927	0.5569	0.1455	0.4833
PLL	θ 0.4194 (3.6791×10^{-12})	α 3.4318 (3.1269×10^{-11})	β 1.1277×10^{-5} (1.1544×10^{-6})	209.3	215.3	216.2	219.6	0.0859	0.5172	0.1617	0.3536
GoP	θ 10.1870 (1.9118×10^{-5})	β 3.8385×10^{-4} (1.7807×10^{-4})	γ 0.0982 (0.0168)	220.9	226.9	227.8	231.2	0.1105	0.7365	0.1995	0.1477
TIHLW	λ 1018.9000 (1.0784×10^{-4})	δ 0.9022 (0.2462)	γ 0.6007 (0.0838)	209.7	215.7	216.5	220.0	0.0815	0.4616	0.1486	0.4565
TL-Mo-LLo	b 634.0444 (14.9885)	δ 118.5343 (120.5198)	c 2.4204 (0.3204)	210.4	216.4	217.3	220.7	0.0824	0.4901	0.1441	0.4955

Table 2.16: Parameter estimates and goodness-of-fit statistics for various models fitted for window strength data set

The estimated variance-covariance matrix for the TIHLLTL-LLoGP model on fatigue life of bearings data is

$$\begin{bmatrix} 244.4500 & 17.2764 & -0.4210 \\ 17.2764 & 1.2210 & -0.029 \\ -0.4210 & -0.029 & 0.0008 \end{bmatrix}$$

and the confidence intervals (95%) for parameters are $\theta \in [24.468 \pm 30.6444]$, $b \in [463.2676 \pm 2.1658]$ and $c \in [0.6594 \pm 0.0550]$.

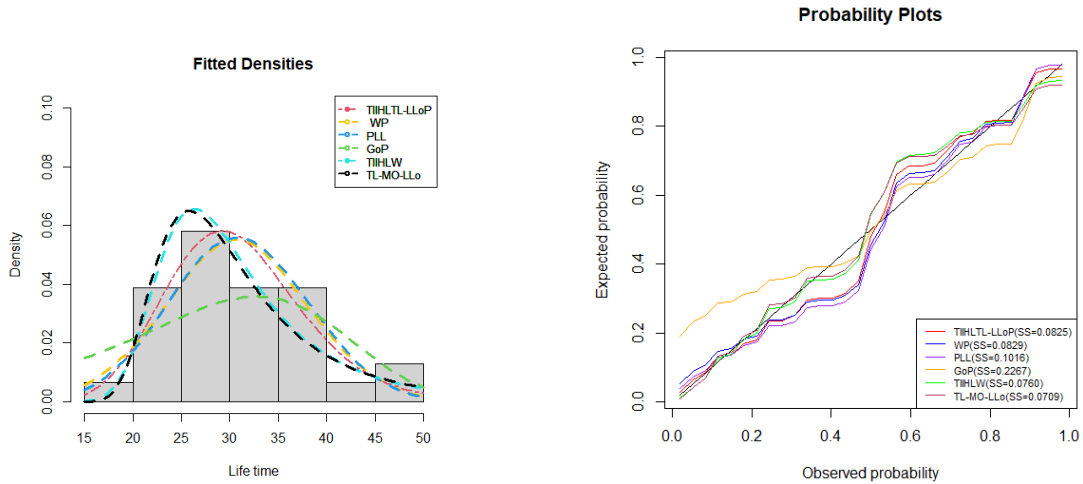


Figure 2.14: Fitted Densities and Probability Plots for the Window Strength Data

Results shown in Table 2.15 also affirms that the TIIHLTL-LLoP model performs better than the other models considered in this thesis as it has the lowest values for the goodness-of-fit statistics. The fitted densities and probability plots in Figure 2.14 also show that the TIIHLTL-LLoGP model fit the window strength data set better than the selected non-nested models.

2.8 Concluding Remarks

A new class of distributions, namely, the Type II Half Logistic Topp-Leone-G Power Series (TIIHLTL-GPS) class of distributions was developed. Special cases and structural properties of the new distribution are presented. A simulation study was conducted to assess the consistency of the maximum likelihood estimates. Two real data examples were presented to illustrate how useful the new class of distributions is. The new model performs better than several existing models that were considered.



3. Beta Type II Half Logistic Topp-Leone-G Distribution

3.1 Introduction

In this chapter, the TIIHLTL-G distribution defined in equation (1.6) is extended using a Beta-G generator by Eugene et al. [39] given in equation (1.16). The general family is called the Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G). Some statistical properties of the new distribution including cdf, pdf, hazard function, reverse hazard function, moments, mean deviation, median deviation, Bonferroni curve and Lorenz curve are established. The pdf of the new distribution is also presented as an infinite linear combination of the exponentiated-G (Exp-G) distribution (see Barreto-Souza and Simas [18]). Hence statistical properties of the new distribution are derived directly from those of the Exp-G distribution. Some special models of the new distribution are presented. The pdf function of the new distribution shows left skewed, right skewed and symmetrical shapes while the hazard rate function shows increasing, decreasing, bathtub and upside-down bathtub shapes. The maximum likelihood method is used to estimate model parameters. Simulation results are presented to assess the performance of the maximum likelihood estimators. Finally, an application of the new model to two real data is conducted.

3.2 Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) Distribution

From equation (1.16) and (1.18), the cdf and pdf of the BTIIHLTL-G distribution is given by

$$F_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j}}{(\alpha+j)}, \quad (3.1)$$

$$\begin{aligned} f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) &= \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j-1} \\ &\quad \times \frac{4bg(x; \xi)\bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi)\right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi)\right)^b\right)^2}, \end{aligned} \quad (3.2)$$

respectively, where, $\alpha > 0, \beta > 0, b > 0$, $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ is the beta function, $\Gamma(\cdot)$ is the gamma function, ξ is vector of parameters. The survival function, hrf and rhrf of the

BTIIHLTL-G distribution are given by

$$\begin{aligned}
S_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) &= 1 - \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j}}{(\alpha+j)}, \\
h_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) &= \frac{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j-1}}{1 - \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j}}{(\alpha+j)}} \\
&\quad \times \frac{4bg(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi) \right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi) \right)^b \right)^2}, \\
\tau_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) &= \frac{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j-1}}{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j}}{(\alpha+j)}} \\
&\quad \times \frac{4bg(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi) \right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi) \right)^b \right)^2}.
\end{aligned}$$

respectively.

3.2.1 Quantile Function

The quantile function is obtained by inverting $F_{BTIIHLTL-G}(x) = u$. We solve

$$\begin{aligned}
I\left[\frac{2(1-\bar{G}^2(x;\xi))^b}{1+(1-\bar{G}^2(x;\xi))^b}\right] &= u \\
\left[\frac{2(1-\bar{G}^2(x;\xi))^b}{1+(1-\bar{G}^2(x;\xi))^b}\right] &= I_{(u)}^{-1}(a, b) \\
(1-\bar{G}^2(x;\xi))^b &= \frac{I_{(u)}^{-1}(a, b)}{2-I_{(u)}^{-1}(a, b)} \\
G(x;\xi) &= 1-\sqrt{1-\sqrt[b]{\frac{I_{(u)}^{-1}(a, b)}{2-I_{(u)}^{-1}(a, b)}}}, \\
x_u &= G^{-1}\left[1-\sqrt{1-\sqrt[b]{\frac{I_{(u)}^{-1}(a, b)}{2-I_{(u)}^{-1}(a, b)}}}\right].
\end{aligned} \tag{3.3}$$

Consequently, random BTIIHLTL-G variables can be generated based on equation 3.3.

Corollary 0.4. *The pdf of the BTIIHLTL-G given in equation (3.2) can be expressed as follows*

$$f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) = \sum_{m=0}^{\infty} \eta_m g_m(x; \xi), \quad (3.4)$$

where,

$$g_m(x; \xi) = (m+1)g(x; \xi)G^m(x; \xi), \quad (3.5)$$

is an Exponentiated-G(Exp-G) distribution with power parameter $m+1$ and

$$\begin{aligned} \eta_m = & \frac{2^{\alpha+j-1}}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \sum_{k,l,m=0}^{\infty} \frac{(-1)^{l+m} 4b}{m+1} \\ & \times \binom{k+\alpha+j}{k} \binom{b(\alpha+j+k)-1}{l} \binom{2l+1}{m}. \end{aligned} \quad (3.6)$$

Proof. Using binomial expansions,

$$[1 + (1 - \bar{G}^2(x; \xi))^b]^{-(\alpha+j+1)} = \sum_{k=0}^{\infty} \binom{k+\alpha+j}{k} (1 - \bar{G}^2(x; \xi))^{bk},$$

$$[(1 - \bar{G}^2(x; \xi))^{b(\alpha+j+k)-1}] = \sum_{l=0}^{\infty} (-1)^l \binom{b(\alpha+j+k)-1}{l} (\bar{G}^2(x; \xi))^l,$$

and

$$[\bar{G}(x; \xi)]^{2l+1} = [1 - G(x; \xi)]^{2l+1} = \sum_{m=0}^{\infty} (-1)^m \binom{2l+1}{m} G^m(x; \xi),$$

□

We get,

$$\begin{aligned} f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) &= \frac{2^{\alpha+j-1}}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \sum_{k,l,m=0}^{\infty} \frac{(-1)^{l+m} 4b}{m+1} \\ &\times \binom{k+\alpha+j}{k} \binom{b(\alpha+j+k)-1}{l} \binom{2l+1}{m} (m+1)g(x; \xi)G^m(x; \xi) \\ &= \sum_{m=0}^{\infty} \eta_m g_m(x; \xi), \end{aligned}$$

Corollary 0.5. *The pdf of the i^{th} order statistic of the BTIIHLTL-G is given by*

$$\begin{aligned}
f_{i:n}(x) &= \frac{n!}{(i-1)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \left[\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j}}{(\alpha+j)} \right]^{p+i-1} \\
&\quad \times \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-\bar{G}^2(x; \xi))^b}{1+(1-\bar{G}^2(x; \xi))^b} \right]^{\alpha+j-1} \\
&\quad \times \frac{4bg(x; \xi) \bar{G}(x; \xi) \left(1 - \bar{G}^2(x; \xi)\right)^{b-1}}{\left(1 + \left(1 - \bar{G}^2(x; \xi)\right)^b\right)^2}, \\
&= \sum_{s=0}^{\infty} \eta_s g_s(x; \xi).
\end{aligned}$$

where,

$$g_s(x; \xi) = (s+1)g(x; \xi)G^s(x; \xi),$$

is an Exponentiated-G(Exp-G) distribution with power parameter $s+1$ and

$$\begin{aligned}
\eta_s &= \frac{n!}{(i-i)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \sum_{j,q,r,s=0}^{\infty} \frac{(-1)^{2j}}{(\alpha+j)(B(\alpha, \beta))} \\
&\quad \times \frac{4b}{s+1} \binom{\beta-1}{j} \binom{\beta-1}{j} \binom{i+\alpha+j+p+q-1}{q} \binom{b(i+\alpha+j+p+q-1)-1}{r} \binom{2r+1}{s}.
\end{aligned}$$

Proof. Using the series expansion representation we have,

$$[1 + (1 - \bar{G}^2(x; \xi))^b]^{-(i+\alpha+j+p)} = \sum_{q=0}^{\infty} \binom{i+\alpha+j+p+q-1}{q} (1 - \bar{G}^2(x; \xi))^{bq},$$

$$[1 - \bar{G}^2(x; \xi)]^{b(i+\alpha+j+p+q-1)-1} = \sum_{r=0}^{\infty} (-1)^r \binom{b(i+\alpha+j+p+q-1)-1}{r} \bar{G}^{2r}(x; \xi),$$

and

$$[\bar{G}(x; \xi)]^{2r+1} = [1 - G(x; \xi)]^{2r+1} = \sum_{s=0}^{\infty} (-1)^s \binom{2r+1}{s} G^s(x; \xi),$$

□

The pdf of the i^{th} order statistic reduces to

$$\begin{aligned}
f_{i:n}(x) &= \frac{n!}{(i-i)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \sum_{j,q,r,s=0}^{\infty} \frac{(-1)^{2j}}{(\alpha+j)(B(\alpha,\beta))} \\
&\times \frac{4b}{s+1} \binom{\beta-1}{j} \binom{\beta-1}{j} \binom{i+\alpha+j+p+q-1}{q} \binom{b(i+\alpha+j+p+q-1)-1}{r} \binom{2r+1}{s} \\
&\times (s+1)g(x;\xi)G^s(x;\xi), \\
&= \sum_{s=0}^{\infty} \eta_s g_s(x;\xi).
\end{aligned}$$

Corollary 0.6. The Rényi Entropy of the BTIIHLTL-G is given by

$$\begin{aligned}
I_{R(v)} &= \frac{1}{1-v} \log \left(\int_0^{\infty} (f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi))^v dx \right), \\
&= \frac{1}{1-v} \log \left(\sum_{w=0}^{\infty} s_w^* e^{(1-v)I_{REG}} \right),
\end{aligned}$$

where $v > 0$ and $v \neq 1$ and where

$$I_{REG} = \frac{1}{1-v} \log \left(\int_0^{\infty} \left(\frac{1}{[\frac{w}{v}+1]^v} g(x;\xi) G^{\frac{w}{v}}(x;\xi) \right)^v dx \right), \quad (3.7)$$

is the Rényi entropy of Exp-G distribution with $\frac{1}{[\frac{w}{v}+1]^v}$ power parameter and

$$\begin{aligned}
s_w^* &= \frac{1}{(B(\alpha,\beta))^v} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \sum_{t,u,w=0}^{\infty} (-1)^{u+w} (4b)^v \\
&\times \binom{3v-1}{t} \binom{b(t+2v)-v}{u} \binom{2u+v}{w} \frac{1}{[\frac{w}{v}+1]^v}.
\end{aligned}$$

Note that

$$\begin{aligned}
f^v(x; \alpha, \beta, b, \xi) &= \frac{1}{(B(\alpha,\beta))^v} \left[\sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-\bar{G}^2(x;\xi))^b}{1+(1-\bar{G}^2(x;\xi))^b} \right]^{\alpha+j-1} \right]^v \\
&\times \frac{(4bg(x;\xi)\bar{G}(x;\xi))^v \left(1-\bar{G}^2(x;\xi)\right)^{vb-v}}{\left(1 + \left(1-\bar{G}^2(x;\xi)\right)^b\right)^{2v}},
\end{aligned}$$

Proof. Using binomial expansions,

$$[1 + (1 - \bar{G}^2(x;\xi))^b]^{-(3v)} = \sum_{t=0}^{\infty} \binom{3v-1}{t} (1 - \bar{G}^2(x;\xi))^{bt},$$

$$[(1 - \bar{G}^2(x;\xi))^b]^{b(t+2v)-v} = \sum_{u=0}^{\infty} (-1)^u \binom{b(t+2v)-v}{u} \bar{G}^{2u}(x;\xi),$$

and

$$[\bar{G}(x; \xi)]^{2u+v} = [1 - G(x; \xi)]^{2u+v} = \sum_{w=0}^{\infty} (-1)^w \binom{2u+v}{w} G^w(x; \xi),$$

□

Therefore,

$$\begin{aligned} f^v(x; \alpha, \beta, b, \xi) &= \frac{1}{(B(\alpha, \beta))^v} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \sum_{t,u,w=0}^{\infty} (-1)^{u+w} (4b)^v \\ &\quad \times \binom{3v-1}{t} \binom{b(t+2v)-v}{u} \binom{2u+v}{w} \\ &\quad \times g^v(x; \xi) G^w(x; \xi), \end{aligned}$$

Substituting in the above equation we have

$$\begin{aligned} I_{R(v)} &= \frac{1}{1-v} \log \left(\int_0^{\infty} \left[\frac{1}{(B(\alpha, \beta))^v} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \sum_{t,u,w=0}^{\infty} (-1)^{u+w} (4b)^v \right. \right. \\ &\quad \times \left. \left. \binom{3v-1}{t} \binom{b(t+2v)-v}{u} \binom{2u+v}{w} g^v(x; \xi) G^w(x; \xi) \right] dx \right) \\ &= \frac{1}{1-v} \log \left(\sum_{w=0}^{\infty} s_w^* e^{(1-v)I_{REG}} \right), \end{aligned}$$

Therefore, the BTIIHLTL-G pdf can be expressed as a linear combination (infinite) of the Exp-G distributions. We can then derive the mathematical and statistical characterizations of the BTIIHLTL-G family of distributions directly from those of the Exp-G distribution. More information regarding statistical properties of the Exp-G distribution can be found in Barreto-Souza and Simas [18] and Aryal and Yousof [13].

3.3 Some Statistical Properties of the BTIIHLTL-G Family of Distributions

In this section, we present some of the properties of the BTIIHLTL-G class of distributions.

3.3.1 Moments, Conditional Moments, and Generating Functions

If X follows the BTIIHLTL-G distribution and $Y \sim \text{Exp-G}(m)$ as in equation (3.5), then the r^{th} moment of the BTIIHLTL-G family of distributions can be obtained as

$$\begin{aligned} E(X^r) &= \int_0^{\infty} x^r \cdot f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) dx \\ &= \int_0^{\infty} x^r \cdot \eta_m g_m(x; \xi) dx, \end{aligned}$$

where $g_m(x; \xi)$ is an Exp-G distribution with power parameter $m+1$ given in equation (3.5) and η_m is given by equation (3.6). The moment generating function is given by

$$M_x(t) = \sum_{m=0}^{\infty} \eta_m M_Y(t),$$

where $M_Y(t)$ is the moment generating function of the Exp-G distribution with power parameter $m+1$ and η_m is given by equation (3.6).

The conditional r^{th} moment of the BTIIHLTL-G distribution is

$$\begin{aligned} E(X^r | X > n) &= \frac{1}{\bar{F}(n; \alpha, \beta, b, \xi)} \int_n^{\infty} x^r \cdot f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) dx \\ &= \frac{1}{\bar{F}(n; \alpha, \beta, b, \xi)} \sum_{m=0}^{\infty} \eta_m \cdot E(Y^r I_{(Y^r > n)}), \end{aligned}$$

where,

$$E(Y^r I_{(Y^r > t)}) = \int_t^{\infty} y^r \cdot g_m(x; \xi) dy.$$

The average lifetime of X is given by

$$E(X) = \sum_{m=0}^{\infty} \eta_m E(Y).$$

where,

$$E(Y) = (m+1) \int_0^{\infty} x \cdot g(x; \xi) G^m(x; \xi) dx \quad (3.8)$$

3.3.2 Mean Deviations

Using equations (1.32) and (1.33), $D(\mu)$ and $D(M)$ of the BTIIHLTL-G distribution are given by

$$D(\mu) = \int_0^{\infty} |x - \mu| f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) dx = 2\mu F_{BTIIHLTL-G}(\mu) - 2\mu + 2T(\mu)$$

and

$$D(M) = \int_0^{\infty} |x - M| f_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi) dx = -\mu + 2T(M)$$

respectively. Note that,

$$T(\mu) = \int_{\mu}^{\infty} x \cdot f(x; \alpha, \beta, b, \xi) dx = \sum_{m=0}^{\infty} \eta_m \int_{\mu}^{\infty} y \cdot g_m(x; \xi) dy,$$

and

$$T(M) = \int_M^{\infty} x \cdot f(x; \alpha, \beta, b, \xi) dx = \sum_{m=0}^{\infty} \eta_m \int_M^{\infty} y \cdot g_m(x; \xi) dy,$$

where $g_m(x; \xi)$ is given in equation (3.5) and η_m is given by equation (3.6), μ and M are the mean and median respectively of $F_{BTIIHLTL-G}(x; \alpha, \beta, b, \xi)$ given in equation (3.1). Therefore,

$$D(\mu) = 2\mu F_{BTIIHLTL-G}(\mu) - 2\mu + 2 \sum_{m=0}^{\infty} \eta_m \int_{\mu}^{\infty} y \cdot g_m(x; \xi) dy,$$

and

$$D(M) = -\mu + 2 \sum_{m=0}^{\infty} \eta_m \int_{\mu}^{\infty} y \cdot g_m(x; \xi) dy.$$

3.3.3 Lorenz and Bonferroni Curves

For $0 < p < 1$ the Bonferroni and Lorenz curves of the BTIIHLTL-G are given by

$$\begin{aligned} B(p) &= \frac{1}{p\mu} \int_0^q x \cdot f_{BTIIHLTL-G}(x) dx \\ &= \frac{1}{p\mu} \sum_{m=0}^{\infty} \eta_m \int_0^q x \cdot g_m(x; \xi) dx, \end{aligned}$$

$$\begin{aligned} L(p) &= \frac{1}{\mu} \int_0^q x \cdot f_{BTIIHLTL-G}(x) dx \\ &= \frac{1}{\mu} \sum_{m=0}^{\infty} \eta_m \int_0^q x \cdot g_m(x; \xi) dx, \end{aligned}$$

respectively, where $\int_0^q x \cdot g_m(x; \xi) dx$ is the equation of the first moment of the Exp-G distribution with parameter $g_m(x; \xi)$ given in equation (3.5) and η_m given in equation (3.6), with $q = F^{-1}(p)$.

3.4 Estimation

The maximum likelihood estimation (MLE) method is used to estimate the model parameters. Let $X_1, \dots, X_n \sim \text{BTIIHLTL-G}(\alpha, \beta, b, \xi)$ and $\Delta = (\alpha, \beta, b, \xi)^T$ be the vector of model parameters. The log-likelihood $\ell = \ell(\Delta)$ based on a random sample of size n from the BTIIHLTL-G family of distributions is given by

$$\begin{aligned} \ell(\Delta) &= (\alpha - 1) \sum_{i=0}^n \ln \left[\frac{2(1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right] + (\beta - 1) \sum_{i=0}^n \ln \left[\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right] + n \ln(4b) \\ &\quad + \sum_{i=0}^n \ln(g(x_i; \xi)) + \sum_{i=0}^n \ln(\bar{G}(x_i; \xi)) + (b - 1) \sum_{i=0}^n \ln[1 - \bar{G}^2(x_i; \xi)] - \ln(B(\alpha, \beta)) \\ &\quad - 2 \sum_{i=0}^n \ln[1 + (1 - \bar{G}^2(x_i; \xi))^b], \end{aligned}$$

The score vector $u = (\frac{\partial \ell}{\partial \alpha}, \frac{\partial \ell}{\partial \beta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})$

$$\frac{\partial \ell}{\partial \alpha} = \sum_{i=0}^n \ln \left[\frac{2(1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right],$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=0}^n \ln \left[\frac{1 - (1 - \bar{G}^2(x_i; \xi))^b}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \right],$$

$$\begin{aligned} \frac{\partial \ell}{\partial b} &= \frac{n}{b} + \sum_{i=0}^n \ln[1 - \bar{G}^2(x_i; \xi)] \\ &+ (\alpha - 1) \sum_{i=1}^n \frac{[1 + (1 - \bar{G}^2(x_i; \xi))^b][2(1 - \bar{G}^2(x_i; \xi))^b] \ln[2(1 - \bar{G}^2(x_i; \xi))] 2[(1 - \bar{G}^2(x_i; \xi))^b]}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]^2 [1 + (1 - \bar{G}^2(x_i; \xi))^b]} \\ &- (\alpha - 1) \sum_{i=1}^n \frac{[2(1 - \bar{G}^2(x_i; \xi))^b][1 - \bar{G}^2(x_i; \xi)] \ln[1 - \bar{G}^2(x_i; \xi)] 2[(1 - \bar{G}^2(x_i; \xi))^b]}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]^2 [1 + (1 - \bar{G}^2(x_i; \xi))^b]} \\ &+ (\beta - 1) \sum_{i=1}^n \frac{[1 + (1 - \bar{G}^2(x_i; \xi))^b][1 - \bar{G}^2(x_i; \xi)] \ln[1 - \bar{G}^2(x_i; \xi)] [1 - (1 - \bar{G}^2(x_i; \xi))^b]}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]^2 [1 + (1 - \bar{G}^2(x_i; \xi))^b]} \\ &+ (\beta - 1) \sum_{i=1}^n \frac{[1 - (1 - \bar{G}^2(x_i; \xi))^b][1 - \bar{G}^2(x_i; \xi)] \ln[1 - \bar{G}^2(x_i; \xi)] [1 - (1 - \bar{G}^2(x_i; \xi))^b]}{[1 + (1 - \bar{G}^2(x_i; \xi))^b]^2 [1 + (1 - \bar{G}^2(x_i; \xi))^b]}, \end{aligned}$$

$$\begin{aligned} \frac{\partial \ell}{\partial \xi_k} &= \sum_{i=1}^n \frac{\frac{\partial g(x_i; \xi)}{\partial \xi_k}}{g(x_i; \xi)} + \sum_{i=1}^n \frac{\frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{\bar{G}(x_i; \xi)} + (b - 1) \sum_{i=1}^n \frac{\frac{\partial (1 - \bar{G}^2(x_i; \xi))}{\partial \xi_k}}{1 - \bar{G}^2(x_i; \xi)} \\ &+ 2 \sum_{i=1}^n \frac{2b(1 - \bar{G}^2(x_i; \xi))^{b-1} \bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{1 + (1 - \bar{G}^2(x_i; \xi))^b} \\ &+ (\alpha - 1) \sum_{i=1}^n \frac{[2(1 - \bar{G}^2(x_i; \xi))^b][b(1 - \bar{G}^2(x_i; \xi))^{b-1}][2\bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}][1 + (1 - \bar{G}^2(x_i; \xi))^b]}{[(1 + (1 - \bar{G}^2(x_i; \xi))^b)^2][2(1 - \bar{G}^2(x_i; \xi))^b]} \\ &- (\alpha - 1) \sum_{i=1}^n \frac{[1 + (1 - \bar{G}^2(x_i; \xi))^b][2b(1 - \bar{G}^2(x_i; \xi))^{b-1}][2\bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}][1 + (1 - \bar{G}^2(x_i; \xi))^b]}{[(1 + (1 - \bar{G}^2(x_i; \xi))^b)^2][2(1 - \bar{G}^2(x_i; \xi))^b]} \\ &+ (\beta - 1) \sum_{i=1}^n \frac{[1 + (1 - \bar{G}^2(x_i; \xi))^b][(b(1 - \bar{G}^2(x_i; \xi))^{b-1})][2\bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}][1 + (1 - \bar{G}^2(x_i; \xi))^b]}{[(1 + (1 - \bar{G}^2(x_i; \xi))^b)^2][1 - (1 - \bar{G}^2(x_i; \xi))^b]} \\ &+ (\beta - 1) \sum_{i=1}^n \frac{[1 - (1 - \bar{G}^2(x_i; \xi))^b][b(1 - \bar{G}^2(x_i; \xi))^{b-1}][2\bar{G}(x_i; \xi) \frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}][1 + (1 - \bar{G}^2(x_i; \xi))^b]}{[(1 + (1 - \bar{G}^2(x_i; \xi))^b)^2][1 - (1 - \bar{G}^2(x_i; \xi))^b]}. \end{aligned}$$

The MLE of the parameters, denoted by $\hat{\Delta}$ are obtained by solving the nonlinear equation $(\frac{\partial \ell}{\partial \alpha}, \frac{\partial \ell}{\partial \beta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})^T = \mathbf{0}$. We use the Newton-Raphson numerical method to solve the nonlinear equations. Observed Fisher information matrix evaluated at $\hat{\Delta}$ is used to construct confidence intervals for the model parameters.

3.5 Some Special Cases of the BTIIHLTL-G Distribution

In this subsection, we present some special cases of the BTIIHLTL-G family of distributions. We considered the Uniform distribution, Exponential distribution and the Log-logistic distribution, defined in equations (1.12), (1.13) and (1.15) respectively.

3.5.1 Beta Type II Half Logistic Topp-Leone Uniform (BTIIHLTL-U) Distribution

The cdf and pdf of the BTIIHLTL-U distribution is given by

$$F_{BTIIHLTL-U}(x; \alpha, \beta, b, \theta) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[1 - \frac{1-(1-(\frac{x}{\theta})^2)^b}{1+(1-(\frac{x}{\theta})^2)^b} \right]^{\alpha+j}}{\alpha+j},$$

$$f_{BTIIHLTL-U}(x; \alpha, \beta, b, \theta) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-(1-(\frac{x}{\theta})^2)^b)}{1+(1-(\frac{x}{\theta})^2)^b} \right]^{\alpha+j-1}}{\alpha+j} \\ \times \frac{4b(\frac{1}{\theta})(1-\frac{x}{\theta})(1-(1-\frac{x}{\theta})^2)^{b-1}}{(1+(1-(1-\frac{x}{\theta})^2)^b)^2},$$

respectively, for $\alpha > 0, \beta > 0, b > 0, \theta > 0$.

The corresponding hrf of the BTIIHLTL-U is given by

$$h_{BTIIHLTL-U}(x; \alpha, \beta, b, \theta) = \frac{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{2(1-(1-(\frac{x}{\theta})^2)^b)}{1+(1-(\frac{x}{\theta})^2)^b} \right]^{\alpha+j-1}}{\alpha+j}}{1 - \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[1 - \frac{1-(1-(\frac{x}{\theta})^2)^b}{1+(1-(\frac{x}{\theta})^2)^b} \right]^{\alpha+j}}{\alpha+j}} \\ \times \frac{4b(\frac{1}{\theta})(1-\frac{x}{\theta})(1-(1-\frac{x}{\theta})^2)^{b-1}}{(1+(1-(1-\frac{x}{\theta})^2)^b)^2}$$

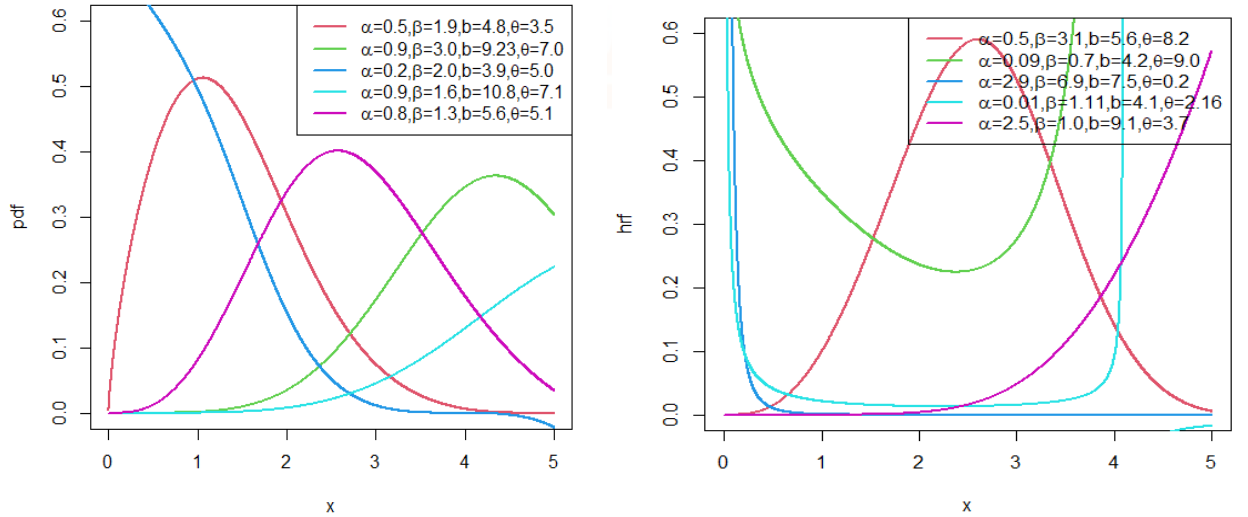


Figure 3.1: Pdf and hrf plots for the BTIIHLTL-U distribution

Figure 3.1 shows the plots of the pdfs and hrfs for the BTIIHLTL-U distribution. The pdf plots of the BTIIHLTL-U distribution display right skewed, left skewed, reverse-J and symmetric shapes. The hrf plots of the BTIIHLTL-U distribution display upside-down bathtub, decreasing,

increasing and bathtub shapes.

	$(\alpha, \beta, b, \theta)$				
u	(1.8,1.8,1.7,1.5)	(2,2.5,0.9,0.5)	(0.5,2.2,1.5,1.5)	(0.9,1,1.8,0.5)	(0.8,1.5,1.6,2.4)
0.1	0.1940938	0.003642693	0.01198728	0.001447779	0.1607597
0.2	0.2720600	0.008729444	0.03070851	0.007531866	0.2411097
0.3	0.3399938	0.015421040	0.05414864	0.020651307	0.3115831
0.4	0.4062106	0.024129198	0.08242812	0.044289198	0.3801674
0.5	0.4749931	0.035581890	0.11655994	0.083568973	0.4510891
0.6	0.5502803	0.051060267	0.15866621	0.146817734	0.5284210
0.7	0.6375523	0.072986480	0.21287857	0.248439979	0.6177821
0.8	0.7471480	0.106821657	0.28854362	0.416756789	0.7298685
0.9	0.9077093	0.169873204	0.41481247	0.722467469	0.8938137

Table 3.1: Random Generated Variables of the BTIIHLTL-U Distribution

Table 3.1 lists the generated random variables of the BTIIHLTL-U distribution via equation (3.3) for selected values of the parameters $(\alpha, \beta, b, \theta)$.

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the BTIIHLTL-U distribution are given in Table 3.2.

	$(\alpha, \beta, b, \theta)$				
	(0.8,1.3,1.9,1.4)	(0.9,0.5,1.6,1.5)	(0.5,2.0,1.5,1.5)	(0.9,0.9,1.8,0.5)	(0.8,1.5,1.6,1.4)
$E(X)$	0.32516604	0.33228252	0.182811755	0.18478192	0.29241699
$E(X^2)$	0.17562763	0.21376055	0.066799197	0.09394100	0.14090124
$E(X^3)$	0.11511271	0.15640165	0.033546920	0.06182631	0.08533554
$E(X^4)$	0.08401425	0.12300599	0.020250203	0.04578368	0.05887512
$E(X^5)$	0.06554967	0.10126387	0.013736796	0.03624926	0.04407257
$E(X^6)$	0.05348365	0.08601668	0.010079593	0.02995796	0.03484641
SD	0.04504689	0.07474491	0.007816507	0.02550632	0.02863720
CV	0.03884529	0.06607812	0.006311485	0.02219517	0.02421409
CS	0.03410881	0.05920903	0.005253913	0.01963840	0.02092323
CK	0.03038063	0.05363190	0.004477953	0.01760585	0.01838962

Table 3.2: Table of Moments for Selected Parameters of the BTIIHLTL-U Distribution

Figure 3.2 shows skewness and kurtosis plots for the BTIIHLTL-U distribution. When we fix the parameters β and θ , skewness and kurtosis decreases.

3.5.2 Beta Type II Half Log Logistic Topp-Leone Exponential (BTIIHLTL-E) Distribution

The cdf and pdf of the BTIIHLTL-E distribution is given by

$$F_{BTIIHLTL-E}(x; \alpha, \beta, b, \lambda) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[1 - \frac{1-(1-e^{-2\lambda x})^b}{1+(1-e^{-2\lambda x})^b} \right]^{\alpha+j}}{\alpha+j},$$

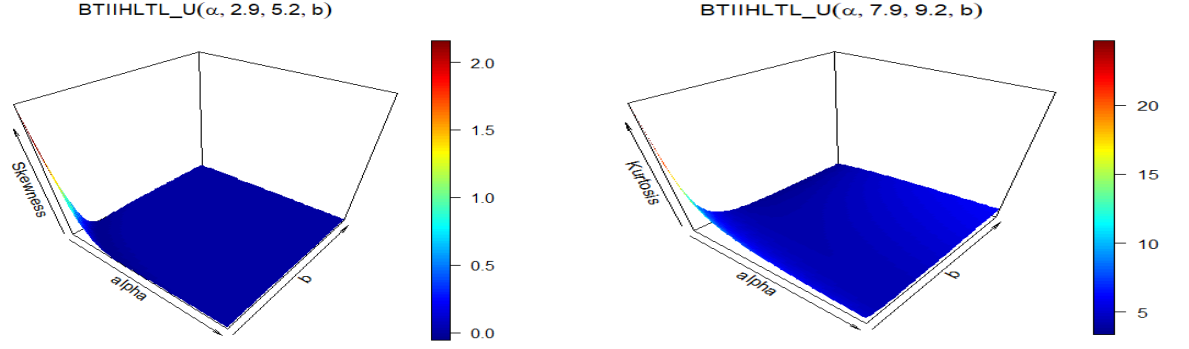


Figure 3.2: Skewness and Kurtosis plots for the BTIIHLTL-U distribution

$$f_{BTIIHLTL-E}(x; \alpha, \beta, b, \lambda) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-e^{-2\lambda x})^b}{1+(1-e^{-2\lambda x})^b} \right]^{\alpha+j-1} \times \frac{4b\lambda e^{-2\lambda x} (1-e^{-2\lambda x})^{b-1}}{(1+(1-e^{-2\lambda x})^b)^2},$$

respectively, for $\alpha > 0, \beta > 0, b > 0, \lambda > 0$. The corresponding hrf of the BTIIHLTL-E is given by

$$h_{BTIIHLTL-E}(x; \alpha, \beta, b, \lambda) = \frac{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-e^{-2\lambda x})^b}{1+(1-e^{-2\lambda x})^b} \right]^{\alpha+j-1}}{1 - \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[1 - \frac{1-(1-e^{-2\lambda x})^b}{1+(1-e^{-2\lambda x})^b} \right]^{\alpha+j}}{\alpha+j}} \times \frac{4b\lambda e^{-2\lambda x} (1-e^{-2\lambda x})^{b-1}}{(1+(1-e^{-2\lambda x})^b)^2}.$$

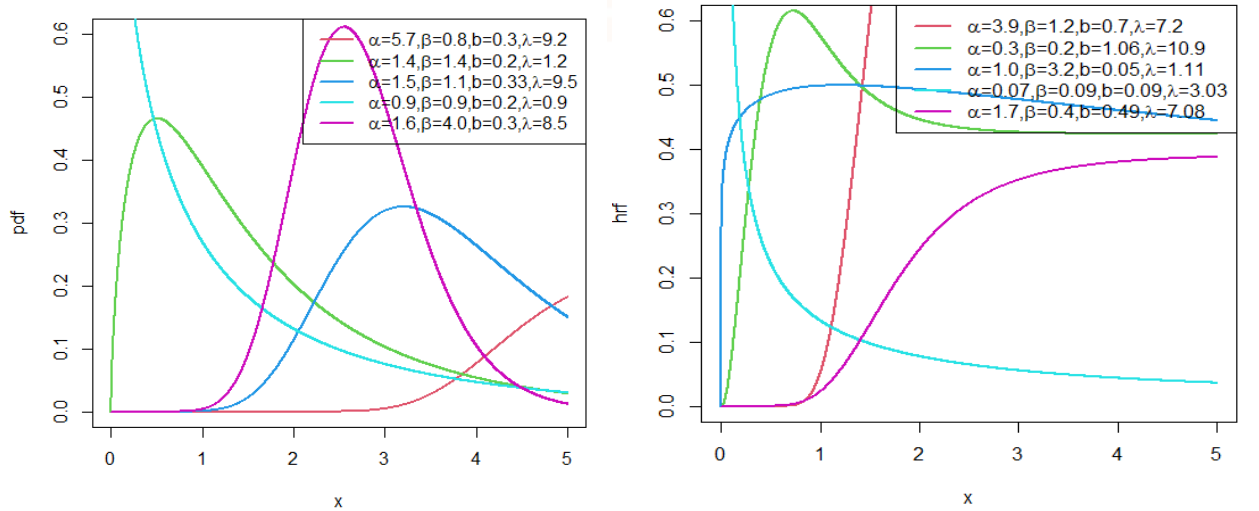


Figure 3.3: Pdf and hrf plots for the BTIIHLTL-E distribution

Figure 3.3 shows the plots of the pdfs and hrfs for the BTIIHLTL-E distribution. The pdf plot of the BTIIHLTL-E display reverse-J, almost symmetric, left and right skewed shapes. The hrf plots of the BTIIHLTL-E distribution display upside-down bathtub, upside-down bathtub

followed by bathtub, decreasing and increasing shapes

	$(\alpha, \beta, b, \lambda)$				
u	(1.8,1.8,1.7,1.5)	(2,2.5,0.9,0.5)	(0.5,2.2,1.5,1.5)	(0.9,1,1.8,0.5)	(0.8,1.5,1.6,2.4)
0.1	0.07130279	0.004506153	0.005354084	0.0004429065	0.06616809
0.2	0.10258411	0.010845751	0.013790122	0.0023273349	0.10207574
0.3	0.13125437	0.019194927	0.024512309	0.0064160972	0.13537579
0.4	0.16061943	0.030208574	0.037675535	0.0138409369	0.16957023
0.5	0.19274904	0.044821798	0.053946214	0.0264215149	0.20701575
0.6	0.23007009	0.064900109	0.074547710	0.0472682325	0.25054913
0.7	0.27650313	0.093958675	0.102053065	0.0825136754	0.30499309
0.8	0.34052729	0.140386014	0.142439528	0.1463319570	0.38069720
0.9	0.44908496	0.232444990	0.215806237	0.2850834792	0.51117672

Table 3.3: Generated Random Variables of the BTIIHLTL-E Distribution

Table 3.3 lists the generated random variables of the BTIIHLTL-E distribution via equation (3.3) for selected values of the parameters $(\alpha, \beta, b, \lambda)$.

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the BTIIHLTL-E distribution are given in Table 3.4.

	$(\alpha, \beta, b, \lambda)$				
	(1.8,0.8,0.7,0.5)	(2.0,2.5,0.9,0.5)	(0.5,2.0,1.5,1.5)	(0.9,5.0,1.8,0.5)	(0.8,4.5,1.6,0.4)
$E(X)$	0.094142387	0.025800682	0.31394007	0.40669569	0.36179626
$E(X^2)$	0.036229258	0.11703589	0.22275282	0.24164568	0.21145043
$E(X^3)$	0.020644669	0.06897486	0.17024100	0.16416981	0.14395931
$E(X^4)$	0.014024926	0.04704435	0.13676575	0.12158623	0.10737194
$E(X^5)$	0.010492751	0.03507085	0.11382735	0.09545645	0.08494762
$E(X^6)$	0.0083333871	0.02770969	0.09724353	0.07809028	0.06998540
SD	0.006890897	0.02279297	0.08475055	0.06583837	0.05936834
CV	0.005863727	0.01930428	0.07502941	0.05678907	0.05147778
CS	0.005097704	0.01671317	0.06726518	0.04986006	0.04539957
CK	0.004505674	0.01471911	0.06092980	0.04439905	0.04058225

Table 3.4: Table of Moments for Selected Parameters of the BTIIHLTL-E Distribution

Figure 3.4 shows skewness and kurtosis plots for the BTIIHLTL-E distribution. When we fix the parameters α and λ , skewness and kurtosis increases.

3.5.3 Beta Type II Half Logistic Topp-Leone Log Logistic (BTIIHLTL-LLoG) Distribution

The cdf and pdf of the BTIIHLTL-LLoG distribution is given by

$$F_{BTIIHLTL-LLoG}(x; \alpha, \beta, b, d) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[1 - \frac{1-(1+(1+x^d)^{-2})^b}{1+(1+(1+x^d)^{-2})^b} \right]^{\alpha+j}}{\alpha+j},$$

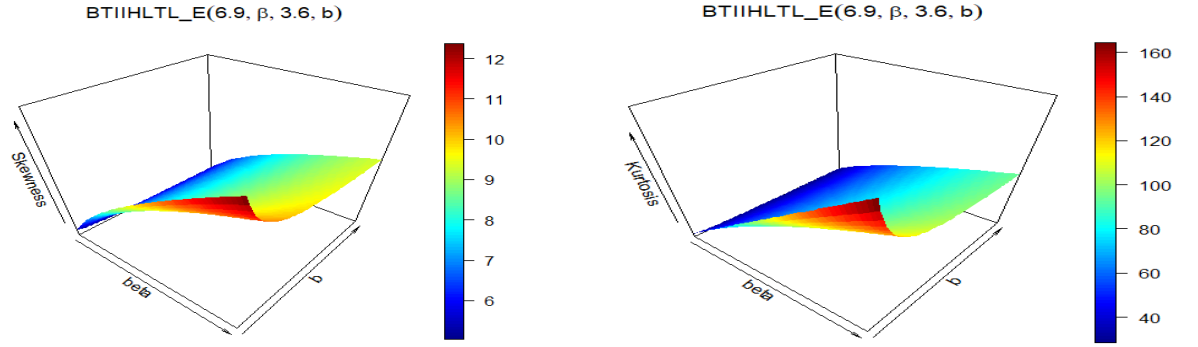


Figure 3.4: Skewness and Kurtosis plots for the BTIIHLTL-E distribution

$$f_{BTIIHLTL-LLoG}(x; \alpha, \beta, b, d) = \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-(1+x^d)^{-2})^b}{1+(1-(1+x^d)^{-2})^b} \right]^{\alpha+j-1} \\ \times \frac{4b(dx^{d-1}(1+x^d)^{-2}(1-(1+x^d)^{-2})^{b-1}}{(1+(1-(1+x^d)^{-2})^b)^2},$$

respectively, for $\alpha > 0, \beta > 0, b > 0, d > 0$.

The corresponding hrf of the BTIIHLTL-LLoG is given by

$$h_{BTIIHLTL-LLoG}(x; \alpha, \beta, b, d) = \frac{\frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \left[\frac{2(1-(1+x^d)^{-2})^b}{1+(1-(1+x^d)^{-2})^b} \right]^{\alpha+j-1}}{1 - \frac{1}{B(\alpha, \beta)} \sum_{j=0}^{\infty} (-1)^j \binom{\beta-1}{j} \frac{\left[\frac{1 - \frac{1-(1+x^d)^{-2})^b}{1+(1-(1+x^d)^{-2})^b} \right]^{\alpha+j}}{\alpha+j}} \\ \times \frac{4b(dx^{d-1}(1+x^d)^{-2}(1-(1+x^d)^{-2})^{b-1}}{(1+(1-(1+x^d)^{-2})^b)^2}.$$

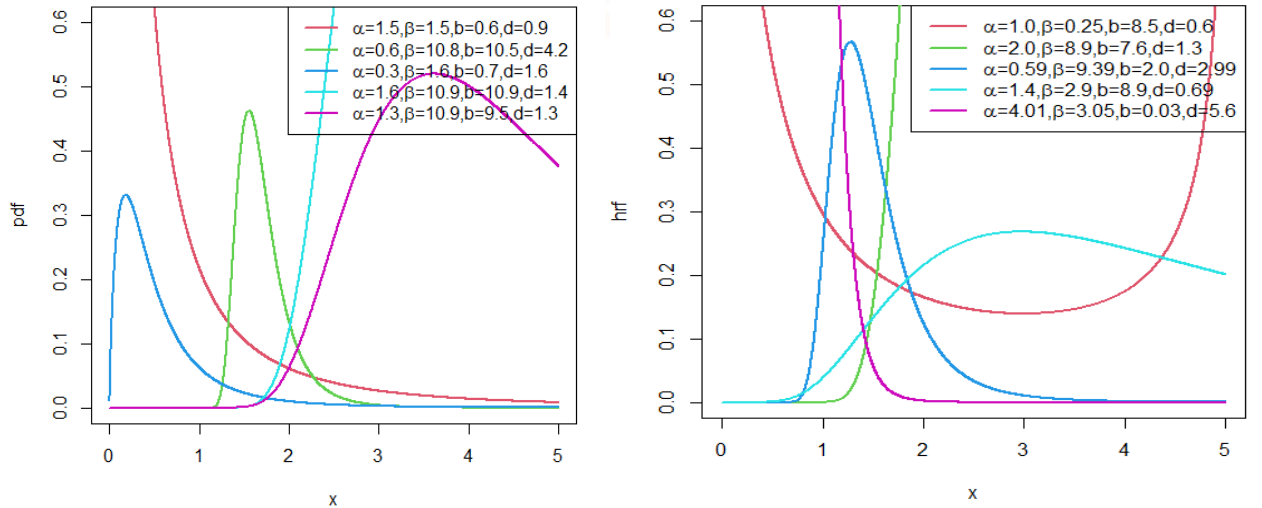


Figure 3.5: Pdf and hrf plots for the BTIIHLTL-LLoG distribution

Figure 3.3 shows the plots of the pdfs and hrfs for the BTIIHLTL-LLoG distribution. The pdf plots of the BTIIHLTL-LLoG display reverse-J, J, right and left skewed and almost symmetric shapes. The hrf plots of the BTIIHLTL-LLoG distribution display upside-down bathtub,

increasing, uni-modal and decreasing shapes.

	(α, β, b, d)				
u	(1.8,1.8,1.7,1.5)	(2,2.5,0.9,0.5)	(0.5,2.2,1.5,1.5)	(0.9,1,1.8,0.5)	(0.8,1.5,1.6,2.4)
0.1	0.2996124	0.002206409	0.04018443	0.01917186	0.2541090
0.2	0.3770865	0.005884757	0.07590124	0.04777125	0.3393035
0.3	0.4424283	0.011122556	0.11194341	0.08408435	0.4118255
0.4	0.5058851	0.018485990	0.15008750	0.12945999	0.4825542
0.5	0.5727518	0.028857862	0.19218434	0.18655147	0.5574834
0.6	0.6482657	0.044012226	0.24096820	0.26051129	0.6428386
0.7	0.7405156	0.067392387	0.30130733	0.36144188	0.7484103
0.8	0.8667021	0.067392387	0.38428076	0.51350277	0.8959568
0.9	1.0832458	0.197897692	0.52670132	0.80085926	1.1586386

Table 3.5: Generated Random Variables of the BTIIHLTL-LLoG Distribution

Table 3.5 lists the generated random variables of the BTIIHLTL-LLoG distribution via equation (3.3) for selected values of the parameters $(\alpha, \beta, b, \theta)$.

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the BTIIHLTL-LLoG distribution are given in Table 3.6.

	(α, β, b, d)				
	(0.8,1.3,1.9,1.4)	(0.9,0.5,1.6,1.5)	(0.5,2.0,1.5,1.5)	(0.9,0.9,1.8,0.5)	(0.8,1.5,1.6,1.4)
$E(X)$	0.32560294	0.28544904	0.13911402	0.20678215	0.29381119
$E(X^2)$	0.21110075	0.14749696	0.09400274	0.09691201	0.18738523
$E(X^3)$	0.15051297	0.09350922	0.06907105	0.05929557	0.13276219
$E(X^4)$	0.11468053	0.06671146	0.05382415	0.04166462	0.10100182
$E(X^5)$	0.09164163	0.05125312	0.04375580	0.03177111	0.08076282
$E(X^6)$	0.07584796	0.04137671	0.03670228	0.02554166	0.06694697
SD	0.06446547	0.03458934	0.03152704	0.02129638	0.05700488
CV	0.05592973	0.02966596	0.02758783	0.01823299	0.04954914
CS	0.04932056	0.02594416	0.02449924	0.01592527	0.04377124
CK	0.04406737	0.02303811	0.02201803	0.01412778	0.03917283

Table 3.6: Table of Moments for Selected Parameters of the BTIIHLTL-LLoG Distribution

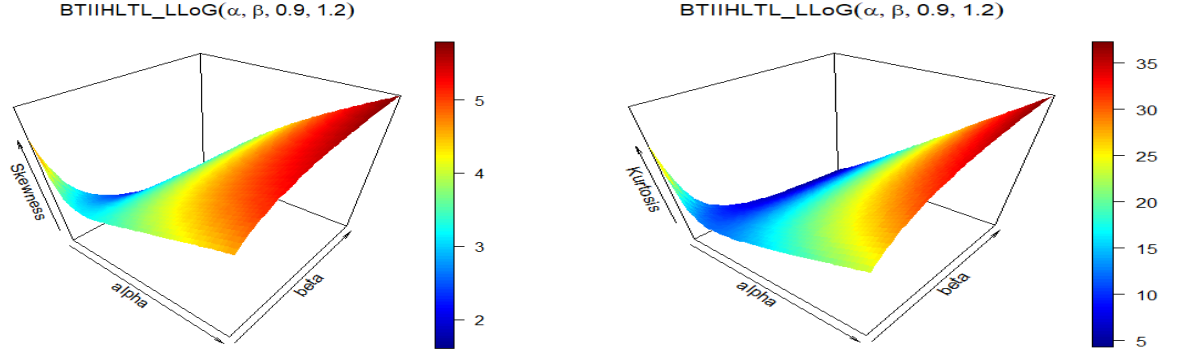


Figure 3.6: Skewness and Kurtosis plots for the BTIIHLTL-LLoG distribution

Figure 3.6 shows skewness and kurtosis plots for the BTIIHLTL-LLoG distribution. When we fix the parameters b and d, skewness and kurtosis increases.

3.6 Simulation Study

In this section, a simulation study was conducted to examine consistency of the maximum likelihood estimators. We considered simulations from the BTIIHLTL-G distribution for the sample sizes $n=80$, $n=100$, $n=200$, $n=400$ and $n=800$. The simulation results are shown in Table 3.7. The mean, root mean square error (RMSE), and average bias for the estimated parameter, say $\hat{\Delta}$ are calculated using the formulae

$$Bias(\hat{\Delta}) = \frac{1}{N} \sum_{i=1}^N (\hat{\Delta}_i - \Delta), \quad \text{and} \quad RMSE(\hat{\Delta}) = \sqrt{\frac{\sum_{i=1}^N (\hat{\Delta}_i - \Delta)^2}{N}},$$

respectively. If the model performs better, we expect the mean to approximate the true parameter values, the RMSE, and bias to decay toward zero for an increase in sample size.

From the results in Table 3.7, the mean figures indicate an approximation to the true parameter values with the RMSE and bias values decreasing towards zero for all the parameter values.

3.7 Applications

In this section, we consider the usefulness of our new model by applying it to two real life data sets. The BTIIHLTL-E is compared to the following models: Type II Exponentiated Half Logistic Weibull (TIIHLW) by Al-Mofleh et al. [8], Beta Weibull (BW) by Cordeiro et al. [32], Transmuted Topp- Leone Weibull (TTLW) by Ibrahim and Yousef [46], Beta Generalized Lindley (BGL) by Oluyede and Yang [77] and Beta Odd Lindley Uniform distribution (BOLU) by Chipepa et al. [31]. We performed assessment of the model using goodness-of-fit statistics which include; $-2 \log$ likelihood ($-2 \log L$), Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC), Corrected Akaike Information Criterion (CAIC), Kolmogorov-Sminorv (K-S) statistic and its p-value, the Anderson-Darling (A^*) and Cramer-von Mises (W^*) statistics. The best model is considered to be the one with the least value of these statistics with a bigger

	$\alpha = 1.6, \beta = 0.4, \lambda = 1.3, b = 1.9$			$\alpha = 1.6, \beta = 1.3, \lambda = 0.4, b = 1.9$			
sample size	Mean	RMSE	Bias	Mean	RMSE	Bias	
α	80	6.2642	14.9034	4.6642	3.1205	4.4745	1.5205
	100	6.3065	13.7567	4.7065	3.0604	4.4948	1.4604
	200	4.3409	8.4325	2.7409	2.5746	3.1714	0.9746
	400	2.8854	3.0586	1.2854	2.5493	2.8339	0.9493
	800	2.1721	1.2874	0.5721	2.0615	2.2837	0.4615
β	80	0.7517	1.2679	0.3517	3.0836	5.6123	1.7836
	100	0.7879	1.3518	0.3879	2.5842	4.6061	1.2842
	200	0.6917	1.0585	0.2917	2.0482	3.1184	0.74823
	400	0.5303	0.9229	0.1303	1.6049	1.8849	0.3049
	800	0.3715	0.1186	-0.0285	1.4446	1.0981	0.1446
λ	80	1.6398	1.1267	0.3398	0.7328	0.7728	0.3328
	100	1.6085	1.1020	0.3085	0.7670	0.7613	0.3670
	200	1.5089	0.8859	0.2089	0.6205	0.5107	0.2205
	400	1.6105	0.7591	0.3105	0.5599	0.3572	0.1599
	800	1.5511	0.5646	0.2512	0.4763	0.2247	0.0763
b	80	6.3285	14.1374	4.4285	7.6725	21.8641	5.7725
	100	4.1983	7.8330	2.2983	7.4690	19.4349	5.5690
	200	4.5436	8.9920	2.6436	5.7799	11.3671	3.8799
	400	3.2264	4.3624	1.3264	4.9347	8.5918	3.0347
	800	1.8894	0.6359	-0.0106	3.7049	5.1437	1.8049

Table 3.7: Monte Carlo Simulation Results for BTIIHLTL-E Distribution: Mean, RMSE and Average Bias

p-value. The AdequacyModel package was used to evaluate the above stated statistics. The pdfs of the non-nested distributions are given by

$$f_{TIIHLTLW}(a, \lambda, \delta, \gamma) = \frac{2a\lambda\delta\gamma(x^{\gamma-1})e^{(-\delta x^{\gamma})}(1 - e^{-\delta x^{\gamma}})^{\lambda-1}(1 - (1 - e^{-\delta x^{\gamma}})^{\lambda})^{a-1}}{(1 + (1 - e^{-\delta x^{\gamma}})^{\lambda})^{a+1}},$$

for $a, \lambda, \delta, \gamma > 0$.

$$f_{BW}(a, b, \gamma, \beta) = \frac{\beta\alpha^{\beta}}{\beta(a, b)}x^{\beta-1}e^{-b(\alpha x)^{\beta}}(1 - e^{(-\alpha x)^{\beta}})^{a-1},$$

for $a, b, \gamma, \beta > 0$.

$$f_{TTLW}(a, b, \gamma, \lambda) = 2\lambda b a^b x^{b-1}e^{-2(ax)^b}(1 - e^{-2(ax)^b})^{\alpha-1}(1 + \lambda - 2\lambda)(1 - e^{-2(ax)^b})^{\alpha},$$

for $a, b, \gamma, \lambda > 0$.

$$f_{BGL}(\alpha, \theta, a, b) = \alpha\theta^2(1+x)\frac{e^{-\theta x}}{\beta(a, b)}\left(\left(1 - \frac{(1 + \theta + \theta x)}{(1 + \theta)}\right)e^{-\theta x}\right)^{a(\alpha-1)}\left(1 - \left(\left(1 - \frac{(1 + \theta + \theta x)}{(1 + \theta)}\right)e^{-\theta x}\right)^{(\alpha)}\right)^{b-1},$$

for $\alpha, \theta, a, b > 0$.

$$f_{BOLU}(a, b, \lambda, \theta) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\left(1 - \frac{(\lambda + (1 - \frac{x}{\theta}))}{(1 + \lambda)(1 - \frac{x}{\theta})}e^{(-\lambda\frac{x}{\theta-x})}\right)^{a-1} \\ \times \left(\frac{\lambda + (1 + \frac{x}{\theta})}{(1 + \lambda)(1 - \frac{x}{\theta})}e^{-\lambda\frac{x}{\theta-x}}\right)^{b-1}\frac{\lambda^2}{(1 + \lambda)}\frac{\theta^2}{(\theta - x)^3}e^{-\lambda\frac{x}{\theta-x}},$$

for $a, b, \lambda, \theta > 0$.

3.7.1 Yarn at 2.3% Strain Level Data Set

The first data set considered, presented in Table 3.8, was taken from Silva and Cordeiro [89] where an experiment was carried out to test the breaking strength of yarn at 2.3% strain level.

86	146	251	653	98	249	400	292	131	169	175	176	76	264	15	364	195	262	88
264	157	220	42	321	180	198	38	20	61	121	282	224	149	180	325	250	196	90
229	166	38	337	65	151	341	40	40	135	597	246	211	180	93	315	353	571	124
279	81	186	497	182	423	185	229	400	338	290	398	71	246	185	188	568	55	55
61	244	20	284	393	396	203	829	239	236	286	194	277	143	198	264	105	203	
124	137	135	350	193	188													

Table 3.8: Yarn at 2.3% Strain Level Data

Model	Estimates				Statistics							
	α	β	λ	b	$-2\log L$	AIC	$CAIC$	BIC	W^*	A^*	KS	$P - value$
BTIIHLTL-E	0.3016 (0.0540)	1.9098 (0.8024)	0.0020 (0.0004)	6.2872 (0.1512)	1248.6	1256.6	1257.0	1267.0	0.0785	0.4421	0.0800	0.5440
TIIHLW	a	λ	δ	γ	1249.8	1257.8	1258.2	1268.2	0.0991	0.5538	0.0837	0.4861
	1.8361×10^3 3.4250×10^{-7}	2.0530×10^1 9.2446×10^{-4}	5.0765×10^{-1} 3.1303×10^{-2}	1.4238×10^{-1} 1.0735×10^{-2}								
BW	a	b	α	β	1249.8	1257.8	1258.2	1268.2	0.0959	0.5397	0.0823	0.5070
	1.3623 (0.5727)	0.9761 (0.0831)	0.0049 (0.0016)	1.3643 (0.3115)								
TTLW	a	b	α	λ	1249.8	1257.8	1258.2	1268.2	0.0906	0.5189	0.1900	0.0015
	0.0029 (-)	1.3625 (0.1416)	1.3645 (0.5031)	0 (-)								
BGL	α	θ	a	c	1717.7	1725.9	1726.3	1736.3	0.4053	2.2130	0.4996	2.2×10^{-16}
	1.0049×10^{-1} (4.1620×10^{-3})	5.3027×10^{-9} (5.5710×10^{-7})	3.0009×10^{-4} (2.1014×10^{-4})	1.0100×10^1 (1.2597×10^{-5})								

Table 3.9: Parameter estimates and goodness-of-fit statistics for various models fitted for Yarn at 2.3% Level Strain Data

The estimated variance-covariance matrix for the BTIIHLTL-E model on Yarn at 2.3% Level Strain data.

$$\begin{bmatrix} 2.9200 \times 10^{-3} & -0.0269 & 1.8998 \times 10^{-5} & -4.9983 \times 10^{-3} \\ -2.6919 \times 10^{-2} & 0.6439 & -3.1598 \times 10^{-4} & 1.2137 \times 10^{-1} \\ 1.8998 \times 10^{-5} & -0.0003 & 1.9095 \times 10^{-7} & -5.9316 \times 10^{-5} \\ -4.9983 \times 10^{-3} & 0.1213 & -5.9316 \times 10^{-5} & 2.2882 \times 10^{-2} \end{bmatrix}$$

with the confidence intervals (95%) for the model parameters given by

$$\alpha \in [0.3016 \pm 0.1059], \beta \in [1.9098 \pm 1.5728], \lambda \in [0.0020 \pm 0.0008], b \in [6.2872 \times 10^1 \pm 0.2964].$$

From the results shown in Table 3.9, we determine that the BTIIHLTL-E model outperforms the other models as it has the least value of goodness of fit statistics with a larger value of the K-S p-value. Also by observing plots of the fitted densities and probability in Figure 3.7, we arrive at the same conclusion.

3.7.2 Monthly Actual Taxes Revenue Data Set

The second data set which represents monthly actual taxes revenue (in 1000 million Egyptian pounds) in Egypt between January 2006 and November 2010 was taken from Mead [67]. The data set is given in Table 3.10.

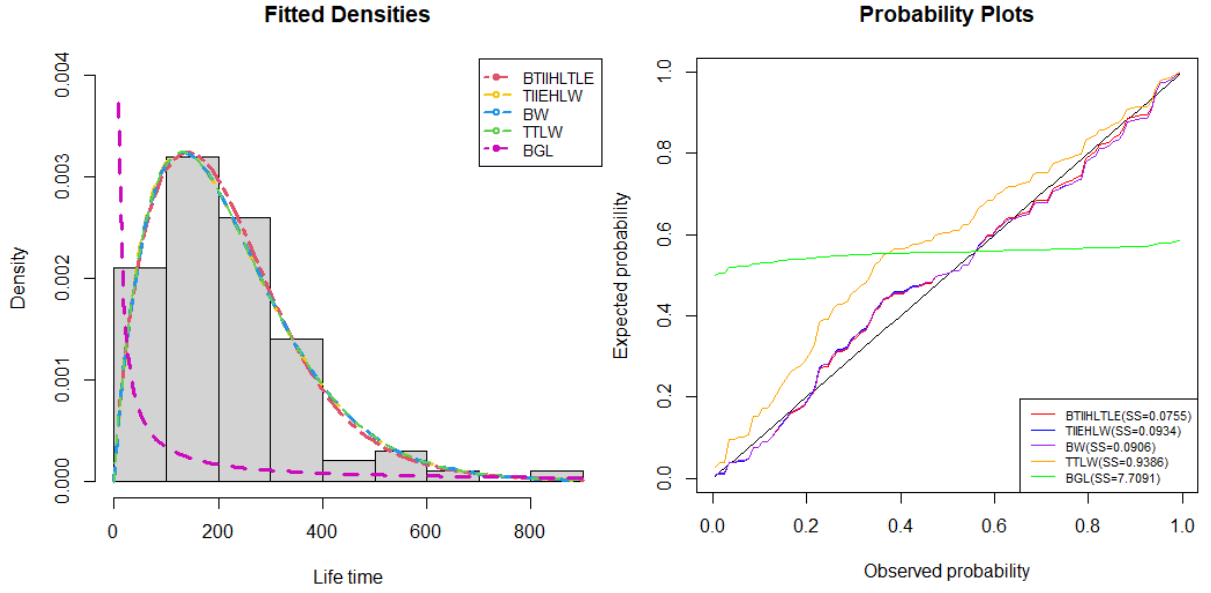


Figure 3.7: Fitted Densities and Probability Plots for the Yarn Data

5.9	20.4	14.9	16.2	17.2	7.8	6.1	9.2	10.2	9.6
13.3	8.5	21.6	18.5	5.1	6.7	17	8.6	9.7	39.2
35.7	15.7	9.7	10	4.1	36	8.5	8	9.2	26.2
16.7	21.3	35.4	14.3	8.5	10.6	19.1	20.5	7.1	
7.7	18.1	16.5	11.9	7	8.6	12.5	10.3	11.2	6.1
8.4	11	11.6	11.9	5.2	6.8	8.9	7.1	10.8	

Table 3.10: Monthly actual taxes revenue

Model	Estimates				Statistics							
	α	β	λ	b	$-2 \log L$	AIC	CAIC	BIC	W^*	A^*	KS	P-value
BTIHLTL-E	2.8040×10^{-1} (1.6033×10^{-1})	2.9689×10^{-1} (7.4067×10^{-2})	1.9223×10^{-1} (5.0620×10^{-2})	7.6389×10^1 (3.5892×10^{-2})	374.9	382.9	383.7	391.3	0.0260	0.1916	0.0550	0.9941
TIEHLW	a 0.3626 (0.2741)	λ 27.4562 (32.9144)	δ 0.4956 (0.4548)	γ 0.9159 (0.3553)	375.6	383.7	384.5	391.9	0.0385	0.2500	0.0654	0.9625
BW	a 90.7890 (0.0042)	b 0.2360 (0.2298)	α 1.0548 (0.5718)	β 0.8346 (0.3353)	375.7	383.7	384.5	392.0	0.0378	0.2463	0.0612	0.9801
TTLW	a 8.7261×10^{00} (4.0828×10^{00})	b 2.9503×10^{-1} (3.7468×10^{-2})	α 1.6253×10^3 (2.1944×10^{-2})	λ 7.7947×10^{-2} (6.2240×10^{-1})	376.5	384.5	385.2	392.8	-	-	1.0745	$< 2.2 \times 10^{-16}$
BOLU	a 3.8800 (6.9603×10^{-1})	b 3.4012 (6.0467×10^{-1})	λ 3.2368×10^5 (6.8236×10^{-1})	θ 5.1664×10^6 (4.2752×10^{-7})	385.5	393.5	394.2	401.8	0.1862	1.1507	0.1321	0.2547

Table 3.11: Parameter estimates and goodness-of-fit statistics for various models fitted for Monthly Actual Taxes Revenue Data

The estimated variance-covariance matrix for the BTIHLTL-E model on monthly actual taxes revenue data is given by

$$\begin{bmatrix} 2.5705 \times 10^{-2} & -7.6398 \times 10^{-3} & 7.7850 \times 10^{-3} & 5.6948 \times 10^{-5} \\ -7.6398 \times 10^{-3} & 5.4858 \times 10^{-3} & -2.8187 \times 10^{-3} & -1.4001 \times 10^{-5} \\ 7.7850 \times 10^{-3} & -2.8187 \times 10^{-3} & 2.5623 \times 10^{-3} & 1.6754 \times 10^{-5} \\ 5.6948 \times 10^{-5} & -1.4001 \times 10^{-5} & 1.6754 \times 10^{-5} & 1.2883 \times 10^{-7} \end{bmatrix}$$

and the confidence intervals (95%) for the model parameters are given by

$$\alpha \in [2.8040 \times 10^{-1} \pm 0.3142], \beta \in [2.9689 \times 10^{-1} \pm 0.1451], \lambda \in [1.9223 \times 10^{-1} \pm 0.0992], b \in [7.6388 \times 10^1 \pm 0.0007].$$

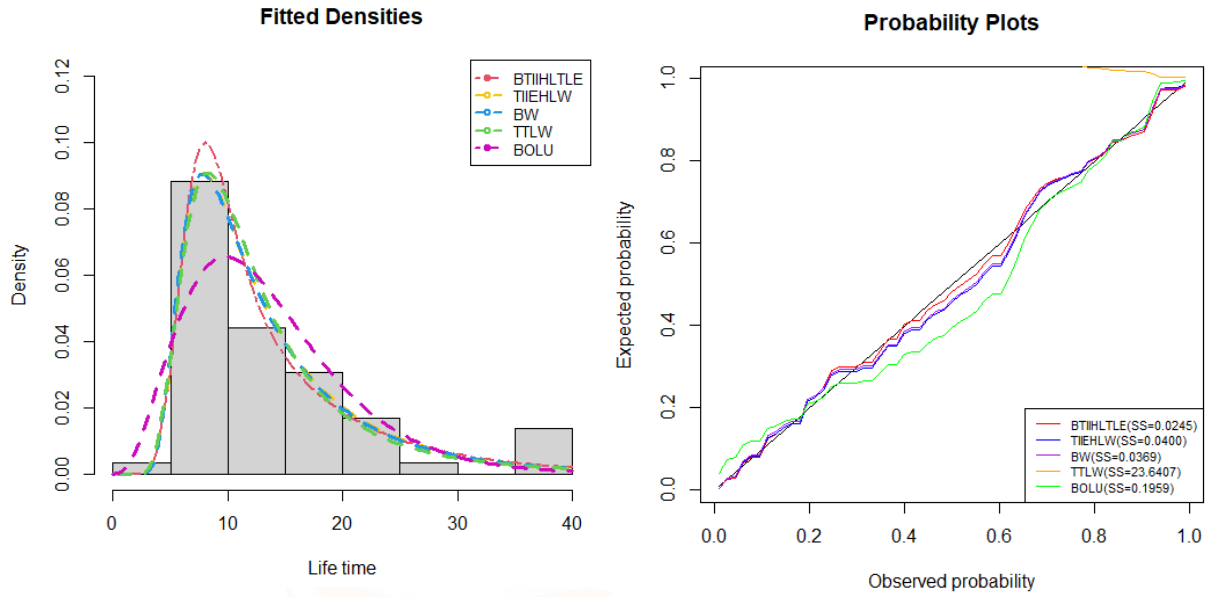


Figure 3.8: Fitted Densities and Probability Plots for the monthly actual taxes revenue Data

From the results shown in Table 3.11, we observe that the BTIIHLTL-E model outperforms the other models as it has the least value of goodness of fit statistics with a larger value of the K-S p-value. Also by observing plots of the fitted densities and probability in Figure 3.8, we arrive at the same conclusion.

3.8 Concluding Remarks

A new family of distributions, namely, the Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) was developed. Special cases and structural properties of the new distribution are presented. A simulation study was conducted to assess the consistency of the maximum likelihood estimates. Two real data examples were presented to illustrate the usefulness of the new distribution. The new model performs better than several existing models that were considered.

4. Type II Half Logistic Topp-Leone Exponentiated Generalized-G Distribution

4.1 Introduction

This chapter consists of the extension of the TIIHLTL-G distribution defined in equation (1.6) using the exponentiated generalized-G method by Cordeiro et al. [36] given in equation (1.20). The general family is called the Type II Half Logistic Topp-Leone Exponentiated Generalized G (TIIHLTLEG-G). Some statistical properties of the new distribution including cdf, pdf, hazard function, reverse hazard function, moments, mean deviation, median deviation, Bonferroni curve and Lorenz curve are established. The pdf of the new distribution is also presented as an infinite linear combination of the exponentiated-G (Exp-G) distribution (see Barreto-Souza and Simas [18]). Hence statistical properties of the new distribution are derived directly from those of the Exp-G distribution. Some special models of the new distribution are presented. The pdf function of the new distribution shows left skewed, right skewed and symmetrical shapes while the hazard rate function shows increasing, decreasing, bathtub and upside-down bathtub shapes. The maximum likelihood method is used to estimate model parameters. Simulation results are presented to assess the performance of the maximum likelihood estimators. Finally, an application of the new model to two real data is conducted.

4.2 Type II Half Logistic Topp-Leone Exponentiated Generalized-G (TIIHLTLEG-G) Family of Distributions

From equation (1.20) and (1.21), the cdf and pdf of the TIIHLTLEG-G distribution is given by

$$F_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \frac{2(1 - (1 - [1 - \bar{G}^\alpha(x; \xi)]^\beta)^2)^b}{1 + (1 - (1 - [1 - \bar{G}^\alpha(x; \xi)]^\beta)^2)^b}, \quad (4.1)$$

and

$$f_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \frac{4bg(x)\alpha\beta\bar{G}^{\alpha-1}[1 - \bar{G}^\alpha]^\beta[1 - [1 - \bar{G}^\alpha]^\beta][1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2]^{b-1}}{[1 + (1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2)^b]^2}, \quad (4.2)$$

respectively, for $\alpha > 0$, $\beta > 0$, $b > 0$ and ξ is a vector of parameters. The survival function, hrf and rhrf of the TIIHLTLEG-G distribution are given by

$$S_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \frac{1 - (1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2)^b}{1 + (1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2)^b},$$

$$h_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \frac{4bg(x)\alpha\beta\bar{G}^{\alpha-1}[1-\bar{G}^\alpha]^\beta[1-[1-\bar{G}^\alpha]^\beta]}{[1+(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b]^2 \times (1-(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b)} \\ \times [1-(1-[1-\bar{G}^\alpha]^\beta)^2]^{b-1}(1+(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b),$$

$$\tau_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \frac{4bg(x)\alpha\beta\bar{G}^{\alpha-1}[1-\bar{G}^\alpha]^\beta[1-[1-\bar{G}^\alpha]^\beta]}{[1+(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b]^2 \times (1-(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b)} \\ \times [1-(1-[1-\bar{G}^\alpha]^\beta)^2]^{b-1}(2(1-(1-[1-\bar{G}^\alpha]^\beta)^2)^b).$$

respectively.

4.2.1 Quantile Function

The quantile function is obtained by inverting $F_{TIIHLTLEG-G}(x) = u$. We solve

$$\left[\frac{2(1-(1-[1-\bar{G}^\alpha(x; \xi)]^\beta)^2)^b}{[1+(1-(1-[1-\bar{G}^\alpha(x; \xi)]^\beta)^2)^b]} \right] = u$$

$$G(x; \xi) = 1 - \sqrt[\alpha]{1 - \sqrt[\beta]{1 - \sqrt[1-\beta]{1 - \sqrt[b]{\frac{u}{2u}}}}}}$$

Consequently,

$$x_u = G^{-1} \left(1 - \sqrt[\alpha]{1 - \sqrt[\beta]{1 - \sqrt[1-\beta]{1 - \sqrt[b]{\frac{u}{2u}}}}} \right). \quad (4.3)$$

Thus random TIIHLTLEG-G variables can be generated based on equation 4.3.

Corollary 0.7. *The pdf of the TIIHLTLEG-G given in equation (4.2) can be expressed as follows*

$$f_{TIIHLTLEG-G}(x; \alpha, \beta, b, \xi) = \sum_{i,j,k,l,m=0}^{\infty} \frac{(-1)^{j+k+l+m} \alpha \beta 4b}{m+1} \\ \times \binom{i-1}{i} \binom{b(i+1)-1}{j} \binom{2j+1}{k} \binom{\beta(k+1)-1}{l} \binom{\alpha l + \alpha - 1}{m} \\ \times (m+1)g(x; \xi)G^m(x; \xi) \\ = \sum_{m=0}^{\infty} \eta_m g_m(x; \xi),$$

where,

$$g_m(x; \xi) = (m+1)g(x; \xi)G^m(x; \xi). \quad (4.4)$$

is an Exponentiated-G(Exp-G) distribution with power parameter $m+1$ and

$$\eta_m = \sum_{i,j,k,l,m=0}^{\infty} \frac{(-1)^{j+k+l+m} \alpha \beta 4b}{m+1} \\ \times \binom{i-1}{i} \binom{b(i+1)-1}{j} \binom{2j+1}{k} \binom{\beta(k+1)-1}{l} \binom{\alpha l + \alpha - 1}{m}. \quad (4.5)$$

Proof. Using binomial expansions,

$$\begin{aligned}
[1 + (1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2)^b]^{-2} &= \sum_{i=0}^{\infty} \binom{i-1}{i} (1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2)^{bi}, \\
[(1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2)^{bi+b-1}] &= \sum_{j=0}^{\infty} (-1)^j \binom{b(i+1)-1}{j} (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^{2j}, \\
[1 - (1 - \bar{G}^\alpha(x; \xi))^\beta]^{2j+1} &= \sum_{k=0}^{\infty} (-1)^k \binom{2j+1}{k} (1 - \bar{G}^\alpha(x; \xi))^{\beta k}, \\
[1 - \bar{G}^\alpha(x; \xi)]^{\beta k + \beta - 1} &= \sum_{l=0}^{\infty} (-1)^l \binom{\beta(k+1)-1}{l} \bar{G}^{\alpha l}(x; \xi), \\
[\bar{G}(x; \xi)]^{\alpha l + \alpha - 1} &= \sum_{m=0}^{\infty} (-1)^m \binom{\alpha l + \alpha - 1}{m} G^m(x; \xi),
\end{aligned}$$

we get,

$$\begin{aligned}
f_{TIIHLEGTLEG-G}(x; \alpha, \beta, b, \xi) &= \sum_{i,j,k,l,m=0}^{\infty} \frac{(-1)^{j+k+l+m} \alpha \beta 4b}{m+1} \\
&\times \binom{i-1}{i} \binom{b(i+1)-1}{j} \binom{2j+1}{k} \binom{\beta(k+1)-1}{l} \binom{\alpha l + \alpha - 1}{m} \\
&\times (m+1) g(x; \xi) G^m(x; \xi) \\
&= \sum_{m=0}^{\infty} \eta_m g_m(x; \xi).
\end{aligned} \tag{4.6}$$

□

Therefore, the TIIHLEGTLEG-G pdf can be expressed as a linear combination (infinite) of the Exp-G distributions. We can then derive the mathematical and statistical characterizations of the TIIHLEGTLEG-G distribution directly from those of the Exp-G distribution. More information regarding statistical properties of the Exp-G distribution can be found in Barreto-Souza and Simas [18] and Aryal and Yousof [13].

Corollary 0.8. *The pdf of the i^{th} order statistic of TIIHLEGTLEG-G is given by*

$$\begin{aligned}
f_{i:n}(x) &= \frac{n!}{(i-1)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \left[\frac{2(1 - (1 - [1 - \bar{G}^\alpha(x; \xi)]^\beta)^2)^b}{1 + (1 - (1 - [1 - \bar{G}^\alpha(x; \xi)]^\beta)^2)^b} \right]^{p+i-1} \\
&\times \frac{4bg(x) \alpha \beta \bar{G}^{\alpha-1} [1 - \bar{G}^\alpha]^\beta [1 - [1 - \bar{G}^\alpha]^\beta] [1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2]^{b-1}}{[1 + (1 - (1 - [1 - \bar{G}^\alpha]^\beta)^2)^b]^2} \\
&= \sum_{t=0}^{\infty} \eta_t g_t(x; \xi),
\end{aligned}$$

where,

$$g_t(x; \xi) = (t+1)g(x; \xi)G^t(x; \xi),$$

is an Exponentiated-G(Exp-G) distribution with power parameter $t+1$ and

$$\eta_t = \frac{n!}{(i-i)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \sum_{n,p,q,r,s,t=0}^{\infty} \frac{2(-1)^{p+q+r+s+t} 4b\alpha\beta}{t+1} \\ \times \binom{i+n+p}{n} \binom{b(i+n+p)-1}{q} \binom{2q+1}{r} \binom{\alpha s + \alpha - 1}{t} \binom{\beta r + \beta}{s}.$$

Proof. Using binomial expansions,

$$[1 + (1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2)^b]^{-(2+p+i-1)} = \sum_{n=0}^{\infty} \binom{i+n+p}{n} (1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2)^{bn},$$

$$[1 - (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^2]^{b(i+n+p)-1} = \sum_{q=0}^{\infty} (-1)^q \binom{b(i+n+p)-1}{q} (1 - (1 - \bar{G}^\alpha(x; \xi))^\beta)^{2q},$$

$$[1 - (1 - \bar{G}^\alpha(x; \xi))^\beta]^{2q+1} = \sum_{r=0}^{\infty} (-1)^r \binom{2q+1}{r} (1 - \bar{G}^\alpha(x; \xi))^{\beta r},$$

$$[1 - \bar{G}^\alpha(x; \xi)]^{\beta r + \beta} = \sum_{s=0}^{\infty} (-1)^s \binom{\beta r + \beta}{s} \bar{G}^{\alpha s}(x; \xi),$$

$$[\bar{G}(x; \xi)]^{\alpha s + \alpha - 1} = \sum_{t=0}^{\infty} (-1)^t \binom{\alpha s + \alpha - 1}{t} G^t(x; \xi),$$

□

yields,

$$f_{i:n}(x) = \frac{n!}{(i-1)!(n-i)!} \sum_{p=0}^{n-i} (-1)^p \binom{n-i}{p} \sum_{n,p,q,r,s,t=0}^{\infty} \frac{2(-1)^{p+q+r+s+t} 4b\alpha\beta}{t+1} \\ \times \binom{i+n+p}{n} \binom{b(i+n+p)-1}{q} \binom{2q+1}{r} \binom{\alpha s + \alpha - 1}{t} \binom{\beta r + \beta}{s} \\ \times (t+1)g(x; \xi)G^t(x; \xi).$$

Corollary 0.9. The Rényi Entropy of the TIIHLEGTLEG-G is given by

$$I_{R(v)} = \frac{1}{1-v} \log \left(\int_0^\infty (f_{TIIHLEGTLEG-G}(x; \alpha, \beta, b, \xi))^v dx \right) \\ = \frac{1}{1-v} \log \left(\sum_{z=0}^{\infty} s_z^* e^{(1-v)I_{REG}} \right),$$

where,

$$I_{REG} = \frac{1}{1-v} \log \left(\int_0^\infty \left(\frac{1}{[\frac{z}{v} + 1]^v} g(x; \xi) G^{\frac{z}{v}}(x; \xi) \right)^v dx \right),$$

is the Rényi entropy of Exp-G distribution with $\frac{1}{[\frac{z}{v}+1]^v}$ power parameter and

$$s_z^* = (4b\alpha\beta)^v \sum_{u,v,w,y,z=0}^{\infty} (-1)^{v+w+y+z} \binom{2v+u+1}{u} \binom{bu+bv-v}{v} \binom{2v+v}{w} \\ \times \binom{\beta w + \beta v}{y} \binom{\alpha y + \alpha v - v}{z}.$$

Proof. Note that,

$$f^v = \frac{(4bg(x;\xi)\alpha\beta)^v \bar{G}^{\alpha v-v}(x;\xi) (1 - \bar{G}^\alpha)^{\beta v}(x;\xi) (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^v (1 - (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^2)^{bv-v}}{(1 + (1 - (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^2)^b)^{2v}},$$

Using binomial expansions,

$$[1 + (1 - (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^2)^b]^{-(2v)} = \sum_{u=0}^{\infty} \binom{2v+u+1}{u} (1 - (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^2)^{bu},$$

$$[(1 - (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^2)^{bu+bv-v}] = \sum_{v=0}^{\infty} (-1)^v \binom{bu+bv-v}{v} (1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^{2v},$$

$$[(1 - (1 - \bar{G}^\alpha(x;\xi))^\beta)^{2v+v}] = \sum_{w=0}^{\infty} (-1)^w \binom{2v+v}{w} (1 - \bar{G}^\alpha(x;\xi))^{\beta w},$$

$$[1 - \bar{G}(x;\xi)]^{\beta w + \beta v} = \sum_{y=0}^{\infty} (-1)^y \binom{\beta w + \beta v}{y} \bar{G}^{\alpha y}(x;\xi),$$

$$[\bar{G}(x;\xi)]^{\alpha y + \alpha v - v} = \sum_{z=0}^{\infty} (-1)^z \binom{\alpha y + \alpha v - v}{z} G^z(x;\xi),$$

□

Therefore,

$$f^v(x; \alpha, \beta, b, \xi) = (4bg(x;\xi)\alpha\beta)^v \sum_{u,v,w,y,z=0}^{\infty} (-1)^{v+w+y+z} \binom{2v+u+1}{u} \binom{bu+bv-v}{v} \binom{2v+v}{w} \\ \times \binom{\beta w + \beta v}{y} \binom{\alpha y + \alpha v - v}{z} G^z(x;\xi),$$

Substituting in the above equation we have,

$$I_{R(v)} = \frac{1}{1-v} \log \left(\int_0^\infty \left[(4bg(x;\xi)\alpha\beta)^v \sum_{u,v,w,y,z=0}^{\infty} (-1)^{v+w+y+z} \binom{2v+u+1}{u} \binom{bu+bv-v}{v} \binom{2v+v}{w} \right. \right. \\ \left. \left. \times \binom{\beta w + \beta v}{y} \binom{\alpha y + \alpha v - v}{z} G^z(x;\xi) \right] dx \right). \\ = \frac{1}{1-v} \log \left(\sum_{z=0}^{\infty} s_z^* e^{(1-v)I_{REG}} \right)$$

4.3 Some Statistical Properties of the TIIHLEGTLEG-G Family of Distributions

In this section, we present some of the properties of the TIIHLEGTLEG-G family of distributions.

4.3.1 Moments, Conditional Moments, and Generating Functions

If X follows the TIIHLEGTLEG-G distribution and $Y \sim \text{Exp-G}(m)$ as in equation (4.4), then the r^{th} moment of the TIIHLEGTLEG-G family of distributions can be obtained as

$$\begin{aligned} E(X^r) &= \int_0^\infty x^r \cdot f_{TIIHLEGTLEG-G}(x; \alpha, \beta, b, \xi) dx \\ &= \int_0^\infty x^r \cdot \eta_m g_m(x; \xi) dx, \end{aligned}$$

where $g_m(x; \xi)$ is an Exp-G distribution with power parameter $m+1$ given in equation (4.4) and η_m is given in equation (4.5). The moment generating function is given by

$$M_x(t) = \sum_{m=0}^{\infty} \eta_m M_Y(t),$$

where $M_Y(t)$ is the moment generating function of the Exp-G distribution with power parameter $m+1$ and η_m is given in equation (4.5).

The conditional r^{th} moment of the TIIHLEGTLEG-G distribution is

$$\begin{aligned} E(X^r | X > n) &= \frac{1}{\bar{F}(n; \alpha, \beta, b, \xi)} \int_n^\infty x^r \cdot f_{TIIHLEGTLEG-G}(x; \alpha, \beta, b, \xi) dx \\ &= \frac{1}{\bar{F}(n; \alpha, \beta, b, \xi)} \sum_{m=0}^{\infty} \eta_m \cdot E(Y^r I_{(Y^r > n)}), \end{aligned}$$

where $E(Y^r I_{(Y^r > n)}) = \int_n^\infty y^r \cdot g_m(y; \xi) dy$.

The average lifetime of X is given by

$$E(X) = \sum_{m=0}^{\infty} \eta_m E(Y).$$

where,

$$E(Y) = (m+1) \int_0^\infty x \cdot g(x; \xi) G^m(x; \xi) dx$$

4.3.2 Mean Deviations

Using equations (1.32) and (1.33), $D(\mu)$ and $D(M)$ of the TIIHLEGTLEG-G are given by

$$D(\mu) = \int_0^\infty |x - \mu| f_{TIIHLEGTLEG-G}(x; \alpha, \beta, b, \xi) dx = 2\mu F_{TIIHLEGTLEG-G}(\mu) - 2\mu + 2T(\mu),$$

and

$$D(M) = \int_0^\infty |x - M| f_{TIIHLEGTLE-G}(x; \alpha, \beta, b, \xi) dx = -\mu + 2T(\mu),$$

respectively. Note that,

$$T(\mu) = \int_\mu^\infty x \cdot f(x; \alpha, \beta, b, \xi) dx = \sum_{m=0}^\infty \eta_m \int_\mu^\infty y \cdot g_m(x; \xi) dy,$$

and

$$T(M) = \int_M^\infty x \cdot f(x; \alpha, \beta, b, \xi) dx = \sum_{m=0}^\infty \eta_m \int_M^\infty y \cdot g_m(x; \xi) dy,$$

where $g_m(x; \xi)$ is given in equation (4.4) and η_m is given by equation (4.5), μ and M are the mean and median respectively of $F_{TIIHLEGTLE-G}(x; \alpha, \beta, b, \xi)$ given in equation (4.1). Therefore,

$$D(\mu) = 2\mu F_{TIIHLEGTLE-G}(\mu) - 2\mu + 2 \sum_{m=0}^\infty \eta_m \int_\mu^\infty y \cdot g_m(x; \xi) dy,$$

and

$$D(M) = -\mu + 2 \sum_{m=0}^\infty \eta_m \int_M^\infty y \cdot g_m(x; \xi) dy.$$

4.3.3 Lorenz and Bonferroni Curves

For $0 < p < 1$ Bonferroni and Lorenz curves are given respectively by

$$\begin{aligned} B(p) &= \frac{1}{p\mu} \int_0^q x \cdot f_{TIIHLEGTLE-G}(x) dx \\ &= \frac{1}{p\mu} \sum_{m=0}^\infty \eta_m \int_0^q x \cdot g_m(x; \xi) dx, \end{aligned}$$

$$\begin{aligned} L(p) &= \frac{1}{\mu} \int_0^q x \cdot f_{TIIHLEGTLE-G}(x) dx \\ &= \frac{1}{\mu} \sum_{m=0}^\infty \eta_m \int_0^q x \cdot g_m(x; \xi) dx, \end{aligned}$$

respectively, where $\int_0^q x \cdot g_m(x; \xi) dx$ is the equation of the first moment of the Exp-G distribution with parameter $g_m(x; \xi)$ given in equation (4.4) and η_m given in equation (4.5), with $q = F^{-1}(p)$.

4.4 Estimation

The maximum likelihood estimation (MLE) method is used to estimate the model parameters. Let $X_1, \dots, X_n \sim TIIHLEGTLE-G(\alpha, \beta, b, \xi)$ and $\Delta = (\alpha, \beta, b, \xi)^T$ be the vector of model parameters. The log-likelihood $\ell = \ell(\Delta)$ based on a random sample of size n from the TIIHLEGTLE-G

class of distributions is given by

$$\begin{aligned}
\ell(\Delta) &= n \ln(4b\alpha\beta) + \sum_{i=1}^n \ln g(x_i; \xi) + (\alpha - 1) \sum_{i=1}^n \ln \bar{G}(x_i; \xi) + (\beta - 1) \sum_{i=1}^n \ln(1 - \bar{G}^\alpha(x_i; \xi)) \\
&+ \sum_{i=1}^n \ln(1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta) + (b - 1) \sum_{i=1}^n \ln(1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2) \\
&- 2 \sum_{i=1}^n \ln(1 + (1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b),
\end{aligned}$$

The score vector $u = (\frac{\partial \ell}{\partial \alpha}, \frac{\partial \ell}{\partial \beta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})$

$$\begin{aligned}
\frac{\partial \ell}{\partial \alpha} &= \frac{n}{\alpha} + \sum_{i=1}^n \ln \bar{G}(x_i; \xi) - (\beta - 1) \sum_{i=1}^n \frac{\bar{G}^\alpha(x_i; \xi) \ln \bar{G}(x_i; \xi)}{1 - \bar{G}^\alpha(x_i; \xi)} \\
&+ \sum_{i=1}^n \frac{\beta(1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^\alpha(x_i; \xi) \ln \bar{G}(x_i; \xi)}{(1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^\beta} \\
&+ (b - 1) \sum_{i=1}^n \frac{2\beta(1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)(1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^\alpha(x_i; \xi) \ln \bar{G}(x_i; \xi)}{(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)} \\
&+ \sum_{i=1}^n \frac{4b\beta(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b}{(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b+1}} \\
&\times (1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b-1} (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)(1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^\alpha(x_i; \xi) \ln \bar{G}^\alpha(x_i; \xi),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \beta} &= \frac{n}{\beta} + \sum_{i=1}^n (1 - \bar{G}^\alpha(x_i; \xi)) - \sum_{i=1}^n \frac{(1 - \bar{G}^\alpha(x_i; \xi))^\beta \ln(1 - \bar{G}^\alpha(x_i; \xi))}{1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta} \\
&+ (b - 1) \sum_{i=1}^n \frac{2(1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)(1 - \bar{G}^\alpha(x_i; \xi))^\beta \ln(1 - \bar{G}^\alpha(x_i; \xi))}{1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2} \\
&- \sum_{i=1}^n \frac{4b(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)(1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b-1}}{(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b+1}} \\
&\times (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)(1 - \bar{G}^\alpha(x_i; \xi))^\beta \ln(1 - \bar{G}^\alpha(x_i; \xi)),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial b} &= \frac{n}{b} + \sum_{i=1}^n \ln(1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2) \\
&+ \sum_{i=1}^n \frac{2(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b(1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b}{(1 + (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b+1}} \\
&\times \ln(1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \xi_k} &= \sum_{i=1}^n \frac{\frac{\partial g(x_i; \xi)}{\partial \xi_k}}{g(x_i; \xi)} + \sum_{i=1}^n \frac{\frac{\partial \bar{G}(x_i; \xi)}{\partial \xi_k}}{\bar{G}(x_i; \xi)} + (\beta - 1) \sum_{i=1}^n \frac{\frac{\partial (1 - \bar{G}^\alpha(x_i; \xi))}{\partial \xi_k}}{1 - \bar{G}^\alpha(x_i; \xi)} \\
&+ \sum_{i=1}^n \frac{\alpha \beta (1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^{\alpha-1}(x_i; \xi) \frac{\partial \bar{G}^{\alpha-1}(x_i; \xi)}{\partial \xi_k}}{1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta} \\
&+ (b - 1) \sum_{i=1}^n \frac{2\alpha \beta (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta) (1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^{\alpha-1}(x_i; \xi) \frac{\partial \bar{G}^{\alpha-1}(x_i; \xi)}{\partial \xi_k}}{1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2} \\
&+ \sum_{i=1}^n \frac{6b\alpha \beta (1 + (1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b (1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^{b-1} \frac{\partial \bar{G}^{\alpha-1}(x_i; \xi)}{\partial \xi_k}}{(1 + (1 - (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta)^2)^b)^2} \\
&\times (1 - (1 - \bar{G}^\alpha(x_i; \xi))^\beta) (1 - \bar{G}^\alpha(x_i; \xi))^{\beta-1} \bar{G}^{\alpha-1}(x_i; \xi).
\end{aligned}$$

The MLE of the parameters, denoted by $\hat{\Delta}$ is obtained by solving the nonlinear equation $(\frac{\partial \ell}{\partial \alpha}, \frac{\partial \ell}{\partial \beta}, \frac{\partial \ell}{\partial b}, \frac{\partial \ell}{\partial \xi_k})^T = \mathbf{0}$. We use the Newton-Raphson numerical method to solve the nonlinear equations. Observed Fisher information matrix evaluated at $\hat{\Delta}$ is used to construct confidence intervals for the model parameters.

4.5 Some Special Cases of TIIHLEGTLEG-G Distribution

In this subsection, we present some special cases of the TIIHLEGTLEG-G family of distributions. We considered the Exponential distribution, Log-logistic distribution and the Weibull distribution defined in equations (1.13), (1.15) and (1.14) respectively.

4.5.1 Type II Half Logistic Topp-Leone Exponentiated Generalized Exponential (TIIHLEGTLEG-E) Distribution

The cdf and pdf of the TIIHLEGTLEG-E distribution is given by

$$F_{TIIHLEGTLEG-E}(x; \alpha, \beta, b, \lambda) = \frac{2(1 - (1 - (1 - e^{-\lambda \alpha x})^\beta)^2)^b}{1 + (1 - (1 - (1 - e^{-\lambda \alpha x})^\beta)^2)^b},$$

$$f_{TIIHLEGTLEG-E}(x; \alpha, \beta, b, \lambda) = \frac{4b\alpha \beta e^{\lambda \alpha x} (1 - e^{\lambda \alpha x})^{\beta-1} (1 - (1 - e^{\lambda \alpha x})^\beta) (1 - (1 - (1 - e^{\lambda \alpha x})^\beta)^2)^{b-1}}{(1 + (1 - (1 - (1 - e^{\lambda \alpha x})^\beta)^2)^b)^2},$$

respectively, for $\alpha > 0$, $\beta > 0$, $b > 0$ and $\lambda > 0$.

The corresponding hrf is given by

$$\begin{aligned}
h_{TIIHLEGTLEG-E}(x; \alpha, \beta, b, \lambda) &= \frac{4b\alpha \beta e^{\lambda \alpha x} (1 - e^{\lambda \alpha x})^{\beta-1} (1 - (1 - e^{\lambda \alpha x})^\beta)}{(1 + (1 - (1 - (1 - e^{\lambda \alpha x})^\beta)^2)^b)^2 (1 + (1 - (1 - (1 - e^{-\lambda \alpha x})^\beta)^2)^b)} \\
&\times (1 - (1 - (1 - (1 - e^{-\lambda \alpha x})^\beta)^2)^b) (1 - (1 - (1 - e^{\lambda \alpha x})^\beta)^2)^{b-1}.
\end{aligned}$$

Figure 4.1 shows the plots of the pdfs and hrfs for the TIIHLEGTLEG-E distribution. The pdf plots of the TIIHLEGTLEG-E display reverse-J, right skewed, almost symmetric and left skewed

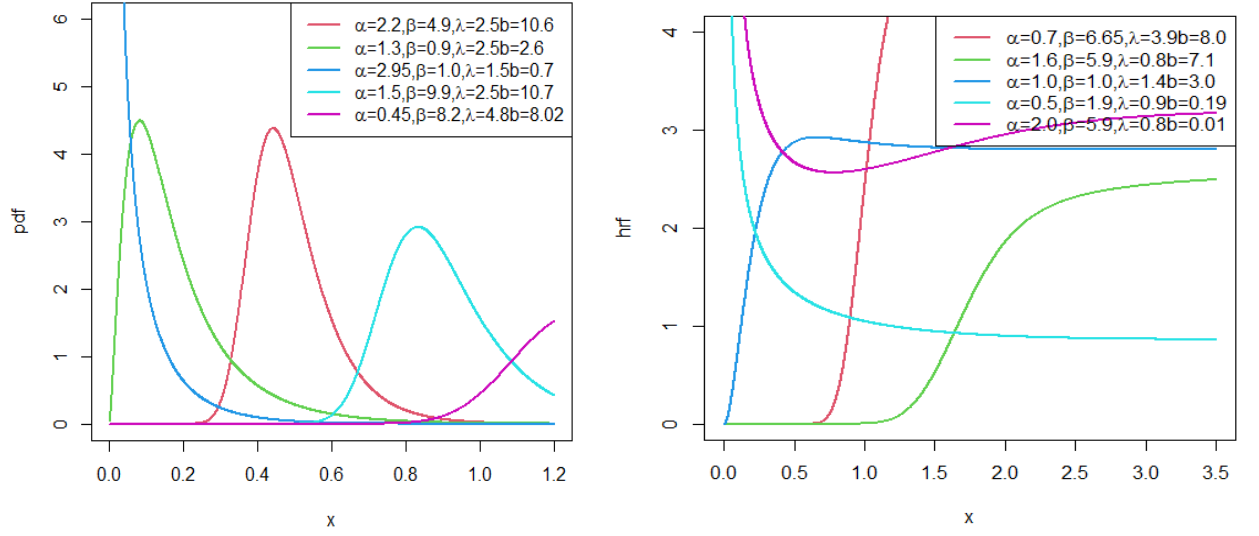


Figure 4.1: Pdf and hrf plots for the TIIHLTLEG-E distribution

shapes. The hrf plots of the TIIHLTLEG-E distribution display decreasing, bathtub, increasing and upside-down bathtub shapes.

	$(\alpha, \beta, b, \lambda)$				
u	(1.5,1.5,1.5,1.5)	(1.5,1,1.5,1.5)	(1.5,0.5,1.5,1)	(1.5,1.5,1,1.5)	(1,1.5,1,0.5)
0.1	0.08522038	0.03365434	0.0002985694	0.1278287	0.01250957
0.2	0.12638587	0.05840475	0.0014778835	0.1895804	0.03428394
0.3	0.16456363	0.08395074	0.0038387693	0.2468303	0.06473485
0.4	0.20404596	0.11235380	0.0080470411	0.3060674	0.10566868
0.5	0.24780282	0.14563769	0.0152250157	0.3716953	0.16068551
0.6	0.29931822	0.18674228	0.0272969425	0.4489820	0.23619353
0.7	0.36438392	0.24094314	0.0481948483	0.4489820	0.34454749
0.8	0.45542681	0.32007174	0.0874648339	0.6831421	0.51374056
0.9	0.61152403	0.46170583	0.1773382752	0.9172910	0.83312399

Table 4.1: Generated Random Variables of the TIIHLTLEG-E Distribution

Table 4.1 lists the generated random variables of the BTIIHLTL-E distribution via equation (4.3) for selected values of the parameters $(\alpha, \beta, b, \lambda)$.

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTLEG-E distribution are given in Table 4.2.

	$(\alpha, \beta, b, \lambda)$				
	(1.8, 0.8, 0.7, 0.5)	(2.0, 2.5, 0.9, 0.5)	(0.5, 2.0, 1.5, 1.5)	(0.9, 5.0, 1.8, 0.5)	(0.8, 4.5, 1.6, 0.4)
$E(X)$	0.094142387	0.25800682	0.31394007	0.40669569	0.36179626
$E(X^2)$	0.036229258	0.11703589	0.22275282	0.24164568	0.21145043
$E(X^3)$	0.020644669	0.06897486	0.17024100	0.16416981	0.14395931
$E(X^4)$	0.014024926	0.04704435	0.13676575	0.12158623	0.10737194
$E(X^5)$	0.010492751	0.03507085	0.11382735	0.09545645	0.08494762
$E(X^6)$	0.008333871	0.02770969	0.09724353	0.07809028	0.06998540
SD	0.006890897	0.02279297	0.08475055	0.06583837	0.05936834
CV	0.005863727	0.01930428	0.07502941	0.05678907	0.05147778
CS	0.005097704	0.01671317	0.06726518	0.04986006	0.04539957
CK	0.004505674	0.01471911	0.06092980	0.04439905	0.04058225

Table 4.2: Table of Moments for Selected Parameters of the TIIHLTLEG-E Distribution

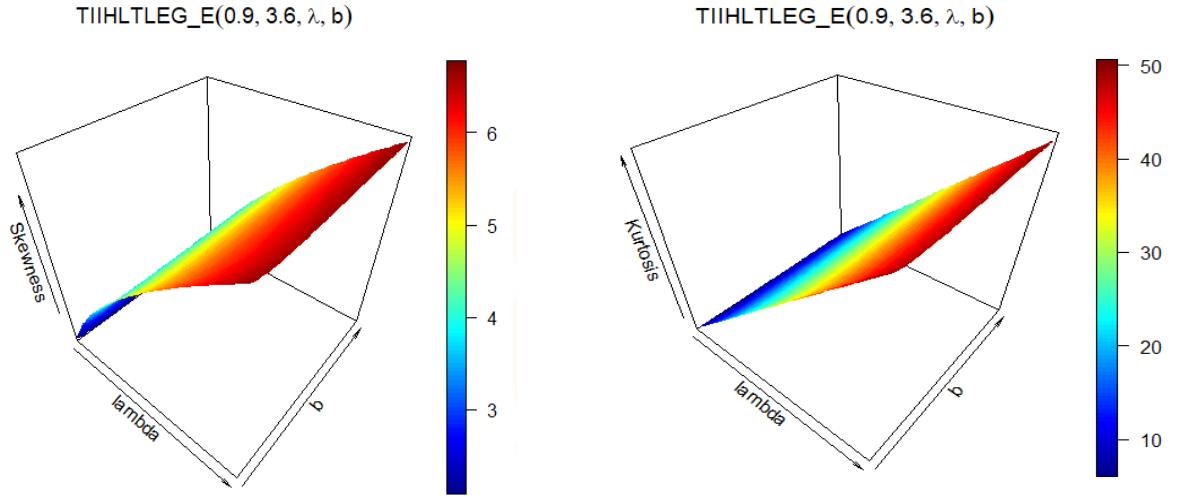


Figure 4.2: Skewness and Kurtosis plots for the TIIHLTLEG-E distribution

Figure 4.2 shows the plots of skewness and kurtosis for the TIIHLTLEG-E distribution. When we fix the parameters α and β , skewness and kurtosis increases.

4.5.2 Type II Half Log Logistic Topp-Leone Exponentiated Generalized Log Logistic (TIIHLTLEG-LLoG) Distribution

The cdf and pdf of the TIIHLTLEG-LLoG distribution is given by

$$F_{TIIHLTLEG-LLoG}(x; \alpha, \beta, b, c) = \frac{2(1 - (1 - (1 - (1 + x^c)^{-\alpha})^\beta)^2)^b}{1 + (1 - (1 - (1 - (1 + x^c)^{-\alpha})^\beta)^2)^b},$$

$$\begin{aligned} f_{TIIHLTLEG-LLoG}(x; \alpha, \beta, b, c) &= \frac{4b\alpha\beta cx^{c-1}(1 + x^c)^{-(\alpha+1)}(1 - (1 + x^c)^{-\alpha})^{\beta-1}}{(1 + (1 - (1 - (1 - (1 + x^c)^{-\alpha})^\beta)^2)^b)^2} \\ &\times (1 - (1 - (1 + x^c)^{-\alpha})^\beta)(1 - (1 - (1 - (1 + x^c)^{-\alpha})^\beta)^2)^{b-1}, \end{aligned}$$

respectively, for $\alpha > 0$, $\beta > 0$, $b > 0$ and $c > 0$.

The corresponding hrf is given by

$$\begin{aligned}
 h_{TIIHLTLEG-LLoG}(x; \alpha, \beta, b, c) &= \frac{4b\alpha\beta cx^{c-1}(1+x^c)^{-(\alpha+1)}(1-(1+x^c)^{-\alpha})^{\beta-1}}{(1+(1-(1-(1-(1+x^c)^{-\alpha})^{\beta})^2)^b)^2} \\
 &\times (1-(1-(1+x^c)^{-\alpha})^{\beta})(1-(1-(1-(1+x^c)^{-\alpha})^{\beta})^2)^{b-1} \\
 &\times \frac{1+(1-(1-(1-(1+x^c)^{-\alpha})^{\beta})^2)^b}{1-(1-(1-(1-(1+x^c)^{-\alpha})^{\beta})^2)^b}.
 \end{aligned}$$

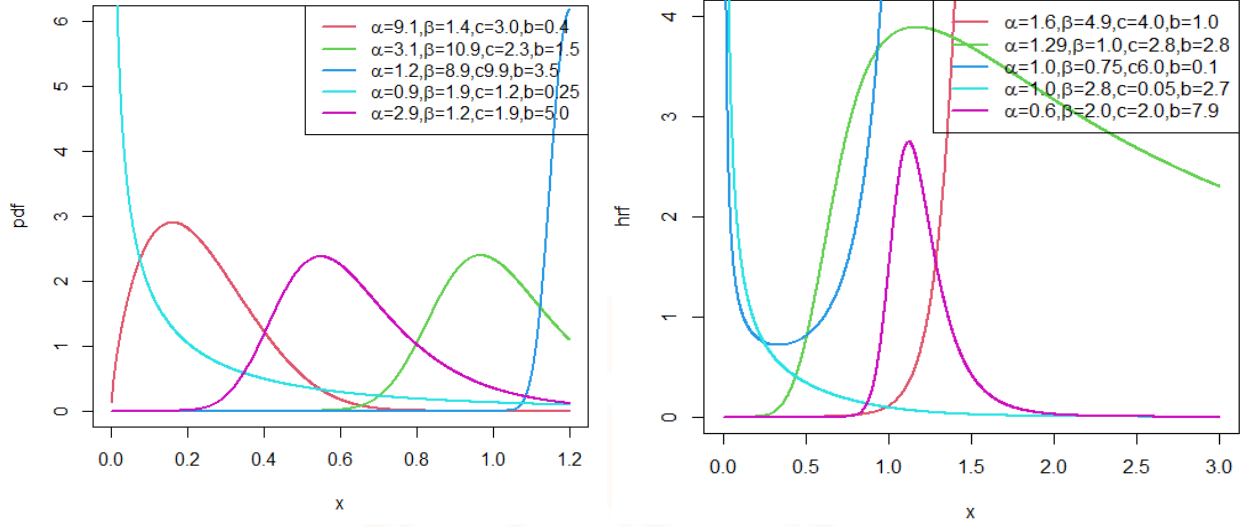


Figure 4.3: Pdf and hrf plots for the TIIHLTLEG-LLoG distribution

Figure 4.3 shows the plots of the pdfs and hrfs for the TIIHLTLEG-LLoG distribution. The pdf plots of the TIIHLTLEG-LLoG display left skewed, J, reverse-J, right skewed and symmetric shapes. The hrf plots of the TIIHLTLEG-LLoG distribution display decreasing, uni-modal, increasing, bathtub and upside-down bathtub shapes.

	(α, β, b, c)				
u	(1.8,1.2,1.7,1.5)	(2,2.5,0.9,0.5)	(1.5,0.5,1.2,1)	(0.9,1,1.8,0.5)	(1.3,1.3,1.4,0.4)
0.1	0.2070031	0.02642106	0.001695530	0.02741348	0.00993828
0.2	0.2765755	0.05427406	0.006083772	0.06312340	0.02786072
0.3	0.3364339	0.08693917	0.013579718	0.10644988	0.05308311
0.4	0.3954082	0.12684692	0.025330086	0.15898138	0.08701084
0.5	0.4584555	0.17771407	0.043283020	0.22389979	0.13248703
0.6	0.5307485	0.24592725	0.070949966	0.30699584	0.19479895
0.7	0.6203298	0.34435932	0.115492748	0.41966866	0.28456082
0.8	0.7446643	0.50474481	0.194487299	0.58943194	0.42739248
0.9	0.9606052	0.84303056	0.371488479	0.91343131	0.71337268

Table 4.3: Generated Random Variables of the TIIHLTLEG-LLoG Distribution

Table 4.3 lists the generated random variables of the TIIHLTLEG-LLoG distribution via equation (4.3) for selected values of the parameters (α, β, b, c) .

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLTLEG-LLoG distribution are given in Table 4.4.

	(α, β, b, c)				
	(0.8,1.3,1.9,1.4)	(0.9,0.5,1.6,1.5)	(0.5,2.0,1.5,1.5)	(0.9,0.9,1.8,0.5)	(0.8,1.5,1.6,1.4)
$E(X)$	0.37693453	0.25127049	0.09231621	0.22557197	0.32727328
$E(X^2)$	0.26232858	0.11837650	0.07254938	0.10643994	0.22884708
$E(X^3)$	0.19663614	0.07219099	0.05944260	0.06554707	0.17257858
$E(X^4)$	0.15529196	0.05060644	0.05019546	0.04628365	0.13710472
$E(X^5)$	0.12739102	0.03854569	0.04335872	0.03542164	0.11307604
$E(X^6)$	0.10752522	0.03097515	0.03811682	0.02855415	0.09589003
SD	0.09277191	0.02582599	0.03397962	0.02385808	0.08306722
CV	0.08143944	0.02211457	0.03063654	0.02045988	0.07317310
CS	0.07249218	0.01932035	0.02788210	0.01789397	0.06532845
CK	0.06526598	0.01714451	0.02557530	0.01589147	0.05896816

Table 4.4: Table of Moments for Selected Parameters of the TIIHLTLEG-LLoG Distribution

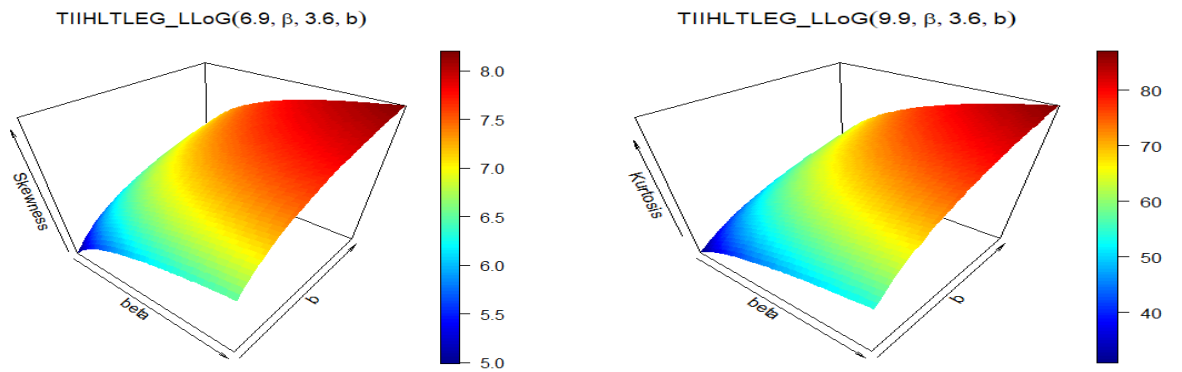


Figure 4.4: Skewness and Kurtosis plots for the TIIHLTLEG-LLoG distribution

Figure 4.4 shows the plots of skewness and kurtosis for the TIIHLTLEG-LLoG distribution. When we fix the parameters α and c , skewness and kurtosis increases.

4.5.3 Type II Half Logistic Topp-Leone Exponentiated Generalized Weibull (TIIHLTLEG-W) Distribution

The cdf and pdf of the TIIHLTLEG-W distribution is given by

$$F_{TIIHLTLEG-W}(x; \alpha, \beta, b) = \frac{2(1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b}{1 + (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b},$$

$$\begin{aligned} f_{TIIHLTLEG-W}(x; \alpha, \beta, b) &= \frac{4b\alpha\beta^2 x^{\beta-1} (e^{-\alpha x^\beta}) (1 - e^{-\alpha x^\beta})^{\beta-1} (1 - (1 - e^{-\alpha x^\beta})^\beta)}{(1 + (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b)^2} \\ &\times (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^{b-1}, \end{aligned}$$

respectively, for $\alpha > 0$, $\beta > 0$ and $b > 0$.

The corresponding hrf is given by

$$\begin{aligned} h_{TIIHLTLEG-W}(x; \alpha, \beta, b) &= \frac{4b\alpha\beta^2 x^{\beta-1} (e^{-\alpha x^\beta}) (1 - e^{-\alpha x^\beta})^{\beta-1} (1 - (1 - e^{-\alpha x^\beta})^\beta)}{(1 + (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b)^2} \\ &\times (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^{b-1} \\ &\times \frac{1 + (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b}{1 - (1 - (1 - (1 - e^{-\alpha x^\beta})^\beta)^2)^b}. \end{aligned}$$

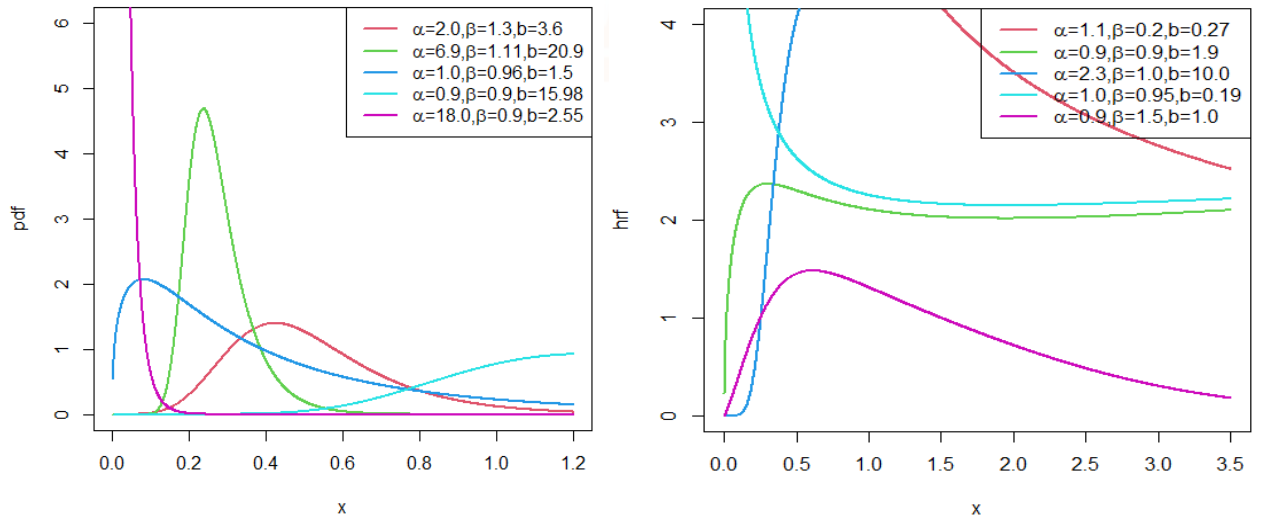


Figure 4.5: Pdf and hrf plots for the TIIHLTLEG-W distribution

Figure 4.5 shows the plots of the pdfs and hrfs for the TIIHLTLEG-W distribution. The pdf plots of the TIIHLTLEG-W display almost symmetric, right skewed and reverse-J shapes. The hrf plots of the TIIHLTLEG-W distribution display decreasing, increasing and upside-down

bathtub shapes.

	(α, β, b)				
u	(0.6,1,0.5)	(0.9,1,1)	0.9,0.7,1.8)	(0.9,1,0.5)	(0.9,0.9,1.1)
0.1	0.002334797	0.03003679	0.01150103	0.001534869	0.01818370
0.2	0.010374366	0.06543657	0.02928575	0.006905339	0.04369864
0.3	0.026369839	0.10787942	0.05405662	0.017580450	0.07677743
0.4	0.053793405	0.15982316	0.08826206	0.035854696	0.11971572
0.5	0.098152454	0.22525838	0.13636249	0.065435633	0.17644752
0.6	0.169137556	0.31091590	0.20643411	0.112743462	0.25407519
0.7	0.285331410	0.42954966	0.31507308	0.190228604	0.36639140
0.8	0.489821995	0.61034243	0.50296658	0.326548278	0.54555060
0.9	0.922428582	0.94708757	0.91443825	0.614951160	0.89774512

Table 4.5: Generated Random Variables of the TIIHLEGTLEG-W Distribution

Table 4.5 lists the generated random variables of the TIIHLEGTLEG-W distribution via equation (4.3) for selected values of the parameters (α, β, b) .

A table of moments, standard deviation (SD), coefficient of variation (CV), coefficient of skewness (CS), and coefficient of kurtosis (CK) for selected parameter values of the TIIHLEGTLEG-W distribution are given in Table 4.6.

	(α, β, b)				
	(0.8,0.5,0.7)	(0.2,0.3,0.8)	(0.5,0.8,0.6)	(0.5,0.5,0.7)	(0.8,0.8,0.4)
$E(X)$	0.07397530	0.043593267	0.14706221	0.08339316	0.08984031
$E(X^2)$	0.03449697	0.021820770	0.07195694	0.04054235	0.04045787
$E(X^3)$	0.02222646	0.014540543	0.04685756	0.02660947	0.02542861
$E(X^4)$	0.01634388	0.010900827	0.03457925	0.01977119	0.01839509
$E(X^5)$	0.01290875	0.008717878	0.02735107	0.01571958	0.01436478
SD	0.17036616	0.141139637	0.22434270	0.18327012	0.17996274
CV	2.30301402	3.237647632	1.52549519	2.19766373	2.00314019
CS	3.11039419	4.215639569	1.90169700	2.86346131	2.74083455
CK	12.83180380	21.680538687	5.90173778	11.02832164	10.50716997

Table 4.6: Table of Moments for Selected Parameters of the TIIHLEGTLEG-W Distribution

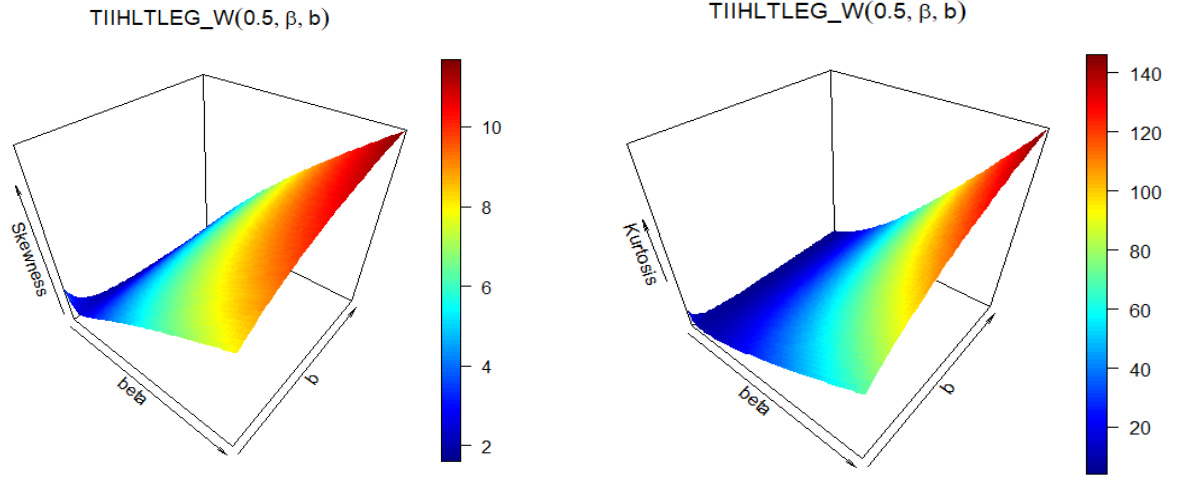


Figure 4.6: Skewness and Kurtosis plots for the TIIHLEGTLEG-W distribution

Figure 4.6 shows the plots of skewness and kurtosis for the TIIHLEGTLEG-W distribution. When we fix the parameter α , skewness and kurtosis increases.

4.6 Simulation Study

In this section, a simulation study was conducted to examine consistency of the maximum likelihood estimators. We considered simulations from the TIIHLEGTLEG-G distribution for the sample sizes $n=80$, $n=100$, $n=200$, $n=400$, $n=800$ and $n=1000$. The mean, root mean square error (RMSE), and average bias for the estimated parameter, say $\hat{\Delta}$ are calculated using the formulae

$$Bias(\hat{\Delta}) = \frac{1}{N} \sum_{i=1}^N (\hat{\Delta}_i - \Delta), \quad \text{and} \quad RMSE(\hat{\Delta}) = \sqrt{\frac{\sum_{i=1}^N (\hat{\Delta}_i - \Delta)^2}{N}},$$

respectively. If the model performs better, we expect the mean to approximate the true parameter values, the RMSE, and bias to decay toward zero for an increase in sample size.

From the results in Table 4.7, the mean figures indicate an approximation to the true parameter values with the RMSE and bias values decreasing towards zero for all the parameter values.

4.7 Applications

In this section, two data sets examples are presented to illustrate how useful the TIIHLEGTLEG-G distribution is in real life by comparing it to several non-nested models. The maximum likelihood estimation technique, via the R software, was used to estimate parameters. Goodness-of-fit statistics were used to determine how the model will perform, and they include; $-2\log\text{likelihood}$ ($-2\log L$), Akaike Information Criterion (AIC), Consistent Akaike Information Criterion (CAIC), Bayesian Information Criterion (BIC), Cramer-von Mises (W^*) and Anderson-Darling (A^*), Kolmogorov-Smirnov (K-S) statistic and its p-value. The model with the smallest values of

	$\alpha = 1.6, \beta = 1.9, \lambda = 1.6, b = 0.4$			$\alpha = 1.6, \beta = 1.6, \lambda = 1.6, b = 0.4$			
samplesize	Mean	RMSE	Bias	Mean	RMSE	Bias	
α	80	1.5548	0.2531	-0.0452	1.5481	0.2663	-0.0519
	100	1.5353	0.2495	-0.0647	1.5435	0.2612	-0.0565
	200	1.5442	0.1832	-0.0558	1.5387	0.1996	-0.0613
	400	1.5600	0.1374	-0.0399	1.5564	0.1464	-0.0436
	800	1.5787	0.1049	-0.0213	1.5772	0.1182	-0.0228
	1000	1.5829	0.0902	-0.0171	1.5781	0.1066	-0.0219
β	80	1.9605	1.4752	0.0605	1.6241	1.2326	0.02406
	100	1.8232	1.4415	-0.0768	1.5714	1.2589	-0.0286
	200	1.7492	1.1577	-0.1508	1.4556	1.0149	-0.1444
	400	1.8033	0.9268	-0.0967	1.4951	0.7934	-0.1049
	800	1.8459	0.7461	-0.0541	1.5663	0.7239	-0.0337
	1000	1.8705	0.6893	-0.0295	1.5673	0.6707	-0.0327
λ	80	1.5548	0.2531	-0.0452	1.5481	0.2663	-0.0516
	100	1.5353	0.2495	-0.0647	1.5435	0.2612	-0.0565
	200	1.5442	0.1832	-0.0558	1.5387	0.1996	-0.0613
	400	1.5600	0.1374	-0.0399	1.5564	0.1464	-0.0436
	800	1.5786	0.1049	-0.0213	1.5772	0.1182	-0.0228
	1000	1.5829	0.0902	-0.0171	1.5781	0.1066	-0.0219
b	80	2.1063	8.2918	1.7064	2.2816	8.8766	1.8816
	100	2.1185	8.5152	1.7185	2.1398	7.9883	1.7399
	200	0.8657	1.0343	0.4657	0.9019	1.0493	0.5019
	400	0.5086	0.4899	0.2137	0.6299	0.5014	0.2299
	800	0.5086	0.2778	0.1086	0.5278	0.2945	0.1278
	1000	0.4833	0.2319	0.0833	0.5043	0.2483	0.1043

Table 4.7: Monte Carlo Simulation Results for TIHLETLG-E Distribution: Mean, RMSE and Average Bias

these statistics and a bigger p-value for the K-S statistic is considered as the best model.

The TIHTLTLEG-G distribution was compared to the following non-nested models: Type II Exponentiated Half Logistic Topp-Leone Weibull Logarithmic (TIEHLTLWL) distribution by Moakofi et al. [68], Exponentiated Half Logistic-Power Generalized Weibull-Log Logistic (EHLPGWLLoG) distribution by Oluyede et al. [78], Type I Half Logistic-Burr XII (HLBXII) distribution by Illikkan and Dil [47], Topp-Leone Weibull Lomax (TLWLx) distribution by Jamal et al. [50] and Type II General Inverse Exponential-Lomax (TIIGIE-Lx) distribution by Jamal et al. [49]. The pdfs of the non-nested distributions are given by

$$f_{TIEHLTLWL}(x; a, b, \theta, \lambda) = \frac{4ab\theta\lambda x^{\lambda-1}e^{-x^\lambda}[1 - e^{-2x^\lambda}]^{b-1}e^{-x^\lambda}}{(1 + [1 - e^{-2x^\lambda}]^b)^{a+1}} \times (1 - [1 - e^{-2x^\lambda}]^b)^{a-1} \frac{(1 - \theta[\frac{1-(1-e^{-2x^\lambda})}{1+(1-e^{-2x^\lambda})}]^a)^{-1}}{-\log(1 - \theta)},$$

for $a, b, \theta, \lambda > 0$.

$$f_{HLPGWLLoG}(x; \alpha, \beta, \delta, c) = 2\alpha\beta\delta \left[1 + \left(\frac{1 - (1+x^c)^{-1}}{(1+x^c)^{-1}} \right)^\alpha \right]^{\beta-1} e^{\left(1 - \left[1 + \left(\frac{1 - (1+x^c)^{-1}}{(1+x^c)^{-1}} \right)^\alpha \right]^\beta \right)} \\ \times ((1+x^c)^{-1})^{\alpha+3} \left(1 + e^{\left(1 - \left[1 + \left(\frac{1 - (1+x^c)^{-1}}{(1+x^c)^{-1}} \right)^\alpha \right]^\beta \right)} \right)^{-2} \\ \times \left[\frac{1 - e^{\left(1 - \left[1 + \left(\frac{1 - (1+x^c)^{-1}}{(1+x^c)^{-1}} \right)^\alpha \right]^\beta \right)}}{1 + e^{\left(1 - \left[1 + \left(\frac{1 - (1+x^c)^{-1}}{(1+x^c)^{-1}} \right)^\alpha \right]^\beta \right)}} \right] cx^{c-1} (1 - (1+x^c)^{-1})^{\alpha-1},$$

for $\alpha, \beta, \delta, c > 0$.

$$f_{HLBXII}(\alpha, \lambda, a, b) = 2\alpha, \lambda, a, bx^{a-1} e^{\lambda(1-(1+x^a)^b)} \\ \times \frac{(1 - e^{\lambda(1-(1+x^a)^b)})^{\alpha-1}}{(1+x^a)^{-b+1} (1 + e^{\lambda(1-(1+x^a)^b)}) (\alpha + 1)},$$

for $a, b, \alpha, \lambda > 0$.

$$f_{TLWLx}(\alpha, \theta, a, b) = 2\theta\alpha ab(1+bx)^{a\alpha-1} (1 - (1+bx)^{-a})^{\alpha-1} \\ \times e^{-2 \left(\frac{1-(1+bx)^{-a}}{(1+bx)^{-a}} \right)^\alpha} \left(1 - e^{-2 \left(\frac{1-(1+bx)^{-a}}{(1+bx)^{-a}} \right)^\alpha} \right)^{\theta-1},$$

for $\alpha, \theta, a, b > 0$.

$$f_{TIIGIE_L}(\lambda, \alpha, a, b) = \lambda\alpha \left(\frac{a}{b} \right) \left(1 + \frac{x}{b} \right)^{-a-1} \left(1 + \left(\frac{x}{b} \right) \right)^{a(\alpha+1)} \\ \times e^{\lambda(1-(1+\frac{a}{b})^{\alpha\alpha})}.$$

for $\lambda, \alpha, a, b > 0$.

4.7.1 Filure Times of Air Conditioning System Data Set

We considered the failure times of air conditioning system data set for our applications obtained from Mukherjee et al. [71]. The data is presented in Table 4.8

1	3	5	7	11	11	11	12	14	14	14	16
16	20	21	23	42	47	52	62	71	71	87	
90	95	120	120	225	246	261					

Table 4.8: Filure Times of Air Conditioning System Data

The estimated variance-covariance matrix for the TIIHLTLEG-G model on failure times of the

Model	Estimates				Statistics							
	α	β	λ	b	$-2 \log L$	AIC	CAIC	BIC	W*	A*	KS	P-value
TIHLTEGE	2.5654 (2.7751×10^{-7})	9.8618×10^{-2} (1.4997×10^2)	1.3277×10^{-3} (5.3695×10^{-4})	2.5564×10^1 (3.4846×10^{-5})	302.3	310.3	311.9	315.9	0.0074	0.4067	0.1644	0.8110
HLBXII	1.5476 -	1.3104×10^{-5} -	2.5273×10^{-2} -	1.5738×10^1 -	374.1	382.2	383.8	387.8	0.0794	0.4282	0.2788	0.0188
TIEHLTLWL	1.0019 (-)	9.9876×10^{-1} (-)	1.3301×10^{-9} (8.6224×10^{-3})	4.9717×10^{-2} (-)	508.1	516.1	517.7	521.7	0.0713	0.4045	0.9278	$< 2.2 \times 10^{-16}$
TLWLx	0.1456 (0.1533)	0.7436 (4.0127)	3.0145 (3.2253)	0.6547 (0.8466)	302.4	310.4	312.0	316.0	0.0770	0.4243	0.1312	0679
TIIGIE-Lx	0.0408 (0.1560)	1.8917 (0.0195)	0.4284 (0.0863)	1.0000 (3.9895)	303.5	311.5	313.1	317.1	0.0967	0.5429	0.14226	0.5783

Table 4.9: Parameter estimates and goodness-of-fit statistics for various models fitted for Failure Times of the Air Conditioning System of an Airplane Data

air conditioning system of an airplane data.

$$\begin{bmatrix} 7.7010 \times 10^{-14} & 3.4164 \times 10^{-9} & 1.4900 \times 10^{-10} & 7.9384 \times 10^{-12} \\ 3.4164 \times 10^{-9} & 2.2491 \times 10^{-4} & 6.6150 \times 10^{-6} & 5.2258 \times 10^{-7} \\ 1.4900 \times 10^{-10} & 6.6150 \times 10^{-6} & 2.8831 \times 10^{-7} & 1.5370 \times 10^{-8} \\ 7.9384 \times 10^{-12} & 5.2258 \times 10^{-7} & 1.5370 \times 10^{-8} & 1.2142 \times 10^{-9} \end{bmatrix}$$

and the 95% confidence intervals for the model parameters are given by

$$\alpha \in [2.5654 \pm 5.4391 \times 10^{-7}], \beta \in [9.8618 \times 10^{-2} \pm 2.9394 \times 10^{-2}], \lambda \in [1.3277 \times 10^{-3} \pm 1.0524 \times 10^{-3}], b \in [2.5564 \times 10^1 \pm 6.8297 \times 10^{-5}].$$

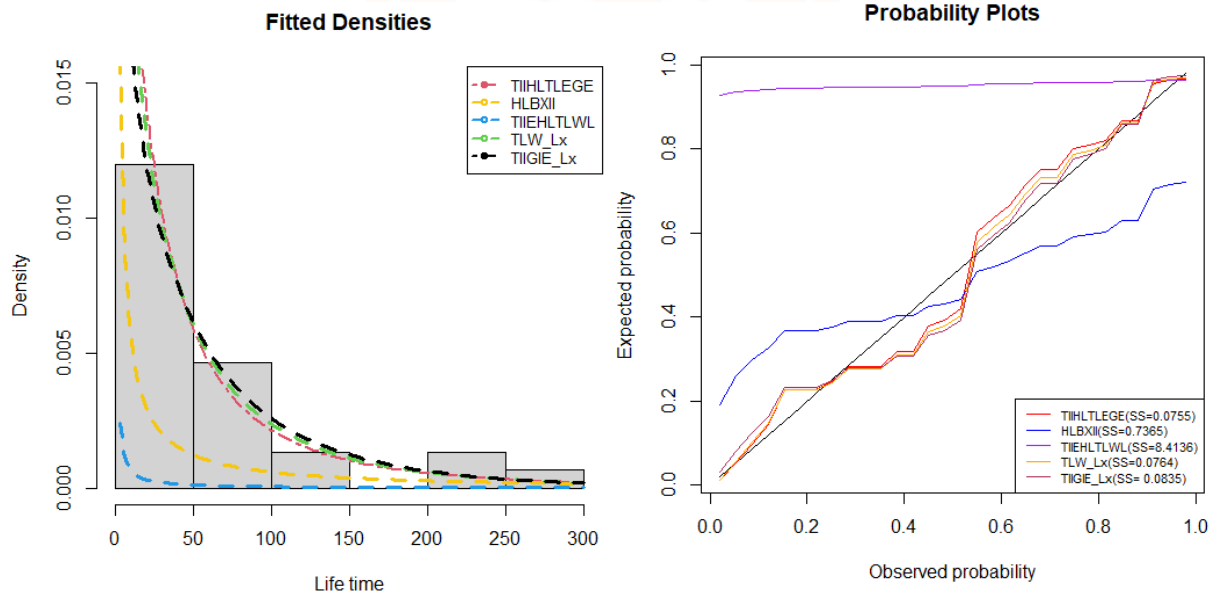


Figure 4.7: Fitted Densities and Probability Plots for the Failure Times of Air Conditioning System of an Airplane Data

From results shown in Table 4.9, we determine that the TIHLTEGL-E model outperforms the other models as it has the least value of goodness of fit statistics with a larger value of the K-S p-value. Also by observing plots of the fitted densities and probability in Figure 4.7, we arrive at the same conclusion.

4.7.2 Survival Times of Guinea Pigs Data Set

The second data set considered was the the survival times (days) of guinea pigs injected with different doses of tubercle bacilli. The regimen number is the common logarithm of the number of bacillary units in 0.5 ml of challenge solution; that is, regimen 6.6 corresponds to 4.0×10^6 bacillary units per 0.5 ml ($\log(4.0 \times 10^6) = 6.6$). The data has been analyzed by Kundu and Howler [54] and Dube et al. [38] and is presented in in Table 4.10.

12	15	22	24	24	32	32	33	34	38	38	43	44	48	52	53	54	54	55
56	57	58	58	59	60	60	60	60	61	62	63	65	65	67	68	70	70	72
73	75	76	76	81	83	84	85	87	91	95	96	98	99	109	110	121	121	127
131	143	146	146	175	175	211	233	258	258	263	297	341	341	376				

Table 4.10: Survival Times of Guinea Pigs Data



Model	Estimates				Statistics							
	α	β	λ	b	$-2 \log L$	AIC	CAIC	BIC	W^*	A^*	KS	P-value
TIHILTLEGE	2.4828×10^1 (1.0808×10^{-9})	2.2748×10^{-2} (2.6602×10^{-3})	1.9383×10^{-4} (3.6833×10^{-5})	1.5640×10^3 (1.9993×10^{-8})	779.4	787.4	788.0	796.5	0.1397	0.7581	0.0094	0.5463
HLBXII	8.0381×10^{-1} (3.7228×10^{-4})	1.9748×10^{-5} (7.1201×10^{-6})	1.8769×10^{-1} (1.0966×10^{-2})	8.8644 (2.6410×10^{-4})	850.0	858.0	858.6	867.1	0.5463	3.0227	0.2084	0.0038
TIHHLTLWL	0.0657 (0.0180)	0.1422 (0.0659)	0.0046 (0.3008)	0.4321 (0.0465)	923.1	931.1	931.7	940.2	0.2447	1.3291	0.4645	6.417×10^{-14}
TLWLx	0.1358 (0.0392)	3.0700 (2.5829)	2.4138 (1.1961)	8.1417 (7.0730)	780.3	788.3	788.9	797.4	0.1406	0.7689	0.0993	0.4754
TIIGIE-Lx	1.3567×10^{-3} (8.5011×10^{-4})	9.8949 (1.7528×10^{-4})	1.4154×10^{-1} (1.2253×10^{-2})	1.0001 (5.8699×10^{-2})	794.3	802.3	802.9	811.4	0.4341	2.3907	0.1458	0.0936

Table 4.11: Parameter estimates and goodness-of-fit statistics for various models fitted for Survival Times of Guinea Pigs Injected with Different Doses of Tubercle Bacilli Data.

The estimated variance-covariance matrix for the TIHILTLEG-G model on survival times of guinea pigs injected with different doses of tubercle bacilli data.

$$\begin{bmatrix} 1.1681 \times 10^{-18} & 2.8565 \times 10^{-12} & 3.7691 \times 10^{-14} & 2.1468 \times 10^{-17} \\ 2.8565 \times 10^{-12} & 7.0768 \times 10^{-6} & 8.8577 \times 10^{-8} & 5.3185 \times 10^{-11} \\ 3.7691 \times 10^{-14} & 8.8577 \times 10^{-8} & 1.3566 \times 10^{-9} & 6.6569 \times 10^{-13} \\ 2.1468 \times 10^{-17} & 5.3185 \times 10^{-11} & 6.6569 \times 10^{-13} & 3.9970 \times 10^{-16} \end{bmatrix}$$

and the 95% confidence intervals for the model parameters are given by

$$\alpha \in [2.4828 \times 10^1 \pm 2.1184 \times 10^{-9}], \beta \in [2.2748 \times 10^{-2} \pm 5.2140 \times 10^{-3}], \lambda \in [1.9383 \times 10^{-3} \pm 7.2192 \times 10^{-5}], b \in [1.5640 \times 10^3 \pm 3.9185 \times 10^{-8}].$$

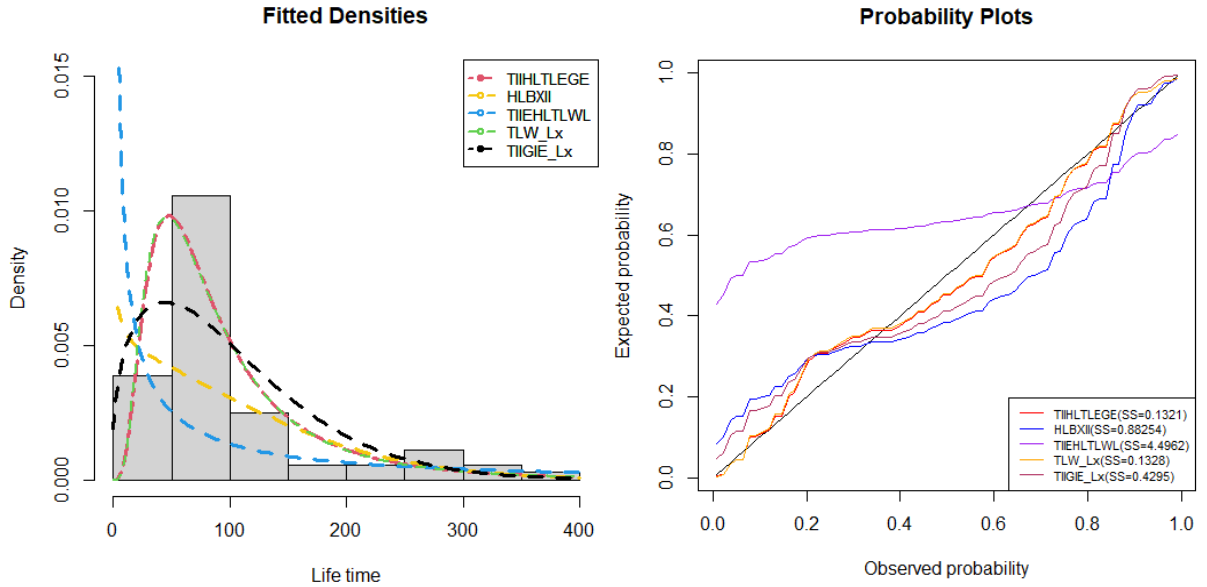


Figure 4.8: Fitted Densities and Probability Plots for the Survival Times of Guinea Pigs Data

From the results shown in Table 4.11, we determine that the TIHILTLEG-E outperforms the other models as it has the least value of goodness of fit statistics with a larger value of the K-S p-value. Also by observing plots of the fitted densities and probability in Figure 4.8, we arrive at the same conclusion.

4.8 Concluding Remarks

A new family of distributions, namely, the Type II Half Logistic Topp-Leone Exponentiated Generalized-G (TIIHLTLEG-G) was developed. Special cases and structural properties of the new distribution are presented. A simulation study was conducted to assess the consistency of the maximum likelihood estimates. Two real data examples were presented to illustrate the usefulness of the new family of distributions. The new model performs better than several existing models that were considered.



5. Conclusion and Discussion

5.1 Discussion and Conclusion

Three new generalized distributions called the Type II Half Logistic Topp-Leone-G Power Series (TIIHLTL-GPS), Beta Type II Half Logistic Topp-Leone-G (BTIIHLTL-G) and the Type II Half Logistic Topp-Leone Exponentiated Generalized-G (TIIHLTLEG-G) were proposed and developed as extensions of the Type II Half Logistic Topp-Leone-G distribution (TIIHLTL-G). We used three methods of generating statistical distributions, namely: the compounding method, the beta-G generator and the exponentiated generalized-G generator. Various characterization methods of distributions were used to define the new distributions, these include the series expansion of pdfs, characterizations such as the survival functions, hazard functions, reverse hazard functions and quantile functions. We also introduced special models of the new distributions, derived their moments, conditional moments, Rényi entropy and distribution of order statistics. An assessment of the performance of the distributions was carried out by performing a simulation study for the special cases of each distribution. Lastly, we applied the special cases of the distributions to real data sets to indicate how useful the models are when fitted to actual data sets.

Bibliography

- [1] Alexander, C., Cordeiro, G. M., Ortega, E. M. M. and Sarabia J. M., (2012). *Generalized beta-generated distributions*. Compu. Stat. Data. Anal., 56, 1880-1897.
- [2] Algeria, J. M. S., Mendoza, F. P., Gil. V. J., and Pérez, L. R., (2016). *Some models of beta-generated distributions with applications in finance*. Anales de ASEPUMA, (24) 18.
- [3] Aljarrah, M. A., Lee C. and Famoye F., (2014). *On Generating T-X family of distributions using quantile functions*. J. Stat. Distrib. App, 1, 2.
- [4] Alkarni, S. H., (2018). *Generalized Lindley Power Series Family of Distributions*. Advances and Applications in Statistics, 53(5), 537-567.
- [5] Allignol, A., Schumacher, M. and Wanner, C., (2011). *Understanding competing risks: A simulation point of view*. BMC Med. Res. Methodol., 11, 86.
- [6] Alizadeh, M., Cordeiro, G. M., Brito, E. D., and B Demétrio, C. G. (2015). *The beta Marshall-Olkin family of distributions*. Journal of Statistical Distributions and Applications, 2(1), 1-18.
- [7] Al-Shomrani, A., Arif, O., Shawky, A., Hanif, S. and Shahbaz, M. Q., (2016). *Topp-Leone Family of Distributions: Some Properties and Application*. Pak. J. Stat. Oper. Res., 12(3), 443-451.
- [8] Al-Mofleh, H., Elgarhy, M., Affify, A. and Zannon, M., (2020). *Type II Exponentiated Half Logistic generated family of distributions with applications*. Electronic Journal of Applied Statistical Analysis, 13(2), 536-561.
- [9] Alzaatreh, A., Lee, C. and Famoye, F., (2013). *A new method for generating families of continuous distributions*. Metron, 71(1), 63-79.
- [10] Amir, E. A. H. E. (2012). *On uses of mean absolute deviation: decomposition, skewness and correlation coefficients*. Metron, 37(2), 70(2), 145-164.
- [11] Arcani, A. and Porro, F., (2014). *The Graphical Representation of Inequality*. Current Topics in Statistical Graphics, 37(2), 419-436.
- [12] Aristov, V. V., Bulchelnikov, A. S. and Nechipurenko, Y. D., (2022). *The Use of the Statistical Entropy in Some New Approaches for the Description of Biosystems*. Entropy (Basel), 24(2), 172.
- [13] Aryal, G. R. and Yousof H, M., (2017). *The Exponentiated Generalized G-Poisson Family of Distributions*. Stochastic and Quality Control, 32(1), 7-23.

- [14] Badmus, N., Faweya, O., and Adeleke, K. (2020). *Generalized Beta-Exponential Weibull Distribution and its Applications*. Journal of Statistics: Advances in Theory and Applications, 24.
- [15] Baez, J., (2011). *Rényi Entropy and Free Energy*. 24.10.3390/e24050706
- [16] Balakrishnan N., (1985). *Order Statistics from the Half Logistic Distribution*. Journal of Statistical Computation and Simulation, 20 , 287-309.
- [17] Bantan, R. A. R., Jamal, F., Chesneau, C. and Elgarhy, M., (2020). *Type II Power Topp-Leone Generated Family of Distributions with Statistical Inference and Applications*. Symmetry, 12(1), 75.
- [18] Barreto-Souza, W., and Simas, A. B. (2013). *The exp-G family of probability distributions*. Brazilian Journal of Probability and Statistics, 27(1), 84–109.
- [19] Bashkirov, A. G., (2006). *Rényi entropy as a statistical entropy for complex systems*. Theor Math Phys, 149, 1559-1573
- [20] Bonferroni, C., (1930). *Elementi di Statistica Generale*. Seeber, Firenze.
- [21] Bourguignon, M., Silva, R. B., Dias, C. R. B. and Cordeiro, G.M., (2014). *A New Class of Fatigue Life Distributions*. Journal of Statistical Computation and Simulation, 12, 2619-2635.
- [22] Buryak, F. and Mishura, Y., (2021). *Convexity and robustness of Rényi entropy*. Modern Stoch. Theory Appl., 3, 387-412.
- [23] Cavanaugh, J. E. and Neath, A. A., (2002). *Generalizing The Derivation of the Schwarz Information Criterion*. Communications in Statistics -Theory and Methods..
- [24] Cavanaugh, J. E. and Neath, A. A., (2019). *The Akaike Information Criterion: Background, derivation, properties ,application, interpretation and refinements*. WIREs Computational Statistics, 11(3), e1460.
- [25] Chahkandi, M., Ganjali, M., (2009). *On Some Lifetime Distributions with Decreasing Failure Rate*. Computational Statistics & Data Analysis, 53, 4433-4440.
- [26] Chambers, J., Cleveland, W., Kleiner, B. and Tukey, P. (1983). *Graphical Methods of Data Analysis*. Chapman and Hall.
- [27] Chamunorwa, S., Oluyede, B., Makubate, B. and Chipepa, F., (2021). *The Exponentiated Odd Weibull-Topp-Leone-G Family of Distributions: Model, Properties and Applications*. Pak. J. Statist, 37(2), 143-158.
- [28] Chehade, S. S. and Vershynina, A., (2019). *Quantumentropies*. Scholarpedia, 14(2), 53131.
- [29] Chen, G. and Balakrishnan, N., (1985). *A General Purpose Approximate Goodness-of-fit Test*. Journal of Quality Technology, 27(2), 154-161.

- [30] Chipepa, F., Oluyede, B. and Makubate, B., (2020). *The Topp-Leone-Marshall-Olkin-G Family of Distributions with Applications*. International Journal of Statistics and Probability, 9(4), 1-15.
- [31] Chipepa, F., Oluyede, B., Makubate, B., and Fagbamigbe, A.F., (2019). *The Beta Odd Lindley-G Family of Distributions with Applications*. International Journal of Statistics and Probability, 17(1), 51-84.
- [32] Cordeiro, G. M., Simas, A. B., and Stosic, B. D., (2008). *The beta Weibull distribution*. Submitted to the Journal of Statistical Computation and Simulation.
- [33] Cordeiro, G. M., and de Castro, M., (2011). *A new family of generalized distributions* Journal of Statistical Computation and Simulation, 81(7), 883-898.
- [34] Cordeiro, G. M., Silva G.O. and Ortega E.M.M., (2012). *The beta extended Weibull distribution*. J Probab Stat Sci 10, 15-40.
- [35] Cordeiro, G. M., Cintra R., Rego, L. and Ortega E.M.M., (2012). *The McDonald Normal Distribution*. Pakistan Journal of Statistics and Operational Research.
- [36] Cordeiro, G. M., Ortega E.M.M. and da Cunha, D. C., (2013). *The exponentiated generalized class of distributions*. Journal of data science, 11(1), 1-27.
- [37] Cordeiro, G.M., Silva G.O., Pescim R.R. and Ortega E.M.M. (2014). *General properties for the beta extended half-normal distribution*. J Stat Comput Simul 84, 881-901.
- [38] Dube, M., Krishna, H. and Garg, R., (2015). *Generalized inverted exponential distribution under progressive first-failure censoring*. Journal of Statistical Computation and Simulation, 86, 1-20.
- [39] Eugene N., Lee C. and Famoye F. (2002). *Beta-normal distribution and its applications..* Commun Statist Theor Meth 31: 497-512.
- [40] Fine, J. and Lindqvist, B., (2014). *Competing Risks*. Lifetime data analysis.
- [41] Fuller Jr, E. R., Freiman, S. W., Quinn, J. B., Quinn, G. D. and Carter, W. C., (1994). *Fracture Mechanics Approach to the Design of Glass Aircraft Windows: A Case Study*. In *Window and Dome Technologies and Materials IV*. International Society for Optics and Photonics, 2286, 419-430.
- [42] Gradshteyn, I. S. and Ryzhik, I. M., (2000). *Tables of Integrals, Series and Products Sixth Edition*. Academic Press, San Diego.
- [43] Gurler, U., (1999). *Reverse Hazard*. Encyclopedia of Statistical Sciences, UV.3, 643-644.
- [44] Harvich, C.M and Tsai, C.L., (1989). *Regression and time series model: selection in small samples*. Biometrika 76, 297-307
- [45] Hassan, A. S., Elgarhy, M. and Shakil, M., (2017). *Type II Half Logistic Family of Distributions with Applications*. Pak. J. Stat. Oper. Res., 13 2, 245-264.

- [46] Ibrahim, M., and Yousef, H., (2020). *Transmuted Topp-Leone Weibull lifetime distribution: statistical properties and different method of estimation*. Pakistan Journal of Statistics and Operation Research, 501-515.
- [47] Illikkan, E., and Dil, E., (2020). *The Type I Half Logistic Burr XII Lifetime Distribution: properties and applications to survival data*.
- [48] Jafari, A. A. and Tahmasebi, S., (2016). *Gompertz-Power Series Distributions*. Communications in Statistics-Theory and Methods, 45(13), 3761-3781.
- [49] Jamal F., Chesneau C., and Elgarhy, M., (2020). *Type II general inverse exponential family of distributions* Journal of Statistics and Management Systems, 23(3), 617-641
- [50] Jamal F., Reyad, H. M., Nasir, M. A., Chesneau C., Shah, M. A. A., and Ahmed, S. O., (2019). *Topp-Leone Weibull-Lomax distribution: properties, regression, model and applications*.
- [51] Johnson, N.L, Kotz, S., and Balakrishnan, N., (1994). *Continuous Univariate Distributions*. volume 1, John Wiley & Sons, New York.
- [52] Kartsonaki, C., (2016). *Survival Analysis: Diagnostic Histopathology*. Mini-Symposium: Medical Statistics, 22.7, 263-270.
- [53] Kotz, S., (1994). *Characteristics of Statistical Distribution: A Supplement to Recent Surveys*. International Statistical Review/ Revue Internationale de Statistique vol.42, no. 1.
- [54] Kundu, D. and Howlader H., (2010). *Bayesian inference and prediction of the inverse Weibull distribution for Type II censored data*. Comput Stat Data Anal, 54:1547-1558.
- [55] Kundu, D., and Gupta, R. D. (1999). *Generalized exponential distribution*. Aust NZJ Stat, 41, 173-188.
- [56] Leahu, A., Munteanu, B. G. and Catarancius, S., (2013). *On the lifetime as the maximum or minimum of the sample with power series distributed size*.
- [57] Leone, L., Salvatore, F., Olivero, E. and Hamma A., (2022). *Stabilizer Rényi Entropy*. Review Letters, 128(5), 4047-4066.
- [58] Louzada, F., Bereta, E. M. P. and Franco, M. A. P., (2012). *On the distribution of the minimum or maximum of a random number of i.i.d lifetime random variable*. Appl. Math., 3, 350-353.
- [59] Lorenz, M., (1905). *Methods of measuring concentration of wealth*. Publications of the American Statistical Association, 9(70), 209-219.
- [60] Mahmoudi, E. and Jafari, A. A., (2012). *Generalized Exponential Power Series Distributions*. Computational Statistics & Data Analysis, 56, 4047-4066.

- [61] Makubate, B., Oluyede, B. and Gabanakgosi, M. (2019). *Dagum Power Series Class of Distribution with Applications to Lifetime Data*. Research and Reviews: Journal of Statistics and Mathematical Sciences, 92, 76-97.
- [62] Makubate, B., Oluyede, B. O., Dingalo, N., and Fagbamigbe, A.F. (2018). *The Beta Log-logistic Weibull Distribution: Model, Properties and Application*. International Journal of Statistics and Probability, 7(6), 49-67
- [63] Makubate, B., Oluyede, B. O., Motobetso, G., Huang, S., and Fagbamigbe, A.F. (2017). *The Beta Weibull-G Family of Distributions: Model, Properties and Application* International Journal of Statistics and Probability, 7(2), 12-32
- [64] Marshall, N. A. and Olkin, I., (1997). *A New Method for Adding a Parameter to a Family of Distributions with Applications to the Exponential and Weibull Families*. Biometrika, 84, 641-652.
- [65] Maydeu-Olivares, A. and Forero, Carlos., (2010). *Goodness-of-Fit Testing* International Encyclopedia of Education, 190-196.
- [66] McCool, J.I., (1974). Inferential Techniques for Weibull Populations. Aerospace Research Laboratories Report ARL TR74-0180. Wright-Patterson Air Force Base, Dayton, OH.
- [67] Mead, M.E., (2016). On five-parameter Lomax distribution: properties and applications. Pakistan Journal of Statistics and Operations Research, 1, 185-199.
- [68] Moakofi, T., Oluyede, B. and Chipepa, F., (2021). *Type II Exponentiated Half Logistic-Topp-Leone-G Power Series*. Pakistan Journal of Statistics and Operation Research, 17(4), 885-909.
- [69] Momeni, F. (2011). *The Generalized Power Series Distribution and Their Application*. The Journal of Mathematics and Computer Science, 2(4), 691-697.
- [70] Mudholkar, G. S., and Srivastava, D. K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*. IEEE transactions on reliability, 42(2), 299-302.
- [71] Mukherjee, I., Maiti, S. S., and Singh, V. V. (2020). *Study on estimators of the PDF and CDF of the one parameter polynomial exponential distribution*. arXiv preprint arXiv: 2006.06272
- [72] Nadarajah, S., and Kotz, S., (2003). *Moments of Some J-Shaped Distributions*. Journal of Applied Statistics, 6(30), 311 - 317.
- [73] Nadarajah, S., and Kotz, S., (2006). *The exponentiated type distributions*. Acta Applicandae Mathematica, 92(2), 97-111.
- [74] Nadarajah, S., Cordeiro, G. M. and Ortega, E. M. M., (2012). *General results for the Kumaraswamy-G distribution*. J. Stat. Comput. Simul., 82, 951-979.
- [75] Natal, J., Avila, I., Tsukahara, V. B., Pinheiro, M. and Maciel, C.D., (2021). *Entropy: From Thermodynamics to Information Processing*. Entropy (Basel), 23(10), 1340.

- [76] Oluyede, B. O., Mdlongwa, P., Makubate, B. and Huang, S., (2019). *The Burr-Weibull Power Series Class of Distributions*. Austrian Journal of Statistics, 48, 1 - 13.
- [77] Oluyede, B., and Yang, T., (2015). *A new class of generalized Lindley distribution with applications*. Journal of Statistical Computation and Simulation, 85(10), 2072-2100.
- [78] Oluyede, B., Chipepa, F. and Wankundu, D., (2021). *Exponentiated Half-Logistic-Power Generalized Weibull-G Family of Distributions: Model, Properties and Applications*. Eurasian Bulletin of Mathematics, (ISSN: 2687-5632), 3(3), 134-161.
- [79] Pinho, L. G. B., Cordeiro, G. M. and Nobre, J. S., (2012). *The Gamma-Exponentiated Weibull Distribution*. Journal of Statistical Theory and Applications, 11(4) , 379 - 395.
- [80] Percontini, A., Blas, B., and Cordeiro, G. M. (2013). *The beta weibull poisson distribution*. Chilean journal of Statistics, 4(2), 3-26.
- [81] Rényi, A., (1960). *On Measures of Entropy and Information*. Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, 547-561.
- [82] Ristic, M. and Balakrishnan, N., (2012). *The Gamma Exponentiated Exponential Distributions*. Journal of Statistical Computation and Simulation, 82(8) , 1191 - 1206.
- [83] Sangsanit, Y. and Bodhisuwan, W. (2016). *The Topp-Leone Generator of Distributions: Properties and Inferences*. Songklanakarin Journal of Science and Technology, 38(5), 537-548.
- [84] Sebastian, N., Joseph, J., and Princy, T. (2022). *Type 1 Topp-Leone q -Exponential Distribution and its Applications*. Reliability: Theory & Applications, 17(3 (69)), 361-375.
- [85] Shannon, C. E., (1951). Prediction and Entropy of Printed English. The Bell System Technical Journal, 30, 50-64.
- [86] Shannon, C. E., (1948). A mathematical Theory of Communication. The Bell System Technical Journal, 20, 379-423.
- [87] Shirke, D. T., and Kakade, C. S. (2006). On exponentiated lognormal distribution. International Journal of Agricultural and Statistical Sciences, 2(2), 319-326.
- [88] Silva, R. B., Bourguignon, M., Dias, C. R. B. and Cordeiro, G.M., (2013). *The Compound Family of Extended Weibull Power Series Distributions*. Computational Statistics & Data Analysis, 58, 352-367.
- [89] Shanker, R., Fesshaye, H., and Selvaraj, S., (2015). *On Modeling of Lifetimes Data Using Exponential and Lindley Distributions*. Biometrics and Biostatistics International Journal, 2(5), 4-9.
- [90] Silva, R. B. and Cordeiro, G. M., (2014). *The Burr XII Power Series Distributions: A New Compounding Family*. Brazilian Journal of Probability and Statistics, 29(3), 565-589.

- [91] Silva, R. B., Bourguignon, M. and Cordeiro, G.M., (2016). *A New Compounding Family of Distributions: The Generalized Gamma Power Series Distributions*. Journal of Computational and Applied Mathematics, 303, 119-139.
- [92] Soliman, A. H., Elgarhy, M. A. E. and Shakil, M., (2017). *Type II Half Logistic Family of Distributions with Applications*. Pakistan Journal of Statistics and Operation Research, 245-264.
- [93] Stephens, M. A.,(1992). *Introduction to Kolmogorov (1793) on the Empirical Determination of a Distribution In: Kotz, S., Johnson, N.L. (eds) Breakthroughs in Statistics. Springer Series in Statistics*. Springer New York. 93–105
- [94] Tahir, M.H. and Cordeiro, G.M., (2016). *Compounding of distributions: a survey and new generalized classes*. J. stat. Distrib. App., 3, 13.
- [95] Topp, C. W. and Leone, F.C., (1955). *A family of J-shaped frequency functions*. Journal of the American Statistical Association, 50 , 209 - 219.
- [96] Van der Laan, M. J. and Rose, S., (2012). *Why Machine Learning Cannot Ignore Maximum Likelihood Estimation*. arXiv preprint arXiv: 2110.12112.
- [97] Yousof, H. M., Afify, A., Nadarajah, S., Hamedani, G. and Aryal, G., (2018). *The Marshall-Olkin Generalized-G Family of Distributions with Applications*. Statistica, 78.
- [98] Yu, R. and Deng, Y., (2022). *A generalization of Rényi entropy for basic probability assignment*. Communications in Statistics-Theory and Methods, 1-18.
- [99] Zografos, K. and Balakrishnan, N., (2009). *On Families of Beta and Generalized Gamma-Generated Distribution and Associated Inference*. Statistical Methodology, 6 , 344 - 362.

Appendix

CHAPTER 2 R CODES

Pdf and Hazard Plots

```
##define TIIHLTLLLoGG cdf
TIIHLTLLLoGG_cdf <- function(theta,b,c,x){
1-((1-theta)*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))/
(1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))
}

##define TIIHLTLLLoGG pdf
TIIHLTLLLoGG_pdf <- function(theta,b,c,x){
(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*(1-theta))/
(((1-theta)*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))^(2))*
(1+(1-(1+x^c)^(-2))^(b))^(2))
}
x=seq(0,1.0,by=0.001)
y1=TIIHLTLLLoGG_pdf(0.99, 1.7, 5.3, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLTLLLoGG_pdf(0.5, 3.0, 5.9, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLLLoGG_pdf(0.8, 1.0, 2.0, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGG_pdf(0.9, 0.50, 0.5, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGG_pdf(0.9, 8.0, 12.0, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(theta,'=0.99,', b,'=1.7,', c,'=5.3')),
expression(paste(theta,'=0.5,', b,'=3.0,', c,'=5.9')),
expression(paste(theta,'=0.8,', b,'=1.0,', c,'=2.0')),
expression(paste(theta,'=0.9,', b,'=0.5,', c,'=0.5')),
expression(paste(theta,'=0.9,', b,'=8.0,', c,'=12.0'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))
```

```
##### TIIHLTLLLoGG hazard
TIIHLTLLLoGG_hazard <-function(theta,b,c,x){
  TIIHLTLLLoGG_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGG_cdf(theta,b,c,x))
}

x=seq(0,1.5,by=0.001)
y1=TIIHLTLLLoGG_hazard(0.5, 2.0, 0.02, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIIHLTLLLoGG_hazard(0.61, 0.9, 2.4, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLLLoGG_hazard(0.9, 0.08, 0.6, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGG_hazard(0.9, 3.0, 1.5, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGG_hazard(0.9, 1.0, 7.0, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.5,', b,'=2.0,', c,'=0.02')),
  expression(paste(theta,'=0.61,', b,'=0.9,', c,'=2.4')),
  expression(paste(theta,'=0.9,', b,'=0.08,', c,'=0.6')),
  expression(paste(theta,'=0.9,', b,'=3.0,', c,'=1.5')),
  expression(paste(theta,'=0.9,', b,'=1.0,', c,'=7.0'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define TIIHLTLLLoGL cdf
TIIHLTLLLoGL_cdf <- function(theta,b,c,x){
  1-log(1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))/log(1-theta)
}

##define TIIHLTLLLoGL pdf
TIIHLTLLLoGL_pdf <- function(theta,b,c,x){
  (4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*(1-theta*
  ((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))^(-1))/(-log(1-theta)*
  (1+(1-(1+x^c)^(-2))^(b))^(-2))
}

x=seq(0,1.0,by=0.001)
y1=TIIHLTLLLoGL_pdf(0.8, 1.2, 0.2, x)
plot(x,y1,ylim=c(0,2.5),col=2,'l',lwd=2,xlab="x",ylab="f(x)")
y2=TIIHLTLLLoGL_pdf(0.9, 3.0, 0.9, x)
lines(x,y2,col=3,lwd=2)
```

```

y3=TIIHLTLLLoGL_pdf(0.3, 0.9, 6.0, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGL_pdf(0.07, 1.2, 15.5, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGL_pdf(0.8, 0.9, 4.0, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.8,', b,'=1.2,', c,'=0.2')),
  expression(paste(theta,'=0.9,', b,'=3.0,', c,'=0.9')),
  expression(paste(theta,'=0.3,', b,'=0.9,', c,'=6.0')),
  expression(paste(theta,'=0.07,', b,'=1.2,', c,'=15.5')),
  expression(paste(theta,'=0.8,', b,'=0.9,', c,'=4.0'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

## define TIIHLTLLLoGL hazard
TIIHLTLLLoGL_hazard <-function(theta,b,c,x){
  TIIHLTLLLoGL_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGL_cdf(theta,b,c,x))
}

x=seq(0,2.3,by=0.001)
y1=TIIHLTLLLoGL_hazard(0.002, 0.001, 4.5, x)
plot(x,y1,ylim=c(0,7.0),col=2,'l',lwd=2,xlab="x",ylab="h(x)")
y2=TIIHLTLLLoGL_hazard(0.9, 0.7, 1.0, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLLLoGL_hazard(0.09, 0.6, 10.0, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGL_hazard(0.5, 0.5, 4.0, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGL_hazard(0.05, 7.8, 1.9, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.08,', b,'=2.0,', c,'=0.5')),
  expression(paste(theta,'=0.9,', b,'=0.7,', c,'=1.0')),
  expression(paste(theta,'=0.09,', b,'=0.6,', c,'=10.0')),
  expression(paste(theta,'=0.5,', b,'=0.5,', c,'=4.0')),
  expression(paste(theta,'=0.05,', b,'=7.8,', c,'=1.9'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

```

##define TIIHLTLLLoGP cdf
TIIHLTLLLoGP_cdf <- function(theta,b,c,x){
1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))-1)/(exp(theta)-1)
}
##define TIIHLTLLLoGP pdf
TIIHLTLLLoGP_pdf <- function(theta,b,c,x){
(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^(2))
}

```

```

x=seq(0,1.1,by=0.001)
y1=TIIHLTLLLoGP_pdf(0.8, 1.2, 0.5, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLTLLLoGP_pdf(0.01, 4.0, 4.9, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLLLoGP_pdf(2.9, 0.8, 4.9, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGP_pdf(5.8, 0.4, 4.5, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGP_pdf(0.2, 0.7, 5.9, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(theta,'=0.8,', b,'=1.2,', c,'=0.5')),
expression(paste(theta,'=0.01,', b,'=4.0,', c,'=4.9')),
expression(paste(theta,'=2.9,', b,'=0.8,', c,'=4.9')),
expression(paste(theta,'=5.8,', b,'=0.4,', c,'=4.5')),
expression(paste(theta,'=0.2,', b,'=0.7,', c,'=5.9'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

```

## define TIIHLTLLLoGP hazard
TIIHLTLLLoGP_hazard <-function(theta,b,c,x){
TIIHLTLLLoGP_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGP_cdf(theta,b,c,x))
}

```

```

x=seq(0,2.0,by=0.001)
y1=TIIHLTLLLoGP_hazard(0.001, 0.01,5.0, x)

```

```

plot(x,y1,ylim=c(0,8.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIIHLTLLLoGP_hazard(0.9, 2.0, 0.99, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLLLoGP_hazard(2.9, 1.5, 0.7, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLLLoGP_hazard(0.001, 0.3, 10.0, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLLLoGP_hazard(2.9, 2.0, 4.0, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.001,', b,'=0.01,', c,'=5.0')),
  expression(paste(theta,'=0.9,', b,'=2.0,', c,'=0.99')),
  expression(paste(theta,'=2.9,', b,'=1.5,', c,'=0.7')),
  expression(paste(theta,'=0.001,', b,'=0.3,', c,'=10.0')),
  expression(paste(theta,'=2.9,', b,'=2.0,', c,'=4.0'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define TIIHLTWLG cdf
TIIHLTWLG_cdf <- function(theta,beta,b,x){
  1- ((1-theta)*((1-(1-exp(-2*x^beta))^(b))/
  (1+(1-exp(-2*x^beta))^(b))))/
  (1-theta*((1-(1-exp(-2*x^beta))^(b))/
  (1+(1-exp(-2*x^beta))^(b))))
}

##define TIIHLTWLG pdf
TIIHLTWLG_pdf <- function(theta,beta,b,x){
  (4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  ((1-(1-exp(-2*x^beta)))^(b-1))*
  (1-theta))/(((1-theta*((1-(1-exp(-2*x^beta))^(b))/
  (1+(1-exp(-2*x^beta))^(b))))^(2))*
  (1+(1-exp(-2*x^beta))^(b))^(2))
}

x=seq(0,0.95,by=0.001)
y1=TIIHLTWLG_pdf(0.25, 2.6, 1.5, x)
plot(x,y1,ylim=c(0,2.5),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLTWLG_pdf(0.9, 0.8, 1.0, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTWLG_pdf(0.8, 2.0, 0.9, x)
lines(x,y3,col=4,lwd=2)

```

```

y4=TIHRTLWG_pdf(0.7, 9.0, 0.7, x)
lines(x,y4,col=5,lwd=2)
y5=TIHRTLWG_pdf(0.1, 39.09, 4.9, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.25,', beta,'=2.6,', b,'=1.5')),
  expression(paste(theta,'=0.9,', beta,'=0.8,', b,'=1.0')),
  expression(paste(theta,'=0.8,', beta,'=2.0,', b,'=0.9')),
  expression(paste(theta,'=0.7,', beta,'=9.0,', b,'=0.7')),
  expression(paste(theta,'=0.1,', beta,'=39.09,',b,'=4.9'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

## define TIHRTLWG hazard
TIHRTLWG_hazard <-function(theta,beta,b,x){
  TIHRTLWG_pdf(theta,beta,b,x)/(1 -TIHRTLWG_cdf(theta,beta,b,x))
}

x=seq(0,2.5,by=0.001)
y1=TIHRTLWG_hazard(0.01, 1.09, 0.01, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIHRTLWG_hazard(0.2, 0.5, 0.01, x)
lines(x,y2,col=3,lwd=2)
y3=TIHRTLWG_hazard(0.99, 1.0, 0.8, x)
lines(x,y3,col=4,lwd=2)
y4=TIHRTLWG_hazard(0.5, 1.9, 1.9, x)
lines(x,y4,col=5,lwd=2)
y5=TIHRTLWG_hazard(0.99, 1.0, 1.6, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.01,', beta,'=1.09,', b,'=0.01')),
  expression(paste(theta,'=0.2,', beta,'=0.5,', b,'=0.01')),
  expression(paste(theta,'=0.99,', beta,'=1.0,', b,'=0.8')),
  expression(paste(theta,'=0.5,', beta,'=1.9,', b,'=1.9')),
  expression(paste(theta,'=0.99,', beta,'=1.0,',b,'=1.6'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define TIHRTLWL cdf
TIHRTLWL_cdf <- function(theta,beta,b,x){

```

```

1-log(1-theta*((1-(1-exp(-2*x^beta))^(b))/(1+(1-exp(-2*x^beta))^(b))))/log(1-theta)
}
##define TIIHLTLWL pdf
TIIHLTLWL_pdf <- function(theta,beta,b,x){
(4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*((1-(1-exp(-2*x^beta)))^(b-1))*
(1-theta)*(1-theta*((1-(1-exp(-2*x^beta))^(b)))/
(1+(1-exp(-2*x^beta))^(b))))^(-1))/(-log(1-theta)*(1+(1-exp(-2*x^beta))^(b))^(2))

}

```

```

x=seq(0,1.2,by=0.001)
y1=TIIHLTLWL_pdf(0.1, 10.2, 0.36, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLTLWL_pdf(0.6, 6.7, 6.9, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLWL_pdf(0.3, 0.9, 9.0, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLWL_pdf(0.09, 50.0, 0.01, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLWL_pdf(0.3, 1.9, 1.1, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.1,', beta,'=10.2,', b,'=0.36')),
  expression(paste(theta,'=0.6,', beta,'=6.7,', b,'=6.9')),
  expression(paste(theta,'=0.3,', beta,'=0.9,', b,'=9.0')),
  expression(paste(theta,'=0.09,', beta,'=50.0,', b,'=0.01')),
  expression(paste(theta,'=0.3,', beta,'=1.9,', b,'=1.1'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

```

## define TIIHLTLWL hazard
TIIHLTLWL_hazard <-function(theta,beta,b,x){
  TIIHLTLWL_pdf(theta,beta,b,x)/(1 -TIIHLTLWL_cdf(theta,beta,b,x))
}

```

```

x=seq(0,2.0,by=0.001)
y1=TIIHLTLWL_hazard(0.01, 5.4, 1.0, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIIHLTLWL_hazard(0.1, 2.5, 0.9, x)

```

```

lines(x,y2,col=3,lwd=2)
y3=TIIHRTLWL_hazard(0.9, 0.5, 0.2, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHRTLWL_hazard(0.2, 0.08, 0.03, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHRTLWL_hazard(0.5, 2.0, 1.6, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.01,', beta,'=5.4,', b,'=1.0')),
  expression(paste(theta,'=0.1,', beta,'=2.5,', b,'=0.9')),
  expression(paste(theta,'=0.9,', beta,'=0.5,', b,'=0.2')),
  expression(paste(theta,'=0.2,', beta,'=0.08,', b,'=0.03')),
  expression(paste(theta,'=0.5,', beta,'=2.0,', b,'=1.6'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define TIIHRTLWP cdf
TIIHRTLWP_cdf <- function(theta,beta,b,x){
  1-(exp(theta*((1-(1-exp(-2*x^beta))^(b))/(1+(1-exp(-2*x^beta))^(b))))-1)/(exp(theta)-1)
}

##define TIIHRTLWP pdf
TIIHRTLWP_pdf <- function(theta,beta,b,x){
  (4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  (1-(1-exp(-2*x^beta)))^(b-1))/((exp(theta)-1)*(1+(1-exp(-2*x^beta))^(b))^(2))
}

x=seq(0,1.2,by=0.001)
y1=TIIHRTLWP_pdf(0.01, 0.5, 3.0, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHRTLWP_pdf(0.1, 3.6, 1.2, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHRTLWP_pdf(0.09, 60.6, 0.05, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHRTLWP_pdf(1.9, 10.9, 1.8, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHRTLWP_pdf(0.6, 1.1, 2.7, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(theta,'=0.01,', beta,'=0.5,', b,'=3.0')),
  expression(paste(theta,'=0.1,', beta,'=3.6,', b,'=1.2')),
  expression(paste(theta,'=0.09,', beta,'=60.6,', b,'=0.05')),
  expression(paste(theta,'=1.9,', beta,'=10.9,', b,'=1.8')),
  expression(paste(theta,'=0.6,', beta,'=1.1,', b,'=2.7'))),

```

```

col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

## define TIIHLLWG hazard
TIIHLLWP_hazard <-function(theta,beta,b,x){
TIIHLLWP_pdf(theta,beta,b,x)/(1 -TIIHLLWP_cdf(theta,beta,b,x))
}

x=seq(0,1.5,by=0.001)
y1=TIIHLLWP_hazard(0.9, 2.2,6.0, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIIHLLWP_hazard(0.9,1.3,0.9, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLLWP_hazard(2.4, 0.7, 0.5, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLLWP_hazard(0.04, 0.9, 2.0, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLLWP_hazard(1.5, 1.2, 1.6, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(theta,'=0.9,', beta,'=2.2,', b,'=6.0')),
expression(paste(theta,'=0.9,', beta,'=1.3,', b,'=0.9')),
expression(paste(theta,'=2.4,', beta,'=0.7,', b,'=0.5')),
expression(paste(theta,'=0.04,', beta,'=0.9,', b,'=2.0')),
expression(paste(theta,'=1.5,', beta,'=1.2,', b,'=1.6'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

Quantiles, Moments and Simulation

```

##TIIHLLLoG

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

```

```

##define TIIHLTLLLoG pdf
TIIHLTLLLoG_pdf <- function(theta,b,c,x){
  (4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
  exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))))/
  ((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^(2))
}

##define TIIHLTLLLoG cdf
TIIHLTLLLoG_cdf <- function(theta,b,c,x){
  1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))-1)/(exp(theta)-1)
}

## define TIIHLTLLLoG hazard
TIIHLTLLLoG_hazard <-function(theta,b,c,x){
  TIIHLTLLLoG_pdf(theta,b,c,x)/(1 -TIIHLTLLLoG_cdf(theta,b,c,x))
}

TIIHLTLLLoG_neglogl=function(theta,b,c,x){
  -sum(log(TIIHLTLLLoG_pdf(theta,b,c,x)))
}

TIIHLTLLLoG_hazard2=function(theta,b,c,x){
  f=TIIHLTLLLoG_pdf(theta,b,c,x)
  F=TIIHLTLLLoG_cdf(theta,b,c,x)
  y=f/(1-F)
  return(y)}

#-----
TIIHLTLLLoG_quantile=function(parameter,u){
  theta=parameter[1]
  b=parameter[2]
  c=parameter[3]
  f=function(x){
    TIIHLTLLLoG_cdf(theta,b,c,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}

```

```

TIIHLTLLLoG_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,function(x)
  {paste0(' ',paste0(x,collapse=', '),",")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHLTLLLoG_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

```

#-----

table of quantile

```

parameter_matrix=as.matrix(rbind(
  par1=c(0.1,1.4,0.8),
  par2=c(0.5,1.4,1.1),
  par3=c(0.9,0.7,1.8),
  par4=c(0.3,1.5,0.9),
  par5=c(0.6,1.2,1.4)

```

))

```

TIIHLTLLLoG_QuantileTable(parameter_matrix)
sink("C:/Users/RK15000146/Desktop/
TIIHLTL-GPS/TIIHLTLLLoG_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLLLoG_QuantileTable(parameter_matrix))
sink()

```

#-----

```

TIIHLTLLLoG_moments=function(parameter,k){
  theta=parameter[1]
  b=parameter[2]

```

```

c=parameter[3]

f=function(x){
  (x^k)*(TIIHLTLLLoG_pdf(theta,b,c,x))
}
y=integrate(f,lower=0,upper=1,subdivisions = 10000000)
return(y$value)
}

TIIHLTLLLoG_MomentsTable=function(parameter_matrix,K){
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
  Table_Moments=matrix(NA,nrow=length(list),ncol=dim(parameter_matrix)[1])
  row.names(Table_Moments)=list
  colnames(Table_Moments)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Moments
  for(iter in 1:dim(parameter_matrix)[1]){
    parameter=parameter_matrix[iter,]
    for(i in 1:K){
      Table_Moments[i,iter]=TIIHLTLLLoG_moments(parameter,i)
    }
    Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]-Table_Moments
    ['EX1',iter]^2)
    Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/Table_Moments
    ['EX1',iter]^2-1)
    Table_Moments['CS',iter]=(Table_Moments['EX3',iter]-3*Table_Moments
    ['EX1',iter]*Table_Moments
    ['EX2',iter]+2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
    -Table_Moments['EX1',iter]^2)^(3/2))
    Table_Moments['CK',iter]=(Table_Moments['EX4',iter]-4*Table_Moments
    ['EX1',iter]*
    Table_Moments
    ['EX3',iter]+6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]-3*
    Table_Moments
    ['EX1',iter]^4)/((Table_Moments['EX2',iter]-Table_Moments['EX1',iter]^2)^2)
  }
  return(Table_Moments)
}

```

```

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(1.3,0.6,0.8),
  par2=c(1.0,0.8,1.2),
  par3=c(1.2,0.3,1.5),
  par4=c(1.8,2.5,0.7),
  par5=c(0.2,2.1,1.4)

))
K=5
TIIHLTLLLoG_MomentsTable(parameter_matrix,K)

sink("C:/Users/RK15000146/Desktop
/TIIHLTL-GPS/TIIHLTLLLoG_moment_table.txt")
print(parameter_matrix)
print(TIIHLTLLLoG_MomentsTable(parameter_matrix,K))
sink()

#-----

TIIHLTLLLoG_simulation=function(size=c(25,50,100,200,400),samp,par){
  set.seed(1000)
  K=length(par)
  Mean=vector()
  RMSE=vector()
  Bias=vector()
  samplesize=as.vector(t(mapply(rep,size,K)))
  for(iter_size in 1:length(size)){
    coef1=matrix(NA,samp,K)
    colnames(coef1)=c('theta','b','c')
    for(nsamp in 1:samp){
      tryCatch(
        {
          x1_TIIHLTLLLoG=NULL
          q=runif(size[iter_size],0,1)
          x1=sapply(q,TIIHLTLLLoG_quantile,parameter=par)
          # MLE
          x1_TIIHLTLLLoG=mle2(TIIHLTLLLoG_neglogl,
            start=list(theta=par[1],b=par[2],c=par[3]),method="L-BFGS

```

```

        -B",data=list(x=x1),lower=c(theta=0,b=0,c=0),
        upper=c(theta=Inf,b=Inf,c=Inf),use.ginv=TRUE)
        coef1[nsamp,]=coef(x1_TIIHLTLLoG)
    },error=function(e){}
)
}
Mean[length(size)*(0:(K-1))+iter_size]=apply
(coef1,2,mean,na.rm=TRUE)
RMSE[length(size)*(0:(K-1))+iter_size]=apply
((coef1-matrix(rep(par,nsamp),ncol=K,byrow=T))^2,2,
function(x){sqrt(mean(x,na.rm=TRUE))})
}
Bias=as.vector(sapply(1:K,function(x)
{Bias[(length(size)*(x-1)+1):
(length(size)*x)]=Mean[(length(size)*(x-1)+1):
(length(size)*x)]-par[x]}))
return(cbind(samplesize,Mean,RMSE,Bias))
}

```

```

#-----
sink("C:/Users/RK15000146/Desktop/TIIHLTL-GPS/
TIIHLTLLoG_simulation_full9.txt",append=F,split=F)
par=c(0.8,0.5)
par_p=c(0.4)
for(i in 1:length(par)){
  for(j in 1:length(par)){
    for(p in 1:length(par)){
      for(q in 1:length(par_p)){
        print(c(par[i],par[j],par[p]))
        print(TIIHLTLLoG_simulation
(samp=1000,par=c(par[i],par[j],par[p])))
      }
    }
  }
}
}

```

```

sink()

```

```

##TIIHLTL_LLoGP

```

```

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

TIIHLTLLLoGP_cdf <- function(theta,b,c,x){
1 - (exp(theta*((1-(1-(1+x^c)^(-2)))^(b)))/
(1+(1-(1+x^c)^(-2)))^(b))))-1)/(exp(theta)-1)
}
##define TIIHLTLLLoGP pdf
TIIHLTLLLoGP_pdf <- function(theta,b,c,x){
(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2)))^(b-1))
*exp(theta*((1-(1-(1+x^c)^(-2)))^(b))/(1+(1-(1+x^c)^(-2)))^(b)))))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2)))^(b))^(2))
}

## define TIIHLTLLLoGP hazard
TIIHLTLLLoGP_hazard <-function(theta,b,c,x){
  TIIHLTLLLoGP_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGP_cdf(theta,b,c,x))
}

TIIHLTLLLoGP_neglogl=function(theta,b,c,x){
  -sum(log(TIIHLTLLLoGP_pdf(theta,b,c,x)))
}

TIIHLTLLLoGP_hazard2=function(theta,b,c,x){
  f=TIIHLTLLLoGP_pdf(theta,b,c,x)
  F=TIIHLTLLLoGP_cdf(theta,b,c,x)
  y=f/(1-F)
  return(y)}

#-----
TIIHLTLLLoGP_quantile=function(parameter,u){
  theta=parameter[1]

```

```

b=parameter[2]
c=parameter[3]
f=function(x){
TIIHLTLLLoGP_cdf(theta,b,c,x)-u
}
x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
return(x)
}

TIIHLTLLLoGP_QuantileTable=function(parameter_matrix){
u=seq(0.1,0.9,0.1)
size=dim(parameter_matrix)[1]
Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
row.names(Table_Quantile)=u
colnames(Table_Quantile)=apply(parameter_matrix,1,
function(x){paste0('(',paste0(x,collapse=','),")")})
Table_Quantile
for(iter in 1:size){
parameter=parameter_matrix[iter,]
for(i in 1:length(u)){
Table_Quantile[i,iter]=TIIHLTLLLoGP_quantile(parameter,u[i])
}
}
return(Table_Quantile)
}

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
par1=c(1.2,1.2,1.7),
par2=c(0.9,1.0,0.8),
par3=c(0.9,0.7,1.8),
par4=c(0.7,0.7,0.9),
par5=c(0.5,0.6,0.9)

))
TIIHLTLLLoGP_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLLLoGP_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLLLoGP_QuantileTable(parameter_matrix))

```

```

sink()

##TIIHLTL_WP

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

##define TIIHLTL_WP pdf
TIIHLTLWP_pdf <- function(theta,beta,b,x){
  (4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*(1-(1-exp(-2*x^beta)))^(b-1))/
  ((exp(theta)-1)*(1+(1-exp(-2*x^beta))^(b))^(2))
}

##define TIIHLTL_WP cdf
TIIHLTLWP_cdf <- function(theta,beta,b,x){
  1-(exp(theta*((1-(1-exp(-2*x^beta))^(b))/(1+(1-exp(-2*x^beta))^(b))))-1)/(exp(theta)-1)
}

## define TIIHLTL_WP hazard
TIIHLTLWP_hazard <-function(theta,beta,b,x){
  TIIHLTLWP_pdf(theta,beta,b,x)/(1 -TIIHLTLWP_cdf(theta,beta,b,x))
}

TIIHLTLWP_neglogl=function(theta,beta,b,x){
  -sum(log(TIIHLTLWP_pdf(theta,beta,b,x)))
}

TIIHLTLWP_hazard2=function(theta,beta,b,x){
  f=TIIHLTLWP_pdf(theta,beta,b,x)
  F=TIIHLTLWP_cdf(theta,beta,b,x)
  y=f/(1-F)
  return(y)}

```

```

#-----
TIIHRTLWP_quantile=function(parameter,u){
  theta=parameter[1]
  beta=parameter[2]
  b=parameter[3]
  f=function(x){
    TIIHRTLWP_cdf(theta,beta,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}

TIIHRTLWP_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHRTLWP_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(0.1,1.4,0.8),
  par2=c(0.5,1.4,1.1),
  par3=c(0.9,0.7,1.8),
  par4=c(0.3,1.5,0.9),
  par5=c(0.6,1.2,1.4)

))
TIIHRTLWP_QuantileTable(parameter_matrix)

```

```

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHRTLWP_quantile_table.txt")
print(parameter_matrix)
print(TIIHRTLWP_QuantileTable(parameter_matrix))
sink()

```

#-----

```

TIIHRTLWP_moments=function(parameter,k){
  theta=parameter[1]
  beta=parameter[2]
  b=parameter[3]

  f=function(x){
    (x^k)*(TIIHRTLWP_pdf(theta,beta,b,x))
  }
  y=integrate(f,lower=0,upper=1,subdivisions = 10000000)
  return(y$value)
}

```

```

TIIHRTLWP_MomentsTable=function(parameter_matrix,K){
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
  Table_Moments=matrix(NA,nrow=length(list),
    ncol=dim(parameter_matrix)[1])
  row.names(Table_Moments)=list
  colnames(Table_Moments)=apply(parameter_matrix,1,
    function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Moments
  for(iter in 1:dim(parameter_matrix)[1]){
    parameter=parameter_matrix[iter,]
    for(i in 1:K){
      Table_Moments[i,iter]=TIIHRTLWP_moments(parameter,i)
    }
    Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)
    Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/
      Table_Moments['EX1',iter]^2-1)
    Table_Moments['CS',iter]=(Table_Moments['EX3',iter]
      -3*Table_Moments['EX1',iter]*Table_Moments['EX2',iter]+

```

```

      2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)^(3/2))
      Table_Moments['CK',iter]=(Table_Moments['EX4',iter]
      -4*Table_Moments['EX1',iter]*Table_Moments['EX3',iter]
      + 6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]
      -3*Table_Moments['EX1',iter]^4)/
      ((Table_Moments['EX2',iter]-Table_Moments['EX1',iter]^2)^2)
    }
    return(Table_Moments)
  }

```

```

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(1.3,0.6,0.8),
  par2=c(1.0,0.8,1.2),
  par3=c(1.2,0.3,1.5),
  par4=c(1.8,2.5,0.7),
  par5=c(0.2,2.1,1.0)

))
K=5
TIIHLTLWP_MomentsTable(parameter_matrix,K)

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLWP_moment_table.txt")
print(parameter_matrix)
print(TIIHLTLWP_MomentsTable(parameter_matrix,K))
sink()
-----

```

```

##TIIHLTL_WL

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

```

```

##define TIIHLTLWG pdf
TIIHLTLWL_pdf <- function(theta,beta,b,x){
  (4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  ((1-(1-exp(-2*x^beta)))^(b-1))
  *(1-theta)*(1-theta*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))^(-1))/(-log(1-theta)*
  (1+(1-exp(-2*x^beta))^(b))^2))

}

##defineTIIHLTLWG cdf
TIIHLTLWL_cdf <- function(theta,beta,b,x){
  1-log(1-theta*((1-(1-exp(-2*x^beta))^(b))/(1+(1-exp(-2*x^beta))^(b))))/log(1-theta)
}

## define TIIHLTLWG hazard
TIIHLTLWL_hazard <-function(theta,beta,b,x){
  TIIHLTLWL_pdf(theta,beta,b,x)/(1 -TIIHLTLWL_cdf(theta,beta,b,x))
}

TIIHLTLWL_neglogl=function(theta,beta,b,x){
  -sum(log(TIIHLTLWL_pdf(theta,beta,b,x)))
}

TIIHLTLWL_hazard2=function(theta,beta,b,x){
  f=TIIHLTLWL_pdf(theta,beta,b,x)
  F=TIIHLTLWL_cdf(theta,beta,b,x)
  y=f/(1-F)
  return(y)}

#-----
TIIHLTLWL_quantile=function(parameter,u){
  theta=parameter[1]
  beta=parameter[2]
  b=parameter[3]
  f=function(x){
    TIIHLTLWL_cdf(theta,beta,b,x)-u

```

```

}
x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
return(x)
}

TIIHLTLWL_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1
  ,function(x){paste0('(',paste0(x,collapse=','),')')})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHLTLWL_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(0.1,1.4,0.8),
  par2=c(0.5,1.4,1.1),
  par3=c(0.9,0.7,1.8),
  par4=c(0.3,1.5,0.9),
  par5=c(0.6,1.2,1.4)

))
TIIHLTLWL_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLWL_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLWL_QuantileTable(parameter_matrix))
sink()
#-----

TIIHLTLWL_moments=function(parameter,k){

```

```

theta=parameter[1]
beta=parameter[2]
b=parameter[3]

f=function(x){
  (x^k)*(TIIHLTLWL_pdf(theta,beta,b,x))
}
y=integrate(f,lower=0,upper=1, subdivisions=10000000)
return(y$value)
}

TIIHLTLWL_MomentsTable=function(parameter_matrix,K){
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
  Table_Moments=matrix(NA,nrow=length(list),
    ncol=dim(parameter_matrix)[1])
  row.names(Table_Moments)=list
  colnames(Table_Moments)=apply(parameter_matrix,1
    ,function(x){paste0('(',paste0(x,collapse=',',''),")")})
  Table_Moments
  for(iter in 1:dim(parameter_matrix)[1]){
    parameter=parameter_matrix[iter,]
    for(i in 1:K){
      Table_Moments[i,iter]=TIIHLTLWL_moments(parameter,i)
    }
    Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)
    Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/
      Table_Moments['EX1',iter]^2-1)
    Table_Moments['CS',iter]=(Table_Moments['EX3',iter]
      -3*Table_Moments['EX1',iter]*Table_Moments['EX2',iter]
      +2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)^(3/2))
    Table_Moments['CK',iter]=(Table_Moments['EX4',iter]
      -4*Table_Moments['EX1',iter]*Table_Moments['EX3',iter]
      +6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]
      -3*Table_Moments['EX1',iter]^4)/
      ((Table_Moments['EX2',iter]-Table_Moments['EX1',iter]^2)^2)
  }
  return(Table_Moments)
}

```

```

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(0.3,0.5,0.4),
  par2=c(0.2,0.3,0.8),
  par3=c(0.5,0.8,0.6),
  par4=c(0.5,0.5,0.7),
  par5=c(0.3,0.8,0.2)

))
K=5
TIIHLTLWL_MomentsTable(parameter_matrix,K)

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLWL_moment_table.txt")
print(parameter_matrix)
print(TIIHLTLWL_MomentsTable(parameter_matrix,K))
sink()

##TIIHLTL_WG

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

##define TIIHLTLWG pdf
TIIHLTLWG_pdf <- function(theta,beta,b,x){
(4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
((1-(1-exp(-2*x^beta)))^(b-1))*(1-theta))/
(((1-theta*((1-(1-exp(-2*x^beta)))^(b)))/
(1+(1-exp(-2*x^beta))^(b))))^(2))*
(1+(1-exp(-2*x^beta))^(b))^(2))

}

##defineTIIHLTLWG cdf

```

```

TIIHRTLWG_cdf <- function(theta,beta,b,x){
1- (((1-theta)*((1-(1-exp(-2*x^beta))^(b)))/
(1+(1-exp(-2*x^beta))^(b))))/
(1-theta*(((1-(1-exp(-2*x^beta))^(b)))/
(1+(1-exp(-2*x^beta))^(b))))))
}

## define TIIHRTLWG hazard
TIIHRTLWG_hazard <-function(theta,beta,b,x){
  TIIHRTLWG_pdf(theta,beta,b,x)/(1 -TIIHRTLWG_cdf(theta,beta,b,x))
}

TIIHRTLWG_neglogl=function(theta,beta,b,x){
  -sum(log(TIIHRTLWG_pdf(theta,beta,b,x)))
}

TIIHRTLWG_hazard2=function(theta,beta,b,x){
  f=TIIHRTLWG_pdf(theta,beta,b,x)
  F=TIIHRTLWG_cdf(theta,beta,b,x)
  y=f/(1-F)
  return(y)}

#-----
TIIHRTLWG_quantile=function(parameter,u){
  theta=parameter[1]
  beta=parameter[2]
  b=parameter[3]
  f=function(x){
    TIIHRTLWG_cdf(theta,beta,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}

TIIHRTLWG_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)

```

```

size=dim(parameter_matrix)[1]
Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
row.names(Table_Quantile)=u
colnames(Table_Quantile)=apply(parameter_matrix,1
,function(x){paste0('(',paste0(x,collapse=','),')')})
Table_Quantile
for(iter in 1:size){
  parameter=parameter_matrix[iter,]
  for(i in 1:length(u)){
    Table_Quantile[i,iter]=TIIHLLWG_quantile(parameter,u[i])
  }
}
return(Table_Quantile)
}

```

#-----

table of quantile

```

parameter_matrix=as.matrix(rbind(
  par1=c(0.1,1.4,0.8),
  par2=c(0.5,1.4,1.1),
  par3=c(0.9,0.7,1.8),
  par4=c(0.3,1.5,0.9),
  par5=c(0.6,1.2,1.4)

```

))

```

TIIHLLWG_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLLWG_quantile_table.txt")
print(parameter_matrix)
print(TIIHLLWG_QuantileTable(parameter_matrix))
sink()

```

#-----

```

TIIHLLWG_moments=function(parameter,k){
  theta=parameter[1]
  beta=parameter[2]
  b=parameter[3]

  f=function(x){
    (x^k)*(TIIHLLWG_pdf(theta,beta,b,x))
  }
}

```

```

}
y=integrate(f,lower=0,upper=1, subdivisions=10)
return(y$value)
}

```

```

TIIHLLWG_MomentsTable=function(parameter_matrix,K){
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
  Table_Moments=matrix(NA,nrow=length(list),
    ncol=dim(parameter_matrix)[1])
  row.names(Table_Moments)=list
  colnames(Table_Moments)=apply(parameter_matrix,1,
    function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Moments
  for(iter in 1:dim(parameter_matrix)[1]){
    parameter=parameter_matrix[iter,]
    for(i in 1:K){
      Table_Moments[i,iter]=TIIHLLWG_moments(parameter,i)
    }
    Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)
    Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/
      Table_Moments['EX1',iter]^2-1)
    Table_Moments['CS',iter]=(Table_Moments['EX3',iter]
      -3*Table_Moments['EX1',iter]*Table_Moments['EX2',iter]+
      2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
      -Table_Moments['EX1',iter]^2)^(3/2))
    Table_Moments['CK',iter]=(Table_Moments['EX4',iter]
      -4*Table_Moments['EX1',iter]*Table_Moments['EX3',iter]
      +6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]
      -3*Table_Moments['EX1',iter]^4)/
      ((Table_Moments['EX2',iter]-Table_Moments['EX1',iter]^2)^2)
  }
  return(Table_Moments)
}

```

```

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(0.3,0.5,0.4),

```

```

    par2=c(0.2,0.3,0.8),
    par3=c(0.5,0.8,0.6),
    par4=c(0.5,0.5,0.7),
    par5=c(0.3,0.8,0.2)

))
K=5
TIIHLTLWG_MomentsTable(parameter_matrix,K)

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLWG_moment_table.txt")
print(parameter_matrix)
print(TIIHLTLWG_MomentsTable(parameter_matrix,K))
sink()

##TIIHLTL_LLoGL

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

##define TIIHLTLLoGL pdf
TIIHLTLLoGL_pdf <- function(theta,b,c,x){
(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*
((1-(1+x^c)^(-2))^(b-1))*
(1-theta*((1-(1-(1+x^c)^(-2))^(b)))/
(1+(1-(1+x^c)^(-2))^(b))))^(-1))/(-log(1-theta)*
(1+(1-(1+x^c)^(-2))^(b))^(2))
}

##define TIIHLTLLoGL cdf
TIIHLTLLoGL_cdf <- function(theta,b,c,x){
1-log(1-theta*((1-(1-(1+x^c)^(-2))^(b)))/
(1+(1-(1+x^c)^(-2))^(b))))/log(1-theta)
}

```

```
## define TIIHLTLLLoGL hazard
TIIHLTLLLoGL_hazard <-function(theta,b,c,x){
  TIIHLTLLLoGL_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGL_cdf(theta,b,c,x))
}
```

```
TIIHLTLLLoGL_neglogl=function(theta,b,c,x){
  -sum(log(TIIHLTLLLoGL_pdf(theta,b,c,x)))
}
```

```
TIIHLTLLLoGL_hazard2=function(theta,b,c,x){
  f=TIIHLTLLLoGL_pdf(theta,b,c,x)
  F=TIIHLTLLLoGL_cdf(theta,b,c,x)
  y=f/(1-F)
  return(y)}
```

```
#-----
TIIHLTLLLoGL_quantile=function(parameter,u){
  theta=parameter[1]
  b=parameter[2]
  c=parameter[3]
  f=function(x){
    TIIHLTLLLoGL_cdf(theta,b,c,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}
```

```
TIIHLTLLLoGL_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
```

```

        Table_Quantile[i,iter]=TIIHLTLLLoGL_quantile(parameter,u[i])
    }
}
return(Table_Quantile)
}

```

#-----

```
## table of quantile
```

```
parameter_matrix=as.matrix(rbind(
```

```
  par1=c(0.9,1.4,0.5),
```

```
  par2=c(0.9,1.4,1.9),
```

```
  par3=c(0.9,0.7,1.8),
```

```
  par4=c(0.7,1.0,0.9),
```

```
  par5=c(0.9,1.2,0.9)
```

```
))
```

```
TIIHLTLLLoGL_QuantileTable(parameter_matrix)
```

```
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
```

```
TIIHLTLLLoGL_quantile_table.txt")
```

```
print(parameter_matrix)
```

```
print(TIIHLTLLLoGL_QuantileTable(parameter_matrix))
```

```
sink()
```

#-----

```
TIIHLTLLLoGL_moments=function(parameter,k){
```

```
  theta=parameter[1]
```

```
  b=parameter[2]
```

```
  c=parameter[3]
```

```
  f=function(x){
```

```
    (x^k)*(TIIHLTLLLoGL_pdf(theta,b,c,x))
```

```
  }
```

```
  y=integrate(f,lower=0,upper=1, subdivisions=10000000)
```

```
  return(y$value)
```

```
}
```

```
TIIHLTLLLoGL_MomentsTable=function(parameter_matrix,K){
```

```
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
```

```

Table_Moments=matrix(NA,nrow=length(list),ncol=dim(parameter_matrix)[1])
row.names(Table_Moments)=list
colnames(Table_Moments)=apply(parameter_matrix,1,
function(x){paste0('(',paste0(x,collapse=',',''),")")})
Table_Moments
for(iter in 1:dim(parameter_matrix)[1]){
  parameter=parameter_matrix[iter,]
  for(i in 1:K){
    Table_Moments[i,iter]=TIIHLTLLLoGL_moments(parameter,i)
  }
  Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]
-Table_Moments['EX1',iter]^2)
  Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/
Table_Moments['EX1',iter]^2-1)
  Table_Moments['CS',iter]=(Table_Moments['EX3',iter]
-3*Table_Moments['EX1',iter]*Table_Moments['EX2',iter]
+2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
-Table_Moments['EX1',iter]^2)^(3/2))
  Table_Moments['CK',iter]=(Table_Moments['EX4',iter]
-4*Table_Moments['EX1',iter]*Table_Moments['EX3',iter]
+6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]
-3*Table_Moments['EX1',iter]^4)/((Table_Moments['EX2',iter]
-Table_Moments['EX1',iter]^2)^2)
}
return(Table_Moments)
}

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(0.8,0.5,0.7),
  par2=c(0.2,0.3,0.8),
  par3=c(0.5,0.8,0.6),
  par4=c(0.5,0.5,0.7),
  par5=c(0.8,0.8,0.4)

))
K=5
TIIHLTLLLoGL_MomentsTable(parameter_matrix,K)

```

```

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLLLoGL_moment_table.txt")
print(parameter_matrix)
print(TIIHLTLLLoGL_MomentsTable(parameter_matrix,K))
sink()
-----

##TIIHLTL_LLoGG

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

##define TIIHLTLLLoGG pdf
TIIHLTLLLoGG_pdf <- function(theta,b,c,x){
  (4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
  (1-theta))/(((1-theta)*((1-(1-(1+x^c)^(-2))^(b)))/
  (1+(1-(1+x^c)^(-2))^(b))))^(2))*
  (1+(1-(1+x^c)^(-2))^(b))^2)
}

##define TIIHLTLLLoGG cdf
TIIHLTLLLoGG_cdf <- function(theta,b,c,x){
  1-(((1-theta)*((1-(1-(1+x^c)^(-2))^(b)))/(1+(1-(1+x^c)^(-2))^(b))))/
  (1-theta*((1-(1-(1+x^c)^(-2))^(b)))/(1+(1-(1+x^c)^(-2))^(b))))
}

## define TIIHLTLLLoGG hazard
TIIHLTLLLoGG_hazard <-function(theta,b,c,x){
  TIIHLTLLLoGG_pdf(theta,b,c,x)/(1 -TIIHLTLLLoGG_cdf(theta,b,c,x))
}

TIIHLTLLLoGG_neglogl=function(theta,b,c,x){
  -sum(log(TIIHLTLLLoGG_pdf(theta,b,c,x)))
}

```

```

TIIHLTLLLoGG_hazard2=function(theta,b,c,x){
  f=TIIHLTLLLoGG_pdf(theta,b,c,x)
  F=TIIHLTLLLoGG_cdf(theta,b,c,x)
  y=f/(1-F)
  return(y)}

#-----
TIIHLTLLLoGG_quantile=function(parameter,u){
  theta=parameter[1]
  b=parameter[2]
  c=parameter[3]
  f=function(x){
    TIIHLTLLLoGG_cdf(theta,b,c,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}

TIIHLTLLLoGG_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=','),')')})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHLTLLLoGG_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(0.6,1.4,0.8),
  par2=c(0.9,1.4,1.9),

```

```

par3=c(0.9,0.7,1.8),
par4=c(0.7,1.5,0.9),
par5=c(0.6,1.2,1.4)

))
TIIHLTLLLoGG_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLLLoGG_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLLLoGG_QuantileTable(parameter_matrix))
sink()

#-----

TIIHLTLLLoGG_moments=function(parameter,k){
  theta=parameter[1]
  b=parameter[2]
  c=parameter[3]

  f=function(x){
    (x^k)*(TIIHLTLLLoGG_pdf(theta,b,c,x))
  }
  y=integrate(f,lower=0,upper=1, subdivisions=10000000)
  return(y$value)
}

TIIHLTLLLoGG_MomentsTable=function(parameter_matrix,K){
  list=c(paste0('EX',1:K),'SD','CV','CS','CK')
  Table_Moments=matrix(NA,nrow=length(list),
ncol=dim(parameter_matrix)[1])
  row.names(Table_Moments)=list
  colnames(Table_Moments)=apply(parameter_matrix,1,
function(x){paste0(' ',paste0(x,collapse=' '),")")})
  Table_Moments
  for(iter in 1:dim(parameter_matrix)[1]){
    parameter=parameter_matrix[iter,]
    for(i in 1:K){
      Table_Moments[i,iter]=TIIHLTLLLoGG_moments(parameter,i)
    }
  }
}

```

```

    Table_Moments['SD',iter]=sqrt(Table_Moments['EX2',iter]
    -Table_Moments['EX1',iter]^2)
    Table_Moments['CV',iter]=sqrt(Table_Moments['EX2',iter]/
    Table_Moments['EX1',iter]^2-1)
    Table_Moments['CS',iter]=(Table_Moments['EX3',iter]
    -3*Table_Moments['EX1',iter]*Table_Moments['EX2',iter]
    +2*Table_Moments['EX1',iter]^3)/((Table_Moments['EX2',iter]
    -Table_Moments['EX1',iter]^2)^(3/2))
    Table_Moments['CK',iter]=(Table_Moments['EX4',iter]
    -4*Table_Moments['EX1',iter]*Table_Moments['EX3',iter]
    + 6*Table_Moments['EX1',iter]^2*Table_Moments['EX2',iter]
    -3*Table_Moments['EX1',iter]^4)/
    ((Table_Moments['EX2',iter]-Table_Moments['EX1',iter]^2)^2)
  }
  return(Table_Moments)
}

```

```

#-----
## Moments Table
parameter_matrix=as.matrix(rbind(
  par1=c(0.8,0.5,0.7),
  par2=c(0.2,0.3,0.8),
  par3=c(0.5,0.8,0.6),
  par4=c(0.5,0.5,0.7),
  par5=c(0.8,0.8,0.4)

))
K=5
TIIHLTLLLoGG_MomentsTable(parameter_matrix,K)

sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLLLoGG_moment_table1.txt")
print(parameter_matrix)
print(TIIHLTLLLoGG_MomentsTable(parameter_matrix,K))
sink()

```

Skewness and Kurtosis Plots

```

###3D plots
#####TIIHLTL_LLoGG

```

```

library(plot3Drgl)
F=function(x,par){
  theta=par[1];b =par[2];c= par[3]
  C = 1-(((1-theta)*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))/
  (1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))))

return(C)
}
#PDF function:
f = function(x,par){
  theta=par[1];b =par[2];c= par[3]
  D =(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
  (1-theta))/
  (((1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))))^(2))*
  (1+(1-(1+x^c)^(-2))^(b))^(2))

  return(D)}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
theta=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(theta,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.9,9.1,theta, b))
Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
  return(Kurtosis)}

#Skewness=skewness(U)
# return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('theta', 'b'))
Kurtosis<-outer(theta,b,G_kurt)
#Skewness<-outer(theta,b,G_kurt)
persp3D(theta,b, Kurtosis,main=expression(paste(TIIHLTL_LLoG_G
  (theta, b, 9.1))),xlab="theta",ylab="b",zlab="Kurtosis" )
#####

###TIIHLTL-LLoGL

```

```

library(plot3Drgl)
F=function(x,par){
  theta=par[1];b =par[2];c= par[3]
  C = 1-log(1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))/
  log(1-theta)

  return(C)
}
#PDF function:
f = function(x,par){
  theta=par[1];b =par[2];c= par[3]
  D =(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
  (1-theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b))))^(-1))/
  (-log(1-theta)*(1+(1-(1+x^c)^(-2))^(b))^(2))

  return(D)}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
theta=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(theta,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.4,2.6,theta, b))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
# return(Kurtosis)}

Skewness=skewness(U)
  return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('theta', 'b'))
#Kurtosis<-outer(theta,b,G_kurt)
Skewness<-outer(theta,b,G_kurt)
persp3D(theta,b, Skewness,main=expression
(paste(TIIHLTL_LLoG_L(theta, b, 2.6))),
xlab="theta",ylab="b",zlab="Skewness" )

```

#####

###TIIHLTL-LLoGP

library(plot3Drgl)

F=function(x,par){

 theta=par[1];b =par[2];c= par[3]

 C = 1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b)))/(1+(1-(1+x^c)^(-2))^(b))))-1)/
 (exp(theta)-1)

 return(C)

}

#PDF function:

f = function(x,par){

 theta=par[1];b =par[2];c= par[3]

 D =(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
 exp(theta*((1-(1-(1+x^c)^(-2))^(b)))/(1+(1-(1+x^c)^(-2))^(b))))/
 ((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^(2))

 return(D)}

#Hazard function:

h = function(x,par) f(x,par)/(1-F(x,par))

#The program is developed in R to obtain the value of skewness:

library(moments)

theta=seq(1,7,.1)

b=seq(1,7,.1)

G_kurt<- function(theta,b) {

 x <- seq(0, 10, le = 2*10^4)

 U=f(x,c(0.9,8.6,theta, b))

Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)

 return(Kurtosis)}

#Skewness=skewness(U)

return(Skewness)}

#Skew

G_kurt<-Vectorize(G_kurt, c('theta', 'b'))

Kurtosis<-outer(theta,b,G_kurt)

#Skewness<-outer(theta,b,G_kurt)

```

persp3D(theta,b, Kurtosis,main=expression
(paste(TIIHRTL_LLoG_P(theta, b, 8.6))),
xlab="theta",ylab="b",zlab="Kurtosis" )
#####

###TIIHRTL-WP
library(plot3Drgl)
F=function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  C = 1-(exp(theta*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))-1)/(exp(theta)-1)

  return(C)
}
#PDF function:
f = function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  D =(4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  (1-(1-exp(-2*x^beta)))^(b-1))/((exp(theta)-1)*
  (1+(1-exp(-2*x^beta))^(b))^(2))

  return(D)}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
theta=seq(1,7,.1)
beta=seq(1,7,.1)
G_kurt<- function(theta,beta) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.9,3.6,theta, beta))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
# return(Kurtosis)}

Skewness=skewness(U)
  return(Skewness)}

```

```

#Skew
G_kurt<-Vectorize(G_kurt, c('theta', 'beta'))
#Kurtosis<-outer(theta,beta,G_kurt)
Skewness<-outer(theta,beta,G_kurt)
persp3D(theta,beta, Skewness,main=expression
(paste(TIIHLTL_WP(theta, beta, 3.6))),
xlab="theta",ylab="beta",zlab="Skewness" )
#####

###TIIHLTL-WG
library(plot3Drgl)
F=function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  C = 1- (((1-theta)*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))/
  (1-theta*((1-(1-exp(-2*x^beta))^(b)))/(1+(1-exp(-2*x^beta))^(b))))

return(C)
}
#PDF function:
f = function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  D =(4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  (((1-(1-exp(-2*x^beta)))^(b-1))*(1-theta))/
  (((1-theta*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))^(2))*((1+(1-exp(-2*x^beta))^(b))^(2))

  return(D)}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
theta=seq(1,7,.1)
beta=seq(1,7,.1)
G_kurt<- function(theta,beta) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.9,2.6,theta, beta))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)

```

```

# return(Kurtosis)}

Skewness=skewness(U)
return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('theta', 'beta'))
#Kurtosis<-outer(theta,beta,G_kurt)
Skewness<-outer(theta,beta,G_kurt)
persp3D(theta,beta, Skewness,main=expression
(paste(TIIHRTL_WG(theta, beta, 2.6))),
xlab="theta",ylab="beta",zlab="Skewness" )

#####
###TIIHRTL-WL
library(plot3Drgl)
F=function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  C = 1-log(1-theta*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))/log(1-theta)

return(C)
}
#PDF function:
f = function(x,par){
  theta=par[1];beta =par[2];b= par[3]
  D =(4*b*theta*beta*(x^(beta-1))*exp(-2*x^beta)*
  ((1-(1-exp(-2*x^beta)))^(b-1))*(1-theta)*
  (1-theta*((1-(1-exp(-2*x^beta))^(b)))/
  (1+(1-exp(-2*x^beta))^(b))))^(-1))/
  (-log(1-theta)*(1+(1-exp(-2*x^beta))^(b))^(2))

return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
theta=seq(1,7,.1)
beta=seq(1,7,.1)
G_kurt<- function(theta,beta) {
  x <- seq(0, 10, le = 2*10^4)

```

```

    U=f(x,c(0.9,1.2,theta, beta))
Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
    return(Kurtosis)}

#Skewness=skewness(U)
#return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('theta', 'beta'))
Kurtosis<-outer(theta,beta,G_kurt)
#Skewness<-outer(theta,beta,G_kurt)
persp3D(theta,beta, Kurtosis,main=expression
(paste(TIIHLTL_WL(theta, beta, 1.2))),
xlab="theta",ylab="beta",zlab="Kurtosis" )
#####

```

Applications and Goodness of Fit

```

##### LIFE OF BEARINGS APPLICATIONS
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)

x<-c(
152.7, 172.0, 172.5, 173.3, 193.0, 204.7, 216.5, 234.9 ,262.6 ,422.6)

###TIIHLTLLLoGP
TIIHLTLLLoGP_LL<-function(theta,b,c){-sum(log((4*b*theta*c*
(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))
*exp(theta*((1-(1-(1+x^c)^(-2))^(b)))/
(1+(1-(1+x^c)^(-2))^(b)))))/
((exp(theta)-1)*
(1+(1-(1+x^c)^(-2))^(b))^(2))))}

mle.result<-mle2(TIIHLTLLLoGP_LL,hessian = NULL,
start=list(theta=10.1, b=10.01, c=0.01),
optimizer="nlminb",lower=0)
summary(mle.result)
vcov(mle.result)
output1<-vcov(mle.result)

```

```
1.96*sqrt(diag(output1))
solve(output1)
```

```
#####WP
```

```
WP_LL<-function(theta,alpha,beta){-sum(log((theta*alpha*
(beta^alpha)*(x^(alpha-1))*exp(-(beta*x)^alpha)*
exp(theta*exp(-(beta*x)^alpha)))/(exp(theta)-1)))}
mle.result<-mle2(WP_LL,hessian = NULL,
start=list(theta=10.17, alpha=0.0001, beta=0.1),
optimizer="nllminb",lower=0)
summary(mle.result)
```

```
#####PLL
```

```
PLL_LL<-function(theta,alpha,beta){-sum(log((theta*alpha*
(beta^2)*(1+x^alpha)*(x^(alpha-1))
*exp(-beta*x^alpha))/(beta+1)*
log(1-theta)*(theta*((beta+1+(beta*x^alpha))/(beta+1))*
exp(-beta*x^alpha)-1))))}
mle.result<-mle2(PLL_LL,hessian = NULL,start=list(theta=0.98901
,alpha=0.001,beta= 0.9641),optimizer="nllminb",lower=0)
summary(mle.result)
```

```
#####GoP
```

```
GoP_LL<-function(theta,beta,gamma){-sum(log((theta*beta*
exp(gamma*x)*
exp(-(beta/gamma)*(exp(gamma*x)-1))*(exp(theta*
exp(-(beta/gamma)*
(exp(gamma*x)-1)))))/(exp(theta)-1)))}
mle.result<-mle2(GoP_LL,hessian = NULL,
start=list(theta=10.187 ,beta=0.001, gamma=0.01)
,optimizer="nllminb",lower=0)
summary(mle.result)
```

```
#####TIIHLW
```

```
TIIHLW_LL<-function(lambda,delta,gamma){-sum(log((2*lambda*
delta*gamma*(x^(gamma-1))*exp(-delta*x^gamma)*
(1-exp(-delta*x^gamma))^(lambda-1))/
((1+(1-exp(-delta*x^gamma))^(lambda))^2))))}
mle.result<-mle2(TIIHLW_LL,hessian = NULL,
```

```

start=list(lambda=10.1 ,delta=0.000000000001,
gamma=0.0000000000000001),optimizer="nlminb",lower=0)
summary(mle.result)

#####TLMoLLo
TLMoLLo_LL<-function(b,delta,c){-sum(log(((2*b*(delta^2)
*c*(x^(c-1))*((1+x^c)^(-3)))/((1-(1-delta)*
(1+x^c)^(-1))^3))*1-((delta^2)*(1+x^c)^(-2))/
((1-(1-delta)*(1+x^c)^(-1))^2))^b-1)))}
mle.result<-mle2(TLMoLLo_LL,hessian = NULL,
start=list(b=10.90 ,delta=0.001,c=0.00001),
optimizer="nlminb",lower=0)

#####WINDOW OF STRENGTH APPLICATION
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)

# maximum likelihood estimators

x<-c(
18.83, 20.8, 21.657, 23.03, 23.23, 24.05, 24.321,
25.5, 25.52, 25.8, 26.69, 26.77, 26.78, 27.05, 27.67,
29.9, 31.11, 33.2, 33.73, 33.76, 33.89, 34.76, 35.75,
35.91, 36.98, 37.08, 37.09, 39.58, 44.045, 45.29, 45.381
)

###TIIHLTLLLoGP
TIIHLTLLLoGP_LL<-function(theta,b,c){-sum(log((4*b*theta*
c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^b-1))
*exp(theta*((1-(1-(1+x^c)^(-2))^b))/
(1+(1-(1+x^c)^(-2))^b))))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^b)^2))}
mle.result<-mle2(TIIHLTLLLoGP_LL,hessian = NULL,
start=list(theta=10.1, b=10.01, c=10.001),
optimizer="nlminb",lower=0)
summary(mle.result)

```

```

vcov(mle.result)
output1<-vcov(mle.result)
1.96*sqrt(diag(output1))
solve(output1)

#####WP
WP_LL<-function(theta,alpha,beta){-sum(log((theta*alpha*(beta^alpha)
*(x^(alpha-1))*exp(-(beta*x)^alpha)*exp(theta*exp(-(beta*x)^alpha)))/
(exp(theta)-1)))}
mle.result<-mle2(WP_LL,hessian = NULL,start=list
(theta=10.17, alpha=0.0001, beta=0.1),optimizer="nllminb",lower=0)
summary(mle.result)

#####PLL
PLL_LL<-function(theta,alpha,beta){-sum(log((theta*alpha*(beta^2)*
(1+x^alpha)*(x^(alpha-1))*exp(-beta*x^alpha))/((beta+1)*
log(1-theta)*(theta*((beta+1+(beta*x^alpha))/
(beta+1))*exp(-beta*x^alpha)-1))))}
mle.result<-mle2(PLL_LL,hessian = NULL,start=list
(theta=0.98901 ,alpha=0.001,beta= 0.9641),optimizer="nllminb",lower=0)
summary(mle.result)

#####GoP
GoP_LL<-function(theta,beta,gamma){-sum(log((theta*beta*exp(gamma*x)*
exp(-(beta/gamma)*(exp(gamma*x)-1))*
(exp(theta*exp(-(beta/gamma)*(exp(gamma*x)-1)))))/
(exp(theta)-1)))}
mle.result<-mle2(GoP_LL,hessian = NULL,start=list
(theta=10.187 ,beta=0.01, gamma=0.1),optimizer="nllminb",lower=0)
summary(mle.result)

#####TIIHLW
TIIHLW_LL<-function(lambda,delta,gamma){-sum(log((2*lambda*delta*
gamma*(x^(gamma-1))*exp(-delta*x^gamma)*
(1-exp(-delta*x^gamma))^(lambda-1))/
((1+(1-exp(-delta*x^gamma))^(lambda))^(2))))}
mle.result<-mle2(TIIHLW_LL,hessian = NULL,start=
list(lambda=10.1 ,delta=0.00000000000001,
gamma=0.0000000000000001),optimizer="nllminb",lower=0)
summary(mle.result)

```

```
#####TLMoLLO
TLMoLLO_LL<-function(b,delta,c){-sum(log(((2*b*(delta^2)*c
*(x^(c-1))*((1+x^c)^(-3)))/((1-(1-delta)*(1+x^c)^(-1))^3)))
*(1-((delta^2)*(1+x^c)^(-2))/
((1-(1-delta)*(1+x^c)^(-1))^2)))^(b-1) )}}
mle.result<-mle2(TLMoLLO_LL,hessian = NULL,
start=list(b=6.90 ,delta=0.1,c=0.1),
optimizer="nllminb",lower=0)
summary(mle.result)

##### LIFE OF BEARINGS GOODNESS OF FIT
library(AdequacyModel)

#####

data=c(
152.7, 172.0, 172.5, 173.3, 193.0, 204.7, 216.5, 234.9 ,262.6 ,422.6)

#####TIIHLTLLLoGP
TIIHLTLLLoGP_pdf<-function(par,x){

theta=par[1]
b=par[2]
c=par[3]

(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))*
exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^2))}

TIIHLTLLLoGP_cdf<-function(par,x){

theta=par[1]
b=par[2]
c=par[3]

1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(
(1+(1-(1+x^c)^(-2))^(b)))-1)/(exp(theta)-1))}

```

```
goodness.fit(pdf=TIIHLTLLLoGP_pdf, cdf=TIIHLTLLLoGP_cdf,
mle=c( 2.0498, 6.7579e+07, 1.6478 ),data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,10000000000000000,10000000000000000))
```

```
#####WP
```

```
WP_pdf<-function(par,x){
```

```
  theta=par[1]
```

```
  alpha=par[2]
```

```
  beta=par[3]
```

```
(theta*alpha*(beta^alpha)*(x^(alpha-1))
*exp(-(beta*x)^alpha)*
exp(theta*exp(-(beta*x)^alpha)))/(exp(theta)-1)
}
```

```
WP_cdf<-function(par,x){
```

```
  theta=par[1]
```

```
  alpha=par[2]
```

```
  beta=par[3]
```

```
1-(exp(theta*exp(-(beta*x)^alpha))-1)/(exp(theta)-1)
}
```

```
goodness.fit(pdf=WP_pdf, cdf=WP_cdf,mle=c(3.88805875,
3.91202012 , 0.00312064 ),data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
```

```
#####PLL
```

```
PLL_pdf<-function(par,x){
```

```
  theta=par[1]
```

```
  alpha=par[2]
```

```
  beta=par[3]
```

```

(theta*alpha*(beta^2)*(1+x^alpha)*(x^(alpha-1))*
exp(-beta*x^alpha))/((beta+1)*log(1-theta)*
(theta*((beta+1+(beta*x^alpha))/(beta+1))*
exp(-beta*x^alpha)-1))
}

PLL_cdf<-function(par,x){

  theta=par[1]
  alpha=par[2]
  beta=par[3]

  1-log(1-theta*((beta+1+(beta*x^alpha))/(beta+1))*exp(-beta*x^alpha))/log(1-theta)
}

goodness.fit(pdf=PLL_pdf, cdf=PLL_cdf,mle=c(7.6192e-04,
  2.1425, 1.6687e-05 ),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
  10000000000000000,10000000000000000))

#####GoP
GoP_pdf<-function(par,x){

  theta=par[1]
  beta=par[2]
  gamma=par[3]

  (theta*beta*exp(gamma*x)*exp(-(beta/gamma)*
  (exp(gamma*x)-1))*(exp(theta*exp(-(beta/gamma)*
  (exp(gamma*x)-1)))))/(exp(theta)-1)
}

GoP_cdf<-function(par,x){

  theta=par[1]
  beta=par[2]
  gamma=par[3]

  1-(exp(theta*exp(-(beta/gamma)*(exp(gamma*x)-1)))-1)/(exp(theta)-1)
}

```

```
goodness.fit(pdf=GoP_pdf, cdf=GoP_cdf,mle=c( 4.0100,
1.6327e-04 , 1.2924e-02 ),data = data,method = "BFGS",
domain = c(0,1), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
```

```
#####TIIHLW
```

```
TIIHLW_pdf<-function(par,x){
```

```
lambda=par[1]
```

```
delta=par[2]
```

```
gamma=par[3]
```

```
(2*lambda*delta*gamma*(x^(gamma-1))*
exp(-delta*x^gamma)*(1-exp(-delta*x^gamma))^(lambda-1))/
(1+(1-exp(-delta*x^gamma))^(lambda))^(2)
}
```

```
TIIHLW_cdf<-function(par,x){
```

```
lambda=par[1]
```

```
delta=par[2]
```

```
gamma=par[3]
```

```
(2*(1-exp(-delta*x^gamma))^(lambda))/(1+(1-exp(-delta*x^gamma))^(lambda))
}
```

```
goodness.fit(pdf=TIIHLW_pdf, cdf=TIIHLW_cdf,mle=c(2.3092e+04,
9.3210e-01, 4.4648e-01 ),data = data,method = "BFGS",
domain = c(0,1), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
```

```
#####TLMoLLO
```

```
TLMoLLO_pdf<-function(par,x){
```

```
b=par[1]
```

```
delta=par[2]
```

```
c=par[3]
```

```

((2*b*(delta^2)*c*(x^(c-1))*((1+x^c)^(-3)))/
((1-(1-delta)*(1+x^c)^(-1))^3))*1-((delta^2)*
(1+x^c)^(-2))/((1-(1-delta)*(1+x^c)^(-1))^2))^b-1
}

TLMoLLO_cdf<-function(par,x){

  b=par[1]
  delta=par[2]
  c=par[3]

  (1-((delta^2)*(1+x^c)^(-2))/((1-(1-delta)*(1+x^c)^(-1))^2))^b
}

goodness.fit(pdf=TLMoLLO_pdf, cdf=TLMoLLO_cdf,mle=c
(1.1095e+05, 3.2119e+03 , 2.6525e+00 ),data = data,
  method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

##### WINDOW OF STRENGTH GOODNESS OF FIT
library(AdequacyModel)

#####

data=c(
18.83, 20.8, 21.657, 23.03, 23.23, 24.05,
24.321, 25.5, 25.52, 25.8, 26.69, 26.77,
26.78, 27.05, 27.67, 29.9, 31.11, 33.2,
33.73, 33.76, 33.89, 34.76, 35.75, 35.91,
36.98, 37.08, 37.09, 39.58, 44.045, 45.29, 45.381
)

#####TIIHLTLLLoGP
TIIHLTLLLoGP_pdf<-function(par,x){

  theta=par[1]
  b=par[2]
  c=par[3]

  (4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*

```

```

((1-(1+x^c)^(-2))^(b-1))
*exp(theta*((1-(1-(1+x^c)^(-2))^(b))/
(1+(1-(1+x^c)^(-2))^(b)))))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^(2))}

TIIHLTLLLoGP_cdf<-function(par,x){

  theta=par[1]
  b=par[2]
  c=par[3]

  1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b))/
(1+(1-(1+x^c)^(-2))^(b))))-1)/(exp(theta)-1)}

goodness.fit(pdf=TIIHLTLLLoGP_pdf, cdf=TIIHLTLLLoGP_cdf,mle=c( 24.469288,
  463.265844, 0.659409 ),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
  10000000000000000,10000000000000000))

#####WP
WP_pdf<-function(par,x){

  theta=par[1]
  alpha=par[2]
  beta=par[3]

  (theta*alpha*(beta^alpha)*(x^(alpha-1))
*exp(-(beta*x)^alpha)*
exp(theta*exp(-(beta*x)^alpha)))/(exp(theta)-1)
}

WP_cdf<-function(par,x){

  theta=par[1]
  alpha=par[2]
  beta=par[3]

  1-(exp(theta*exp(-(beta*x)^alpha))-1)/(exp(theta)-1)
}

```

```
goodness.fit(pdf=WP_pdf, cdf=WP_cdf,mle=c(2.1095319,
  5.4073066 , 0.0264007 ),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
  10000000000000000,10000000000000000))
```

```
#####PLL
```

```
PLL_pdf<-function(par,x){
```

```
  theta=par[1]
```

```
  alpha=par[2]
```

```
  beta=par[3]
```

```
(theta*alpha*(beta^2)*(1+x^alpha)*(x^(alpha-1))
```

```
*exp(-beta*x^alpha))/((beta+1)*log(1-theta)*
```

```
(theta*((beta+1+(beta*x^alpha))/
```

```
(beta+1))*exp(-beta*x^alpha)-1))
```

```
}
```

```
PLL_cdf<-function(par,x){
```

```
  theta=par[1]
```

```
  alpha=par[2]
```

```
  beta=par[3]
```

```
1-log(1-theta*((beta+1+(beta*x^alpha))/
```

```
(beta+1))*exp(-beta*x^alpha))/log(1-theta)
```

```
}
```

```
goodness.fit(pdf=PLL_pdf, cdf=PLL_cdf,mle=c(4.1940e-01,
```

```
3.4318,1.1277e-05 ),data = data,method = "BFGS",
```

```
  domain = c(0,1), lim_inf = c(0,0,0,0),
```

```
  lim_sup = c(10000000000000000,
```

```
10000000000000000,10000000000000000))
```

```
#####GoP
```

```
GoP_pdf<-function(par,x){
```

```
  theta=par[1]
```

```

    beta=par[2]
    gamma=par[3]

    (theta*beta*exp(gamma*x)*exp(-(beta/gamma)*
    (exp(gamma*x)-1))*(exp(theta*exp(-(beta/gamma)*
    (exp(gamma*x)-1)))))/(exp(theta)-1)
  }

GoP_cdf<-function(par,x){

  theta=par[1]
  beta=par[2]
  gamma=par[3]

  1-(exp(theta*exp(-(beta/gamma)*(exp(gamma*x)-1)))-1)/(exp(theta)-1)
}

goodness.fit(pdf=GoP_pdf, cdf=GoP_cdf,mle=c( 1.0187e+01,
3.8385e-04 , 9.8168e-02 ),data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

#####TIIHLW
TIIHLW_pdf<-function(par,x){

  lambda=par[1]
  delta=par[2]
  gamma=par[3]

  (2*lambda*delta*gamma*(x^(gamma-1))*
  exp(-delta*x^gamma)*
  (1-exp(-delta*x^gamma))^(lambda-1))/
  (1+(1-exp(-delta*x^gamma))^(lambda))^(2)
  }

TIIHLW_cdf<-function(par,x){

  lambda=par[1]
  delta=par[2]
  gamma=par[3]

```

```
(2*(1-exp(-delta*x^gamma))^(lambda))/(1+(1-exp(-delta*x^gamma))^(lambda))
}
```

```
goodness.fit(pdf=TIIHLW_pdf, cdf=TIIHLW_cdf,mle=c(1.0189e+03,
  9.0216e-01, 6.0067e-01 ),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
  10000000000000000,10000000000000000))
```

```
#####TLMoLLo
```

```
TLMoLLo_pdf<-function(par,x){

  b=par[1]
  delta=par[2]
  c=par[3]

  ((2*b*(delta^2)*c*(x^(c-1))*((1+x^c)^(-3)))/
  ((1-(1-delta)*(1+x^c)^(-1))^(-3)))*
  (1-((delta^2)*(1+x^c)^(-2)))/
  ((1-(1-delta)*(1+x^c)^(-1))^(-2)))^(b-1)
}
```

```
TLMoLLo_cdf<-function(par,x){

  b=par[1]
  delta=par[2]
  c=par[3]

  (1-((delta^2)*(1+x^c)^(-2)))/((1-(1-delta)*(1+x^c)^(-1))^(-2)))^(b)
}
```

```
goodness.fit(pdf=TLMoLLo_pdf, cdf=TLMoLLo_cdf,mle=c(634.04444
,118.53432 , 2.42041 ),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
  10000000000000000,10000000000000000))
```

Fitted Densities and Probability Plots

```
##### LIFE OF BEARINGS FITTED DENSTIES
```

```
par(mfrow=c(0,1))
x<-c(152.7, 172.0, 172.5, 173.3, 193.0, 204.7,
```

216.5, 234.9 ,262.6, 422.6)

#####TIIHLTLLLoGP

```
TIIHLTLLLoGP<-function(theta,b,c,x) {  
  
  (4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*  
  ((1-(1+x^c)^(-2))^(b-1))*exp(theta*((1-(1-(1+x^c)^(-2))^(b))/  
  (1+(1-(1+x^c)^(-2))^(b)))))/  
  ((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^2))  
}
```

#####WP

```
WP<-function(theta,alpha,beta,x){  
  (theta*alpha*(beta^alpha)*(x^(alpha-1))*exp(-(beta*x)^alpha)  
  *exp(theta*exp(-(beta*x)^alpha)))/(exp(theta)-1)  
}
```

```
PLL<-function(theta,alpha,beta,x){  
  (theta*alpha*(beta^2)*(1+x^alpha)*(x^(alpha-1))*  
  exp(-beta*x^alpha))/((beta+1)*log(1-theta)*  
  (theta*((beta+1+(beta*x^alpha))/(beta+1))*exp(-beta*x^alpha)-1))  
}
```

#####GoP

```
GoP<-function(theta,beta,gamma,x){  
  (theta*beta*exp(gamma*x)*exp(-(beta/gamma)*  
  (exp(gamma*x)-1))*(exp(theta*exp(-(beta/gamma)*  
  (exp(gamma*x)-1)))))/(exp(theta)-1)  
}
```

#####TIIHLW

```
TIIHLW<-function(lambda,delta,gamma,x){  
  (2*lambda*delta*gamma*(x^(gamma-1))*exp(-delta*x^gamma)*  
  (1-exp(-delta*x^gamma))^(lambda-1))/  
  (1+(1-exp(-delta*x^gamma))^(lambda))^2)  
}
```

```
#####TLMoLLO
```

```
TLMoLLO<-function(b,delta,c,x){
  ((2*b*(delta^2)*c*(x^(c-1))*((1+x^c)^(-3)))/((1-(1-delta)*
  (1+x^c)^(-1))^3))*1-((delta^2)*(1+x^c)^(-2))/
  ((1-(1-delta)*(1+x^c)^(-1))^2))^b-1
}
```

```
hist(x, prob=TRUE, br=9, main='Fitted Densities',
  xlab="Life time", ylab ="Density", ylim=c(0,0.015))
```

```
curve(TIIHLTLLLoGP(x, theta=2.0498e+00,b= 6.7579e+07,
c= 1.6478e+00),lty=6, col=2, add=TRUE, lwd=2)
#par(new=TRUE)
curve(WP(x, theta=3.88805875,alpha=3.91202012 ,beta= 0.00312064),
  lty=2, col=7, add=TRUE, lwd=3)
# par(new=TRUE)
curve(PLL(x,theta=7.6192e-04 ,alpha= 2.1425e+00,
  beta=1.6687e-05 ), lty=2, col=4, add=TRUE, lwd=3)
# par(new=TRUE)
curve(GoP(x, theta=4.0100e+00, beta=1.6327e-04,
gamma=1.2924e-02 ),lty=2, col=3, add=TRUE,lwd=3)
#par(new=TRUE)
curve(TIIHLW (x,lambda= 2.3092e+04 ,delta=9.3210e-01,
gamma=4.4648e-01),lty=5, col=6, add=TRUE,lwd=3)
#par(new=TRUE)
curve(TLMoLLO(x, b=1.1095e+05, delta=3.2119e+03,
c=2.6525e+00 ),lty=2, col=1, add=TRUE,lwd=3)
#par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,c("TIIHLTL-LLoP",
" WP", "PLL", "GoP", "TIIHLW", "TL-MO-LLo"),
col=c(2,7,4,3,6,1),lwd=c(2,2,2),lty=c(6,2,2,2,2))
```

```
##### WINDOW OF STRENGTH FITTED DENSITIES
```

```

par(mfrow=c(0,1))
x<-c(
18.83, 20.8, 21.657, 23.03, 23.23, 24.05,
24.321, 25.5, 25.52, 25.8, 26.69, 26.77,
26.78, 27.05, 27.67, 29.9, 31.11, 33.2,
33.73, 33.76, 33.89, 34.76, 35.75, 35.91,
36.98, 37.08, 37.09, 39.58, 44.045, 45.29, 45.381
)

#####TIIHLTLLLoGP

TIIHLTLLLoGP<-function(theta,b,c,x) {

(4*b*theta*c*(x^(c-1))*((1+x^c)^(-3))*((1-(1+x^c)^(-2))^(b-1))
*exp(theta*((1-(1-(1+x^c)^(-2))^(b))/(1+(1-(1+x^c)^(-2))^(b)))))/
((exp(theta)-1)*(1+(1-(1+x^c)^(-2))^(b))^(2))
}

#####WP

WP<-function(theta,alpha,beta,x){
(theta*alpha*(beta^alpha)*(x^(alpha-1))*
exp(-(beta*x)^alpha)*exp(theta*exp(-(beta*x)^alpha)))/
(exp(theta)-1)
}

PLL<-function(theta,alpha,beta,x){
(theta*alpha*(beta^2)*(1+x^alpha)*(x^(alpha-1))*
exp(-beta*x^alpha))/((beta+1)*log(1-theta)*
(theta*((beta+1+(beta*x^alpha))/(beta+1))*exp(-beta*x^alpha)-1))
}

#####GoP

GoP<-function(theta,beta,gamma,x){
(theta*beta*exp(gamma*x)*exp(-(beta/gamma)*
(exp(gamma*x)-1))*(exp(theta*exp(-(beta/gamma)*
(exp(gamma*x)-1)))))/(exp(theta)-1)
}

```

```
#####TIIHLW
```

```
TIIHLW<-function(lambda,delta,gamma,x){  
  (2*lambda*delta*gamma*(x^(gamma-1))*exp(-delta*x^gamma)*  
  (1-exp(-delta*x^gamma))^(lambda-1))/(1+(1-exp(-delta*x^gamma))^(lambda))^(2)  
}
```

```
#####TLMoLLO
```

```
TLMoLLO<-function(b,delta,c,x){  
  ((2*b*(delta^2)*c*(x^(c-1))*((1+x^c)^(-3)))/  
  (((1-(1-delta)*(1+x^c)^(-1))^(3)))*(1-((delta^2)*  
  (1+x^c)^(-2))/((1-(1-delta)*(1+x^c)^(-1))^(2)))^(b-1)  
}
```

```
hist(x, prob=TRUE, br=9, main='Fitted Densities'  
,xlab="Life time", ylab="Density",ylim=c(0,0.1))
```

```
curve(TIIHLTLLLoGP(x, theta=24.468758 ,b= 463.267634  
,c= 0.659410),lty=6, col=2, add=TRUE, lwd=2)  
#par(new=TRUE)  
curve(WP(x, theta=2.1095319 ,alpha=5.4073066,  
  beta=0.0264007), lty=2, col=7, add=TRUE, lwd=3)  
# par(new=TRUE)  
curve(PLL(x,theta=4.1940e-01,alpha=3.4318e+00,  
beta=1.1277e-05), lty=2, col=4, add=TRUE, lwd=3)  
# par(new=TRUE)  
curve(GoP(x, theta=1.0187e+01 , beta=3.8385e-04,  
gamma=9.8168e-02 ),lty=2, col=3, add=TRUE,lwd=3)  
#par(new=TRUE)  
curve(TIIHLW (x,lambda= 1.0189e+03, delta=9.0216e-01,  
  gamma=6.0067e-01),lty=2, col=5, add=TRUE,lwd=3)  
#par(new=TRUE)  
curve(TLMoLLO(x, b= 634.04444, delta=118.53432,  
c= 2.42041 ),lty=2, col=1, add=TRUE,lwd=3)  
#par(new=TRUE)  
par(new=TRUE)  
par(new=TRUE)  
par(new=TRUE)  
par(new=TRUE)  
par(new=TRUE)
```

```

par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,c("TIIHLTL-LLoP","
WP","PLL","GoP","TIIHLW","TL-MO-LLo"),
col=c(2,7,4,3,5,1),lwd=c(2,2,2),lty=c(6,2,2,2))

##### LIFE OF BEARINGS PROBABILITY PLOTS
data=c(
152.7, 172.0, 172.5, 173.3, 193.0, 204.7, 216.5, 234.9 ,262.6 ,422.6)

data<-sort(data)
n=length(data)

#####TIIHLTLLLoGP
#####TIIHLTLLLoGP
theta1=2.0498e+00
b1= 6.7579e+07
c1= 1.6478e+00
#####WP
theta2= 3.88805875
alpha2= 3.91202012
beta2= 0.00312064
#####PLL
theta3=7.6192e-04
alpha3= 2.1425e+00
beta3=1.6687e-05
#####GoP
theta4=4.0100
beta4=1.6327e-04
gamma4=1.2924e-02
#####TIIHLW
lambda5= 2.3092e+04
delta5=9.3210e-01
gamma5= 4.4648e-01
#####TLMoLLo
b6= 1.1095e+05
delta6=3.2119e+03
c6= 2.6525e+00

F_observed<-c(rep(0,n))

```

```

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

require(gsl)
require(stats)

data=c(
152.7, 172.0, 172.5, 173.3, 193.0, 204.7, 216.5, 234.9 ,262.6 ,422.6)

data<-sort(data)
n=length(data)

TIIHLTLLLoGP_cdf<-function(data,theta,b,c,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1 - (exp(theta*((1-(1-(1+x^c)^(-2)))^(b)))/
      (1+(1-(1+x^c)^(-2)))^(b))))-1)/(exp(theta)-1)

  }
  return(F)
}

```

```
}
```

```
WP_cdf<-function(data,theta,alpha,beta,n){  
  F<-c(rep(0,n))  
  for (i in 1:n)  
  {  
    x<-data[i]  
    F[i]<-1-(exp(theta*exp(-(beta*x)^alpha))-1)/(exp(theta)-1)  
  }  
  return(F)  
}
```

```
PLL_cdf<-function(data,theta,alpha,beta,n){  
  F<-c(rep(0,n))  
  for (i in 1:n)  
  {  
    x<-data[i]  
    F[i]<-1-log(1-theta*((beta+1+(beta*x^alpha))/  
      (beta+1))*exp(-beta*x^alpha))/log(1-theta)  
  }  
  return(F)  
}
```

```
GoP_cdf<-function(data,theta,beta,gamma,n){  
  F<-c(rep(0,n))  
  for (i in 1:n)  
  {  
    x<-data[i]  
    F[i]<-1-(exp(theta*exp(-(beta/gamma)*  
      (exp(gamma*x)-1)))-1)/(exp(theta)-1)  
  }  
  return(F)  
}
```

```
TIIHLW_cdf<-function(data,lambda,delta,gamma,n){  
  F<-c(rep(0,n))  
  for (i in 1:n)  
  {  
    x<-data[i]  
    F[i]<- (2*(1-exp(-delta*x^gamma))^(lambda))/
```

```

      (1+(1-exp(-delta*x^gamma))^(lambda))
    }
    return(F)
  }

```

```

TLMoLLO_cdf<-function(data,b,delta,c,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-(1-((delta^2)*(1+x^c)^(-2))/
      ((1-(1-delta)*(1+x^c)^(-1))^(2)))^(b)

  }
  return(F)
}

```

```

F1<-TIIHLTLLLoGP_cdf(data,theta1,b1,c1,n)
F1
F2<-WP_cdf(data,theta2,alpha2,beta2,n)
F2
F3<-PLL_cdf(data,theta3,alpha3,beta3,n)
F3
F4<-GoP_cdf(data,theta4,beta4,gamma4,n)
F4
F5<-TIIHLW_cdf(data,lambda5,delta5,gamma5,n)
F5
F6<-TLMoLLO_cdf(data,b6,delta6,c6,n)
F6

```

```

SS<-function(F,F_line,n){
  SS<-0
  for (i in 1:n)
  {
    SS<-SS+(F[i]-F_line[i])^2
  }
  return(SS)
}

```

```

SS1<-SS(F1,F_observed,n)
SS1
SS2<-SS(F2,F_observed,n)
SS2
SS3<-SS(F3,F_observed,n)
SS3
SS4<-SS(F4,F_observed,n)
SS4
SS5<-SS(F5,F_observed,n)
SS5
SS6<-SS(F6,F_observed,n)
SS6

```

```

op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l',main="Probability Plots"
,xlab="Observed probability",ylab="Expected probability")
lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
lines(F_observed,F6,col='maroon',type='l')
legend("bottomright",cex=0.7,c("TIIHLTL-LLoP(SS=0.0206)",
"WP(SS=0.0747)", "PLL(SS=0.1133)",
"GoP(SS=0.1427)", "TIIHLW(SS=0.0261)", "TL-M0-LLo(SS=0.0253)"),
,lty=c(1,1,1,1,1,1),col=c("red","blue",
"purple","orange","green","maroon"),lwd=c(1,1,1,1,1,1))

```

WINDOW OF STRENGTH PROBABILITY PLOTS

```

data=c(
18.83, 20.8, 21.657, 23.03, 23.23,
24.05, 24.321, 25.5, 25.52, 25.8,
26.69, 26.77, 26.78, 27.05, 27.67,
29.9, 31.11, 33.2, 33.73, 33.76, 33.89,
34.76, 35.75, 35.91, 36.98, 37.08, 37.09,
39.58, 44.045, 45.29, 45.381
)

```

```

data<-sort(data)
n=length(data)

#####TIIHLTLLLoGP
#####TIIHLTLLLoGP
theta1=24.468758
b1= 463.267634
c1= 0.659410
#####WP
theta2= 2.1095319
alpha2= 5.4073066
beta2= 0.0264007
#####PLL
theta3= 4.1940e-01
alpha3= 3.4318e+00
beta3=1.1277e-05
#####GoP
theta4=1.0187e+01
beta4=3.8385e-04
gamma4=9.8168e-02
#####TIIHLW
lambda5=1.0189e+03
delta5=9.0216e-01
gamma5= 6.0067e-01
#####TLMoLLo
b6= 634.04444
delta6=118.53432
c6= 2.42041

F_observed<-c(rep(0,n))

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

require(gsl)

```

```

require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

TIIHLTLLLoGP_cdf<-function(data,theta,b,c,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1 - (exp(theta*((1-(1-(1+x^c)^(-2))^(b)))/
      (1+(1-(1+x^c)^(-2))^(b))))-1)/(exp(theta)-1)

  }
  return(F)
}

WP_cdf<-function(data,theta,alpha,beta,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1-(exp(theta*exp(-(beta*x)^alpha))-1)/(exp(theta)-1)
  }
  return(F)
}

PLL_cdf<-function(data,theta,alpha,beta,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1-log(1-theta*((beta+1+(beta*x^alpha))/

```

```

        (beta+1))*exp(-beta*x^alpha))/log(1-theta)
    }
    return(F)
}

```

```

GoP_cdf<-function(data,theta,beta,gamma,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1-(exp(theta*exp(-(beta/gamma)*
      (exp(gamma*x)-1)))-1)/(exp(theta)-1)
  }
  return(F)
}

```

```

TIIHLW_cdf<-function(data,lambda,delta,gamma,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<- (2*(1-exp(-delta*x^gamma))^(lambda))/
      (1+(1-exp(-delta*x^gamma))^(lambda))
  }
  return(F)
}

```

```

TLMoLLo_cdf<-function(data,b,delta,c,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-(1-((delta^2)*(1+x^c)^(-2))/
      ((1-(1-delta)*(1+x^c)^(-1))^(2)))^(b)

  }
  return(F)
}

```

```

F1<-TIIHLTLLLoGP_cdf(data,theta1,b1,c1,n)
F1
F2<-WP_cdf(data,theta2,alpha2,beta2,n)
F2
F3<-PLL_cdf(data,theta3,alpha3,beta3,n)
F3
F4<-GoP_cdf(data,theta4,beta4,gamma4,n)
F4
F5<-TIIHLW_cdf(data,lambda5,delta5,gamma5,n)
F5
F6<-TLMoLLo_cdf(data,b6,delta6,c6,n)
F6

SS<-function(F,F_line,n){
  SS<-0
  for (i in 1:n)
  {
    SS<-SS+(F[i]-F_line[i])^2
  }
  return(SS)
}

SS1<-SS(F1,F_observed,n)
SS1
SS2<-SS(F2,F_observed,n)
SS2
SS3<-SS(F3,F_observed,n)
SS3
SS4<-SS(F4,F_observed,n)
SS4
SS5<-SS(F5,F_observed,n)
SS5
SS6<-SS(F6,F_observed,n)
SS6

op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l',main="Probability Plots",
xlab="Observed probability",ylab="Expected probability")

```

```

lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
lines(F_observed,F6,col='maroon',type='l')
legend("bottomright",cex=0.7,c("TIIHLTL-LLoP
(SS=0.0825)", "WP(SS=0.0829)", "PLL(SS=0.1016)",
"GoP(SS=0.2267)", "TIIHLW(SS=0.0760)", "TL-MO-LLo
(SS=0.0709)"),lty=c(1,1,1,1,1,1),col=c
("red","blue","purple","orange","green","maroon")
,lwd=c(1,1,1,1,1,1))

```

CHAPTER 3 R CODES

Pdf and Hazard Plots

```

library(zipfR)

##define BTIIHLTL UNIFORM CDF AND PDF
BTIIHLTL_U_pdf<-function(alpha,beta,theta,b,x)
{((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
((1-((1-(1-(1-(x/theta))^(2))^(b)))/
(1+(1-(1-(x/theta))^(2))^(b))))^(alpha-1))*
(((1-(1-(1-(x/theta))^(2))^(b)))/
(1+(1-(1-(x/theta))^(2))^(b))))^(beta-1))*
((4*b*(1/theta)*(1-(x/theta))*
((1-(1-(x/theta))^(2))^(b-1)))/
((1+(1-(1-(x/theta))^(2))^(b))^(2))))}

BTIIHLTL_U_cdf<-function(alpha,beta,theta,b,x){(1-
((1-(1-(1-(x/theta))^(2))^(b))
/(1+(1-(1-(x/theta))^(2))^(b)))))}

x=seq(0,5.0,by=0.001)
y1=BTIIHLTL_U_pdf(0.5,1.9,4.8,3.5,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l',
lwd=2,xlab='x',ylab="pdf")
y2=BTIIHLTL_U_pdf(0.9,3.0,9.23,7.0,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTL_U_pdf(0.2,2.0,3.9,5.0,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTL_U_pdf(0.9,1.6,10.8,7.1,x)

```

```

lines(x,y4,col=5,lwd=2)
y5=BTIIHLTL_U_pdf(0.8,1.3,5.6,5.1,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=0.5,
    ',beta,'=1.9,',b,'=4.8,',theta,'=3.5')),
  expression(paste(alpha,'=0.9,
    ',beta,'=3.0,',b,'=9.23,',theta,'=7.0')),
  expression(paste(alpha,'=0.2,
    ',beta,'=2.0,',b,'=3.9,',theta,'=5.0')),
  expression(paste(alpha,'=0.9,
    ',beta,'=1.6,',b,'=10.8,',theta,'=7.1')),
  expression(paste(alpha,'=0.8,
    ',beta,'=1.3,',b,'=5.6,',theta,'=5.1'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

BTIIHLTLU_hazard<- function(alpha,beta,b,theta,x)
{BTIIHLTL_U_pdf(alpha,beta,b,theta,x)/
(1- BTIIHLTL_U_cdf(alpha,beta,b,theta,x))
}
x=seq(0,5.0,by=0.001)
y1=BTIIHLTLU_hazard(0.5,3.1,5.6,8.2,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l'
,lwd=2,xlab='x',ylab="hrf")
y2=BTIIHLTLU_hazard(0.09,0.7,4.2,9.0,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTLU_hazard(2.9,6.9,7.5,0.2,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTLU_hazard(0.01,1.11,4.1,2.16,x)
lines(x,y4,col=5,lwd=2)
y5=BTIIHLTLU_hazard(2.5,1.0,9.1,3.7,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=0.5,
    ',beta,'=3.1,',b,'=5.6,',theta,'=8.2')),
  expression(paste(alpha,'=0.09,
    ',beta,'=0.7,',b,'=4.2,',theta,'=9.0')),
  expression(paste(alpha,'=2.9,
    ',beta,'=6.9,',b,'=7.5,',theta,'=0.2')),
  expression(paste(alpha,'=0.01,
    ',beta,'=1.11,',b,'=4.1,',theta,'=2.16')),
  expression(paste(alpha,'=2.5,

```

```

',beta,'=1.0,',b,'=9.1,',theta,'=3.7'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define BTIIHLTL EXPONENTIAL CDF AND PDF

BTIIHLTL_pdf<-function(alpha,beta,lambda,b,x)
{((gamma(alpha+beta))/(gamma(alpha)*
gamma(beta)))*(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
(((1-(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x))))*
((1-exp(-(2*lambda*x)))^(b-1)))/
((1+(1-exp(-(2*lambda*x)))^(b))^(2))))}

BTIIHLTL_cdf<-function(alpha,beta,lambda,b,x){
pbeta(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
}
x=seq(0,5.0,by=0.001)
y1=BTIIHLTL_pdf(5.7,0.8,0.3,9.2,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l'
,lwd=2,xlab='x',ylab="pdf")
y2=BTIIHLTL_pdf(1.4,1.4,0.2,1.2,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTL_pdf(1.5,1.1,0.33,9.5,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTL_pdf(0.9,0.9,0.2,0.9,x)
lines(x,y4,col=5,lwd=2)
y5=BTIIHLTL_pdf(1.6,4.0,0.3,8.5,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(alpha,'=5.7,
',beta,'=0.8,',b,'=0.3,',lambda,'=9.2')),
expression(paste(alpha,'=1.4,
',beta,'=1.4,',b,'=0.2,',lambda,'=1.2')),
expression(paste(alpha,'=1.5,
',beta,'=1.1,',b,'=0.33,',lambda,'=9.5')),
expression(paste(alpha,'=0.9,
',beta,'=0.9,',b,'=0.2,',lambda,'=0.9')),
expression(paste(alpha,'=1.6,
',beta,'=4.0,',b,'=0.3,',lambda,'=8.5'))),

```

```

col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

BTIIHLTL_hazard<- function(alpha,beta,b,lambda,x)
{BTIIHLTL_pdf(alpha,beta,b,lambda,x)/
(1- BTIIHLTL_cdf(alpha,beta,b,lambda,x))
}
x=seq(0,5.0,by=0.001)
y1=BTIIHLTL_hazard(3.9,1.2,0.7,7.2,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l',
lwd=2,xlab='x',ylab="hrf")
y2=BTIIHLTL_hazard(0.3,0.20,1.06,10.9,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTL_hazard(1.0,3.2,0.05,1.11,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTL_hazard(0.07,0.09,0.09,3.03,x)
lines(x,y4,col=5,lwd=2)
y5=BTIIHLTL_hazard(1.7,0.4,0.49,7.08,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(alpha,'=3.9,'
,beta,'=1.2','b,'=0.7','lambda,'=7.2')),
expression(paste(alpha,'=0.3,'
,beta,'=0.2','b,'=1.06','lambda,'=10.9')),
expression(paste(alpha,'=1.0,'
,beta,'=3.2','b,'=0.05','lambda,'=1.11')),
expression(paste(alpha,'=0.07,
',beta,'=0.09','b,'=0.09','lambda,'=3.03')),
expression(paste(alpha,'=1.7,
',beta,'=0.4','b,'=0.49','lambda,'=7.08'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##define BTIIHLTL LOG LOGISTIC CDF, PDF AND HAZARD FUNCTIONS
BTIIHLTL_LOG_pdf<-function(alpha,beta,b,c,x)
{((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))
*(((2*(1-(1+x^(c))^(-(2)))^(b)))/
(1+(1-(1+x^(c))^(-(2)))^(b)))^(beta-1))*
((4*b*(c*x^(c-1))*((1+x^(c))^(-(3))))*
((1-(1+x^(c))^(-(2)))^(b-1)))/
((1+(1-(1+x^(c))^(-(2)))^(b))^2))}

BTIIHLTL_LOG_cdf <-function(alpha,beta,b,c,x)

```

```
{Ibeta(1-((1-(1-(1+x^(c))^(-(2))))^(b)))/
(1+(1-(1+x^(c))^(-(2))))^(b))),alpha,beta)}
```

```
x=seq(0,5.0,by=0.001)
y1=BTIIHLTL_LOG_pdf(1.5,1.5,0.6,0.9,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l',
lwd=2,xlab='x',ylab="pdf")
y2=BTIIHLTL_LOG_pdf(0.6,10.8,10.5,4.2,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTL_LOG_pdf(0.3,1.6,0.7,1.6,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTL_LOG_pdf(1.6,10.9,10.9,1.4,x)
lines(x,y4,col=5,lwd=2)
y5=BTIIHLTL_LOG_pdf(1.3,10.9,9.5,1.3,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=1.5,
    ',beta,'=1.5,',b,'=0.6,',d,'=0.9')),
  expression(paste(alpha,'=0.6,
    ',beta,'=10.8,',b,'=10.5,',d,'=4.2')),
  expression(paste(alpha,'=0.3,
    ',beta,'=1.6,',b,'=0.7,',d,'=1.6')),
  expression(paste(alpha,'=1.6,
    ',beta,'=10.9,',b,'=10.9,',d,'=1.4')),
  expression(paste(alpha,'=1.3,
    ',beta,'=10.9,',b,'=9.5,',d,'=1.3'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))
```

```
BTIIHLTL_LOG_hazard<- function(alpha,beta,b,c,x)
{BTIIHLTL_LOG_pdf(alpha,beta,b,c,x)/
(1- BTIIHLTL_LOG_cdf(alpha,beta,b,c,x))
}
x=seq(0,5.0,by=0.001)
y1=BTIIHLTL_LOG_hazard(1.0,0.25,8.5,0.6,x)
plot(x,y1,ylim=c(0,0.6),col=2,'l',
lwd=2,xlab='x',ylab="hrf")
y2=BTIIHLTL_LOG_hazard(2.0,8.9,7.6,1.3,x)
lines(x,y2,col=3,lwd=2)
y3=BTIIHLTL_LOG_hazard(0.59,9.39,2.00,2.99,x)
lines(x,y3,col=4,lwd=2)
y4=BTIIHLTL_LOG_hazard(1.4,2.9,8.9,0.69,x)
lines(x,y4,col=5,lwd=2)
```

```

y5=BTIIHRTL_LOG_hazard(4.01,3.05,0.03,5.6,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=1.0,',
    beta,'=0.25,',b,'=8.5,',d,'=0.6')),
  expression(paste(alpha,'=2.0,',
    beta,'=8.9,',b,'=7.6,',d,'=1.3')),
  expression(paste(alpha,'=0.59,',
    beta,'=9.39,',b,'=2.0,',d,'=2.99')),
  expression(paste(alpha,'=1.4,',
    beta,'=2.9,',b,'=8.9,',d,'=0.69')),
  expression(paste(alpha,'=4.01,
    ',beta,'=3.05,',b,'=0.03,',d,'=5.6'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

Quantiles, Moments and Simulation

```

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)
library(zipfR)

##define pdf BTIIHRTLLE

BTIIHRTLLE_pdf<-function(alpha,beta,lambda,b,x)
{
  ((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  (((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
  (((1-(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
  ((4*b*lambda*(exp(-(2*lambda*x))))*
  ((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b))^(2)))
}

##define cdf BTIIHRTLLE
BTIIHRTLLE_cdf=function(alpha,beta,lambda,b,x){
  y= Ibeta((1-((1-(1-exp(-(2*lambda*x)))^(b)))/

```

```

(1+(1-exp(-(2*lambda*x)))^(b))))),alpha,beta)

}

BTIIHLTLE_neglogl=function(alpha,beta,lambda,b,x){
  -sum(log(BTIIHLTLE_pdf(alpha,beta,lambda,b,x)))}

BTIIHLTLE_hazard=function(alpha,beta,lambda,b,x){
  f=BTIIHLTLE_pdf(alpha,beta,lambda,b,x)
  F=BTIIHLTLE_cdf(alpha,beta,lambda,b,x)
  y=f/(1-F)
  return(y)}

##-----
## define BTIIHLTLE quantile
BTIIHLTLE_quantile =function(parameter,u){
  alpha=parameter[1]
  beta=parameter[2]
  lambda=parameter[3]
  b=parameter[4]

  f=function(x){
    BTIIHLTLE_cdf(alpha,beta,lambda,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=10,tol=1000))
  return(x)
}

BTIIHLTLE_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1
,function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=BTIIHLTLE_quantile(parameter,u[i])
    }
  }
}

```

```

    }
  }
  return(Table_Quantile)
}
#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(1.8,1.8,1.7,1.5),
  par2=c(2.0,2.5,0.9,0.5),
  par3=c(0.5,2.2,1.5,1.5),
  par4=c(0.9,1.0,1.8,0.5),
  par5=c(0.8,1.5,1.6,2.4)

))

BTIIHLTLE_QuantileTable(parameter_matrix)
print(parameter_matrix)
print(BTIIHLTLE_QuantileTable(parameter_matrix))

##End of Quantile
#-----

##define pdf BTIIHLTLE
BTIIHLTLE_pdf=function(alpha,beta,lambda,b,x){
  y=((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  (((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
  (((1-(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
  ((4*b*lambda*(exp(-(2*lambda*x))))*
  ((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b)))^(2)))
}

BTIIHLTLE_moments=function(alpha,beta,lambda,b,n){
  f=function(alpha,beta,lambda,b,n,x){
    (x^n)*(BTIIHLTLE_pdf(alpha,beta,lambda,b,x))
  }
  y=integrate(f,lower=0,upper=1,subdivisions=10,
  alpha=alpha,beta=beta,lambda=lambda,b=b,n=n)
  return(y$value)
}

```

```

### table of moments
parameter_matrix_Moments=matrix(c(1.8,0.8,0.7,0.5,2.0,
2.5,0.9,0.5,0.5,2.0,1.5,
1.5,0.9,5.0,1.8,0.5,0.8,4.5,1.6,0.4),ncol=4,byrow=T)
parameter_matrix_Moments
list_Moments=c('EX','EX2','EX3','EX4','EX5','EX6',
'SD','CV','CS','CK')
Table_Moments=matrix(0,nrow=length(list_Moments),
ncol=dim(parameter_matrix_Moments)[1])
row.names(Table_Moments)=list_Moments
Table_Moments

for(iter in 1:dim(parameter_matrix_Moments)[1]){
  parameter=parameter_matrix_Moments[iter,]
  for(i in 1:10){
    Table_Moments[i,iter]=BTIIHLTLE_moments(parameter[1]
,parameter[2],parameter[3],parameter[4],i)
  }
}
print(Table_Moments)
###moments end

```

##-----

```

BTIIHLTLE_simulation=function(size=c(10,50,100,200,400,800),samp,par){
  set.seed(2000)
  K=length(par)
  Mean=vector()
  RMSE=vector()
  Bias=vector()
  samplesize=as.vector(t(mapply(rep,size,K)))
  for(iter_size in 1:length(size)){
    coef1=matrix(NA,samp,K)
    colnames(coef1)=c('alpha','beta','lambda','b')
    for(nsamp in 1:samp){
      tryCatch(
        {
          x1_BTIIHLTLE=NULL
          q=runif(size[iter_size],0,1)
          x1=sapply(q,BTIIHLTLE_quantile,parameter=par)
          # MLE

```

```

        x1_BTIIHLTLE=mle2(BTIIHLTLE_neglogl,
        start=list(alpha=par[1],beta=par[2],lambda=par[3],b=par[4])
        ,method="L-BFGS-B",data=list(x=x1),lower=c(alpha=0,beta=0,
        lambda=0,b=0),upper=c(alpha=Inf,beta=Inf,lambda=Inf,b=Inf)
        ,use.ginv=TRUE)
        coef1[nsamp,]=coef(x1_BTIIHLTLE)
    },error=function(e){}
)
}
Mean[length(size)*(0:(K-1))+iter_size]=apply
(coef1[,2,mean,na.rm=TRUE)
RMSE[length(size)*(0:(K-1))+iter_size]=
apply((coef1-matrix(rep(par,nsamp),ncol=K,byrow=T))^2,2,
function(x){sqrt(mean(x,na.rm=TRUE))})
}
Bias=as.vector(sapply(1:K,function(x)
{Bias[(length(size)*(x-1)+1):(length(size)*x)]
=Mean[(length(size)*(x-1)+1):
(length(size)*x)]-par[x]}))
return(cbind(samplesize,Mean,RMSE,Bias))
}

##-----

sink("C:/Users/DELL/Desktop/SIMULATIONS3/BTIIHLTLE_simulation_full1.txt",append=F,split=F)

par=c(1.0,1.2,1.5)
par_k=c(3.5)
for(i in 1:length(par)){
  for(j in 1:length(par)){
    for(l in 1:length(par)){
      for(p in 1:length(par_k)){
        print(c(par[i],par[j],par[l],par_k[p]))
        print(BTIIHLTLE_simulation(samp=1000,par=c(par[i],par[j],par[l],par_k[p])))
      }
    }
  }
}
sink()

#####BTIIHLTLE_U

```

```

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)
library(zipfR)

##define pdf BTIIHLTLU

BTIIHLTL_U_pdf<-function(alpha,beta,theta,b,x)
{((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
((1-((1-(1-(1-(x/theta))^(2))^(b)))/(1+(1-(1-(x/theta))^(2))^(b))))^(alpha-1))*
(((1-(1-(1-(x/theta))^(2))^(b)))/(1+(1-(1-(x/theta))^(2))^(b))))^(beta-1))*
((4*b*(1/theta)*(1-(x/theta))*
((1-(1-(x/theta))^(2))^(b-1)))/((1+(1-(1-(x/theta))^(2))^(b))^(2))))}

##define cdf BTIIHLTLU
BTIIHLTL_U_cdf<-function(alpha,beta,theta,b,x)
{y=pbeta((1-((1-(1-(1-(x/theta))^(2))^(b)))/(
1+(1-(1-(x/theta))^(2))^(b)))),alpha,beta)}

BTIIHLTL_LOG_cdf <-function(alpha,beta,b,d,x){Ibeta
(1-((1-(1-(1+x^(c))^(-(2))^(b)))/(1+(1-(1+x^(c))^(-(2))^(b)))),alpha,beta)}

BTIIHLTL_U_neglogl=function(alpha,beta,theta,b,x){
  -sum(log(BTIIHLTL_U_pdf(alpha,beta,theta,b,x)))}

BTIIHLTL_U_hazard=function(alpha,beta,theta,b,x){
  f=BTIIHLTL_U_pdf(alpha,beta,theta,b,x)
  F=BTIIHLTL_U_cdf(alpha,beta,theta,b,x)
  y=f/(1-F)
  return(y)}

##-----
## define BTIIHLTL quantile
BTIIHLTL_U_quantile =function(parameter,u){
  alpha=parameter[1]
  beta=parameter[2]
  theta=parameter[3]
  b=parameter[4]

```

```

f=function(x){
  BTIIHLTL_U_cdf(alpha,beta,theta,b,x)-u
}
x=min(uniroot.all(f,lower=0,upper=10,tol=0.0001))
return(x)
}

BTIIHLTL_U_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=BTIIHLTL_U_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

```

```

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(1.8,1.8,1.7,1.5),
  par2=c(2.0,2.5,0.9,0.5),
  par3=c(0.5,2.2,1.5,1.5),
  par4=c(0.9,1.0,1.8,0.5),
  par5=c(0.8,1.5,1.6,2.4)

))

```

```

BTIIHLTL_U_QuantileTable(parameter_matrix)
print(parameter_matrix)
print(BTIIHLTL_U_QuantileTable(parameter_matrix))

```

```
##End of Quantile
```

```
#-----
```

```
#####
#define pdf BTIIHLTLU
BTIIHLTL_U_pdf=function(alpha,beta,theta,b,x){
  y=((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  ((1-((1-(1-(1-(x/theta))^(2))^(b)))/(1+(1-(1-(x/theta))^(2))^(b))))^(alpha-1))*
  (((1-(1-(1-(x/theta))^(2))^(b)))/(1+(1-(1-(x/theta))^(2))^(b))))^(beta-1))*
  ((4*b*(1/theta)*(1-(x/theta))*((1-(1-(x/theta))^(2))^(b-1))))/
  ((1+(1-(1-(x/theta))^(2))^(b))^(2)))
}

BTIIHLTL_U_moments=function(alpha,beta,theta,b,n){
  f=function(alpha,beta,theta,b,n,x){
    (x^n)*(BTIIHLTL_U_pdf(alpha,beta,theta,b,x))
  }
  y=integrate(f,lower=0,upper=1,subdivisions=10000,
  alpha=alpha,beta=beta,theta=theta,b=b,n=n)
  return(y$value)
}

### table of moments
parameter_matrix_Moments=matrix(c(0.8,1.3,1.9,1.4,
0.9,0.5,1.6,1.5,0.5,2.0,1.5,1.5,0.9,0.9,1.8,0.5,
0.8,1.5,1.6,1.4),ncol=4,byrow=T)
parameter_matrix_Moments
list_Moments=c('EX','EX2','EX3','EX4','EX5',
'EX6','SD','CV','CS','CK')
Table_Moments=matrix(0,nrow=length(list_Moments),
ncol=dim(parameter_matrix_Moments)[1])
row.names(Table_Moments)=list_Moments
Table_Moments

for(iter in 1:dim(parameter_matrix_Moments)[1]){
  parameter=parameter_matrix_Moments[iter,]
  for(i in 1:10){
    Table_Moments[i,iter]=BTIIHLTL_U_moments
    (parameter[1],parameter[2],parameter[3],parameter[4],i)
  }
}

```

```
print(Table_Moments)
###moments end
```

Skewness and Kurtosis Plots

```
#####BTIIHRTL_LLoG
library(plot3Drgl)
F=function(x,par){
  alpha=par[1];beta =par[2];c= par[3];b=par[4]
  C = pbeta(1-(((1-(1-(1+x^(c))^(-(2))))^(b)))/
    (1+(1-(1+x^(c))^(-(2))))^(b))),alpha,beta)
return(C)
}
#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2];c= par[3];b=par[4]
  D=((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
    (((2*(1-(1+x^(c))^(-(2))))^(b))/(1+(1-(1+x^(c))^(-(2))))^(b)))^(beta-1))*
    ((4*b*(c*x^(c-1))*((1+x^(c))^(-(3))))*
    ((1-(1+x^(c))^(-(2))))^(b-1)))/((1+(1-(1+x^(c))^(-(2))))^(b))^(2)))

return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
alpha=seq(1,7,.1)
beta=seq(1,7,.1)
G_kurt<- function(alpha,beta) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(alpha, beta,0.9,1.2))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
  #return(Kurtosis)}

Skewness=skewness(U)
return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('alpha', 'beta'))
#Kurtosis<-outer(alpha,beta,G_kurt)
```

```

Skewness<-outer(alpha,beta,G_kurt)
persp3D(alpha,beta, Skewness,
main=expression(paste(BTIIHLTL_LLoG(alpha, beta,0.9, 1.2)))
,xlab="alpha",ylab="beta",zlab="Skewness" )
#####

####BTIIHLTL_U
library(plot3Drgl)
library(zipfR)
F=function(x,par){
  alpha=par[1];beta =par[2];theta= par[3];b=par[4]
  C = pbeta((1-((1-(1-(1-(x/theta))^(2))^(b)))/
  (1+(1-(1-(x/theta))^(2))^(b))))),alpha,beta)

return(C)
}
#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2];theta= par[3];b=par[4]
  D=((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  ((1-((1-(1-(1-(x/theta))^(2))^(b)))/
  (1+(1-(1-(x/theta))^(2))^(b))))^(alpha-1))*
  (((1-(1-(1-(x/theta))^(2))^(b)))/(1+(1-(1-(x/theta))^(2))^(b))))^(beta-1))*
  ((4*b*(1/theta)*(1-(x/theta))*
  ((1-(1-(x/theta))^(2))^(b-1)))/((1+(1-(1-(x/theta))^(2))^(b))^(2)))

return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
alpha=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(alpha,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(alpha, 2.9,5.2,b))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
#return(Kurtosis)}

Skewness=skewness(U)

```

```

return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('alpha', 'b'))
#Kurtosis<-outer(alpha,b,G_kurt)
Skewness<-outer(alpha,b,G_kurt)
persp3D(alpha,b, Skewness,
main=expression(paste(BTIIHLTL_U(alpha,2.9,5.2, b))),
xlab="alpha",ylab="b",zlab="Skewness" )
#####
####BTIIHLTL_E
BTIIHLTL_E_pdf<-function(alpha,beta,lambda,b){((gamma(alpha+beta))/
(gamma(alpha)*gamma(beta)))*(((2*(1-exp(-(2*lambda*x)))^(b))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
(((1-(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x))))*
((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b))^(2))))}

BTIIHLTL_E_cdf<-function(alpha,beta,lambda,b){
pbeta(((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
}

####BTIIHLTL_E
library(plot3Drgl)
library(zipfR)
F=function(x,par){
  alpha=par[1];beta =par[2];lambda= par[3];b=par[4]
  C = pbeta(((2*(1-exp(-(2*lambda*x)))^(b))/
  (1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)

  return(C)
}

#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2];lambda= par[3];b=par[4]
  D =((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  (((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
  (((1-(1-exp(-(2*lambda*x)))^(b)))/
  (1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*((4*b*lambda*(exp(-(2*lambda*x))))*
  ((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b))^(2))))

```

```

return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
beta=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(beta,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c( 6.9,beta,3.6,b))
Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
return(Kurtosis)}

#Skewness=skewness(U)
#return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('beta', 'b'))
Kurtosis<-outer(beta,b,G_kurt)
#Skewness<-outer(beta,b,G_kurt)
persp3D(beta,b, Kurtosis,main=expression
(paste(BTIIHLTL_E(6.9,beta,3.6, b))),
xlab="beta",ylab="b",zlab="Kurtosis" )

```

Applications and Goodness of Fit

```

#####MONTHLY ACTUAL TAXES REVENUE EGYTP(JAN 2006-NOV 2010) APPLICATION
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(zipfR)

x<-c(5.9, 20.4, 14.9, 16.2, 17.2, 7.8,
  6.1, 9.2, 10.2, 9.6, 13.3, 8.5, 21.6,
18.5,5.1,6.7, 17, 8.6, 9.7, 39.2, 35.7,
15.7, 9.7, 10, 4.1, 36, 8.5, 8, 9.2, 26.2,21.9,16.7,
21.3, 35.4, 14.3, 8.5, 10.6, 19.1, 20.5,
  7.1, 7.7, 18.1, 16.5, 11.9, 7,8.6,12.5, 10.3, 11.2,
6.1, 8.4, 11, 11.6, 11.9, 5.2, 6.8, 8.9, 7.1, 10.8

```

)

```
BTIIHLTLE_pdf<-function(alpha,beta,lambda,b)
{-sum(log(((gamma(alpha+beta))/(gamma(alpha)*gamma(beta))
)*(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
(((1-(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x))))*
((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b))^(2))))}
mle.result<-mle2(BTIIHLTLE_pdf,hessian = NULL,
start=list(alpha=1.000902,beta=1.091700,
lambda=0.009371, b=30.870),optimizer="nlminb",lower=0)
summary(mle.result)
vcov(mle.result)
output1<-vcov(mle.result)
1.96*sqrt(diag(output1))
```


TIIHLW

```
TIIHLW_LL<- function(a,lambda,delta,gamma) {
  -sum(log(2*a*lambda*delta*gamma*(x^(gamma-1))*
    (exp(-delta*x^gamma))*((1-exp(-delta*x^gamma))^
    (lambda-1))*(((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/
    (1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
```

```
))
```

```
}
```

```
mo.result<-mle2(TIIHLW_LL,hessian = NULL,
start=list(a=1.01,lambda= 0.93,delta=
0.000000000009,gamma=0.0006),optimizer="nlminb",lower=0)
summary(mo.result)
```

#####

```
BW_LL<-function(a,b,alpha,beta){-sum(log((beta*
(alpha^beta)/
beta(a,b))*(x^(beta-1))*(exp(-b*(alpha*x)^beta))*
```

```

(1-exp(-(alpha*x)^beta))^(a-1)))}
mle.result<-mle2(BW_LL,hessian = NULL,
start=list(a=9.7,b=1.3,alpha=1.1,beta=0.20),
optimizer="nlminb",lower=0)
summary(mle.result)

#####TTLW
TTLW_LL<-function(a,b,alpha,lambda){-sum(log(2*alpha*b*(a^b)*
(x^(b-1))*exp(-2*(a*x)^b)*((1-exp(-2*(a*x)^b))^(alpha-1))*
(1+lambda-2*lambda*(1-exp(-2*(a*x)^b))^alpha)))}
mle.result<-mle2(TTLW_LL,hessian = NULL,start=list(a=0.2,b=0.6,
alpha=0.891,lambda=0.009),optimizer="nlminb",lower=0)
summary(mle.result)
#####
BOLU_LL<-function(a,b,lambda,theta){-sum(log((gamma(a+b)/
(gamma(a)*gamma(b)))*
((1-((lambda+(1-x/theta))/((1+lambda)*(1-x/theta)))
*exp(-lambda*x/(theta-x)))^(a-1))*
(((lambda+(1-x/theta))/((1+lambda)*
(1-x/theta)))*exp(-lambda*x/(theta-x)))^(b-1))*
(lambda^2/(1+lambda))*
(theta^2/(theta-x)^3)*exp(-lambda*x/(theta-x))))}
mle.result<-mle2(BOLU_LL,hessian = NULL,
start=list(a=0.005,b=2.702175 ,
lambda=1.0035190,theta= 50.178770),optimizer="nlminb",lower=0)
summary(mle.result)

##### DATA 33 YARN AT 2.3 PERCENT APPLICATION

x<-c(86, 146 ,251 ,653 ,98, 249 ,400 ,292, 131, 169, 175, 176 ,76 ,264,
15, 364, 195, 262, 88, 264, 157, 220, 42 ,321, 180, 198 ,38 ,20 ,61,
121, 282 ,224, 149, 180, 325, 250 ,196, 90, 229, 166, 38, 337, 65, 151
,341 ,40 ,40 ,135, 597 ,246, 211 ,180, 93 ,315, 353, 571, 124, 279, 81
,186, 497, 182, 423, 185, 229 ,400 ,338
,290, 398, 71, 246, 185 ,188 ,568, 55 ,55 ,61, 244 ,20 ,284 ,393, 396,
203 ,829, 239, 236 ,286 ,194 ,277, 143 ,198, 264, 105
,203 ,124 ,137, 135, 350, 193 ,188)

BTIIHLE_pdf<-function(alpha,beta,lambda,b)
{-sum(log(((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))
*(((2*(1-exp(-(2*lambda*x)))^(b)))/

```

```

(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
(((1-(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x)))*
((1-exp(-(2*lambda*x)))^(b-1))))/
((1+(1-exp(-(2*lambda*x)))^(b))^(2))))}
mle.result<-mle2(BTIIHLTLE_pdf,hessian = NULL,
start=list(alpha=1.000902,beta=1.091700,
lambda=0.009371, b=30.870),optimizer="nlminb",lower=0)
summary(mle.result)
vcov(mle.result)
output1<-vcov(mle.result)
1.96*sqrt(diag(output1))

```

```

#####
TIIEHLW

```

```

TIIEHLW_LL<- function(a,lambda,delta,gamma) {
  -sum(log(2*a*lambda*delta*gamma*(x^(gamma-1))*
    (exp(-delta*x^gamma))*((1-exp(-delta*x^gamma))^(lambda-1))*
    (((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/
    (1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
  ))
}
mle.result<-mle2(TIIEHLW_LL,hessian = NULL,
start=list(a=1.01,lambda= 0.93,delta= 0.000000000009,
gamma=0.0006),optimizer="nlminb",lower=0)
summary(mle.result)

```

```

#####
BW_LL<-function(a,b,alpha,beta){-sum(log((beta*
  (alpha^beta)/beta(a,b))*(x^(beta-1))*
  (exp(-b*(alpha*x)^beta))*
  (1-exp(-(alpha*x)^beta))^(a-1))))}
mle.result<-mle2(BW_LL,hessian = NULL,start=list
(a=9.7,b=1.3,alpha=1.1,beta=0.20),
optimizer="nlminb",lower=0)

```

```

summary(mle.result)

#####TTLW
TTLW_LL<-function(a,b,alpha,lambda){-sum(log
(2*alpha*b*(a^b)*(x^(b-1))
*exp(-2*(a*x)^b)*((1-exp(-2*(a*x)^b))^(alpha-1))*
(1+lambda-2*lambda*(1-exp(-2*(a*x)^b))^alpha)))}
mle.result<-mle2(TTLW_LL,hessian = NULL,start=list(a=0.2,b=0.6,
alpha=0.891,lambda=0.009),optimizer="nlminb",lower=0)
summary(mle.result)
#####

BGL <- function(alpha,theta,a,b){
  -sum(log(
    (alpha*(theta^2)*(1+x)*(exp(-theta*x))/beta(a,b))*
    ((1-((1+theta+theta*x)/(1+theta))*
    exp(-theta*x))^(a*alpha-1))*
    (1-(1-((1+theta+theta*x)/(1+theta))*
    exp(-theta*x))^alpha)^(b-1)))
  }
BGL.result<-mle2(BGL,hessian = NULL,
start=list(alpha=0.099,theta=0.03
,a=0.3,b=10.0999),optimizer="nlminb",lower=0)
summary(BGL.result)

##### YARN AT 2.3 PERCENT DATASET GOODNESS OF FIT
library(AdequacyModel)
library(zipfR)
data<-c(86, 146 ,251 ,653 ,98, 249 ,400 ,292, 131, 169, 175, 176 ,76 ,264,
15, 364, 195, 262, 88, 264, 157, 220, 42 ,321, 180, 198 ,38 ,20 ,61,
121, 282 ,224, 149, 180, 325, 250 ,196, 90, 229, 166, 38, 337, 65, 151
,341 ,40 ,40 ,135, 597 ,246, 211 ,180, 93 ,315, 353, 571, 124, 279, 81
,186, 497, 182, 423, 185, 229 ,400 ,338
,290, 398, 71, 246, 185 ,188 ,568, 55 ,55 ,61, 244 ,20 ,284 ,393, 396,
203 ,829, 239, 236 ,286 ,194 ,277, 143 ,198, 264, 105
,203 ,124 ,137, 135, 350, 193 ,188)

BTIIHLTLE_pdf<-function(par,x){

  alpha=par[1]
  beta=par[2]

```

```

lambda=par[3]
b=par[4]
((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
(((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
(((1-(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x))))*
((1-exp(-(2*lambda*x)))^(b-1)))/((1+(1-exp(-(2*lambda*x)))^(b))^(2)))
}

BTIIHLE_cdf<-function(par,x){

alpha=par[1]
beta=par[2]
lambda=par[3]
b=par[4]
###pbeta(((1-((1-(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b))))),alpha,beta)
pbeta(((2*(1-exp(-(2*lambda*x)))^(b))/(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
}

goodness.fit(pdf=BTIIHLE_pdf, cdf=BTIIHLE_cdf,
             mle=c(0.197149
,4.684423,0.304916 ,108.128432),data =
             data,method = "BFGS",
             domain = c(0,1), lim_inf = c(0,0,0,0),
             lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

#####
BTIIHLE_LOG_pdf<-function(par,x){

alpha=par[1]
beta=par[2]
c=par[3]
b=par[4]
((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
(((2*(1-(1+x^(c))^(b-2))^(b))/(1+(1-(1+x^(c))^(b-2))^(b)))^(beta-1))
*((4*b*(c*x^(c-1))*((1+x^(c))^(b-3)))*
((1-(1+x^(c))^(b-2))^(b-1)))/((1+(1-(1+x^(c))^(b-2))^(b))^(2)))
}

```

```

BTIIHLTL_LOG_cdf<-function(par,x){

  alpha=par[1]
  beta=par[2]
  c=par[3]
  b=par[4]
  Ibeta(1-((1-(1-(1+x^(c))^(-(2)))^(b))/(1+(1-(1+x^(c))^(-(2)))^(b))),alpha,beta)}

goodness.fit(pdf=BTIIHLTL_LOG_pdf, cdf=BTIIHLTL_LOG_cdf,
mle=c( 3.2265e-01 ,3.8944e-03 , 1.0911e+00,5.0565e+02 ),
data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

#####
#TIIHLW
TIIHLW_pdf<-function(parameter,x){
  a=parameter[1]
  lambda=parameter[2]
  delta=parameter[3]
  gamma=parameter[4]
  2*a*lambda*delta*gamma*(x^(gamma-1))*(exp(-delta*x^gamma))*
  ((1-exp(-delta*x^gamma))^(lambda-1))*
  (((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/(1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
}

TIIHLW_cdf<-function(parameter,x){
  a=parameter[1]
  lambda=parameter[2]
  delta=parameter[3]
  gamma=parameter[4]
  1-((1-((1-exp(-delta*x^gamma))^(lambda)))/(1+(1-exp(-delta*x^gamma))^(lambda))))^a
}

goodness.fit(pdf=TIIHLW_pdf,cdf= TIIHLW_cdf,mle=
c(1.8361e+03 ,2.0530e+01 ,5.0765e-01 ,1.4238e-01),
data=data,method = "BFGS",
      domain = c(0,100),lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

```

```
#####
BW_pdf<-function(par,x){

  a=par[1]
  b=par[2]
  alpha=par[3]
  beta=par[4]
  (beta*(alpha^beta)/beta(a,b))*(x^(beta-1))*
  (exp(-b*(alpha*x)^beta))
  *(1-exp(-(alpha*x)^beta))^(a-1)
}

BW_cdf<-function(par,x){

  a=par[1]
  b=par[2]
  alpha=par[3]
  beta=par[4]

  pbeta(1-(exp(-(alpha*x)^beta)),a,b)
}

goodness.fit(pdf=BW_pdf, cdf=BW_cdf,mle=c(1.3623214,0.9761200 ,
0.0049185,1.3643917),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

goodness.fit(pdf=BW_pdf, cdf=BW_cdf,mle=c(12.55933 ,
0.52700,0.11082,0.63868),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
#####
####BOLU
BOLU_pdf<-function(par,x){
```

```

a=par[1]
b=par[2]
lambda=par[3]
theta=par[4]
(gamma(a+b)/(gamma(a)*gamma(b)))*((1-((lambda+(1-x/theta))/
((1+lambda)*(1-x/theta))))*exp(-lambda*x/(theta-x)))^(a-1))*
(((lambda+(1-x/theta))/((1+lambda)*(1-x/theta))))*
exp(-lambda*x/(theta-x)))^(b-1))*(lambda^2/(1+lambda))*
(theta^2/(theta-x)^3)*exp(-lambda*x/(theta-x))
}

```

```

BOLU_cdf<-function(par,x){

a=par[1]
b=par[2]
lambda=par[3]
theta=par[4]

pbeta(1-((lambda+(1-x/theta))/((1+lambda)*(1-x/theta))))
*exp(-lambda*x/(theta-x)),a,b,log=FALSE)
}

```

```

goodness.fit(pdf=BOLU_pdf, cdf=BOLU_cdf,mle=c( 1.1778e+00
,1.5097e+00 , 3.1835e+05,3.9174e+06),data = data,method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup =c(100000000000000000,
100000000000000000,100000000000000000) )

```

#####

```

TTLW_pdf<-function(par,x){

```

```

a=par[1]
b=par[2]
alpha=par[3]

```

```

lambda=par[4]

2*alpha*b*(a^b)*(x^(b-1))*exp(-2*(a*x)^b)*
((1-exp(-2*(a*x)^b))^(alpha-1))*
(1+lambda-2*lambda*(1-exp(-2*(a*x)^b))^alpha)
}

TTLW_cdf<-function(par,x){

a=par[1]
b=par[2]
alpha=par[3]
lambda=par[4]

((1+lambda)*((1-exp(-2*(a*x)^b))))-lambda*(1-exp(-2*(a*x)^b))^(2*alpha)
}

goodness.fit(pdf=TTLW_pdf, cdf=TTLW_cdf,mle=c(0.184759,
3.416975,2.177136,0.0000000 ),data = data,
method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup = c(1000000000000,
1000000000000,1000000000000,1000000000000))
#####
####BGL
BGL_pdf<-function(parameter,x){
alpha=parameter[1]
theta=parameter[2]
a=parameter[3]
b=parameter[4]
(alpha*(theta^2)*(1+x)*(exp(-theta*x))/beta(a,b))*
((1-((1+theta+theta*x)/(1+theta))*exp(-theta*x))^(a*alpha-1))
*(1-(1-((1+theta+theta*x)/(1+theta))*exp(-theta*x))^alpha)^(b-1)
}
# define BGL cdf
BGL_cdf<-function(parameter,x){

```

```

alpha=parameter[1]
theta=parameter[2]
a=parameter[3]
b=parameter[4]
(1/(beta(a,b)))*pbeta(((1-((1+theta+theta*x)/(1+theta))
*exp(-theta*x))^(alpha)),a,b,log=FALSE)
}

```

```

goodness.fit(pdf= BGL_pdf, cdf= BGL_cdf,mle=c(
      1.0049e-01 ,5.3027e-09,3.0009e-01 ,1.0100e+01 ),
data = data,method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup = c(1000000000000,
1000000000000,1000000000000))
#####

```

#####MONTHLY ACTUAL TAXES REVENUE EGYTP(JAN 2006-NOV GOODNESS OF FIT 2010

```

library(AdequacyModel)
library(zipfR)

```

```

data=c(
5.9, 20.4, 14.9, 16.2, 17.2, 7.8, 6.1, 9.2,
10.2, 9.6, 13.3, 8.5, 21.6,
18.5,5.1,6.7, 17, 8.6, 9.7, 39.2, 35.7, 15.7,
9.7, 10, 4.1, 36, 8.5, 8, 9.2, 26.2,21.9,16.7,
21.3, 35.4, 14.3, 8.5, 10.6, 19.1, 20.5, 7.1,
7.7, 18.1, 16.5, 11.9, 7,8.6,12.5, 10.3, 11.2,
6.1, 8.4, 11, 11.6, 11.9, 5.2, 6.8, 8.9, 7.1, 10.8
)

```

```

BTIIHLE_pdf<-function(par,x){

```

```

alpha=par[1]
beta=par[2]
lambda=par[3]
b=par[4]
((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*

```

```

(((1-(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
((4*b*lambda*(exp(-(2*lambda*x)))*
((1-exp(-(2*lambda*x)))^(b-1))))/
((1+(1-exp(-(2*lambda*x)))^(b))^(2)))
}

```

```
BTIIHLTLE_cdf<-function(par,x){
```

```

  alpha=par[1]
  beta=par[2]
  lambda=par[3]
  b=par[4]

```

```

pbeta(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
}

```

```

goodness.fit(pdf=BTIIHLTLE_pdf, cdf=BTIIHLTLE_cdf,mle=c(2.8040e-01
,2.9689e-01, 1.9223e-01 ,7.6389e+01),data = data,method = "BFGS",
  domain = c(0,1), lim_inf = c(0,0,0,0),
  lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

```

```
#####
```

```
####BTIIHRTL_LLoG
```

```
BTIIHRTL_LOG_pdf<-function(par,x){
```

```

  alpha=par[1]
  beta=par[2]
  c=par[3]
  b=par[4]

```

```

((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
(((2*(1-(1+x^(c))^(-(2))))^(b))/
(1+(1-(1+x^(c))^(-(2))))^(b)))^(beta-1))*
((4*b*(c*x^(c-1))*((1+x^(c))^(-(3))))*
((1-(1+x^(c))^(-(2))))^(b-1)))/
((1+(1-(1+x^(c))^(-(2))))^(b))^(2)))
}

```

```
BTIIHLTL_LOG_cdf<-function(par,x){
```

```
  alpha=par[1]
```

```
  beta=par[2]
```

```
  c=par[3]
```

```
  b=par[4]
```

```
  Ibeta(1-((1-(1-(1+x^(c))^(-(2))))^(b))/  
(1+(1-(1+x^(c))^(-(2))))^(b))),alpha,beta)}
```

```
goodness.fit(pdf=BTIIHLTL_LOG_pdf, cdf=BTIIHLTL_LOG_cdf,mle=c(  
3.2265e-01 ,3.8944e-03 , 1.0911e+00,5.0565e+02 )  
,data = data,method = "BFGS",  
      domain = c(0,1), lim_inf = c(0,0,0,0),  
      lim_sup = c(10000000000000000,  
10000000000000000,10000000000000000))
```

```
#####
```

```
#TIIHLW
```

```
TIIHLW_pdf<-function(parameter,x){
```

```
  a=parameter[1]
```

```
  lambda=parameter[2]
```

```
  delta=parameter[3]
```

```
  gamma=parameter[4]
```

```
2*a*lambda*delta*gamma*(x^(gamma-1))*(exp(-delta*x^gamma))*  
((1-exp(-delta*x^gamma))^(lambda-1))*  
(((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/  
(1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
```

```
}
```

```
TIIHLW_cdf<-function(parameter,x){
```

```
  a=parameter[1]
```

```
  lambda=parameter[2]
```

```
  delta=parameter[3]
```

```
  gamma=parameter[4]
```

```
1-((1-((1-exp(-delta*x^gamma))^(lambda)))/  
(1+(1-exp(-delta*x^gamma))^(lambda)))^a
```

```
}
```

```
goodness.fit(pdf=TIIEHLW_pdf,cdf= TIIEHLW_cdf,mle=c(0.36261
,27.45624 ,0.49560 ,0.91597),data=data,method = "BFGS",
      domain = c(0,100),lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
```

```
#####
```

```
#####BW
```

```
BW_pdf<-function(par,x){
```

```
  a=par[1]
  b=par[2]
  alpha=par[3]
  beta=par[4]
  (beta*(alpha^beta)/beta(a,b))*(x^(beta-1))*
  (exp(-b*(alpha*x)^beta))
  *(1-exp(-(alpha*x)^beta))^(a-1)
```

```
}
```

```
BW_cdf<-function(par,x){
```

```
  a=par[1]
  b=par[2]
  alpha=par[3]
  beta=par[4]

  pbeta(1-(exp(-(alpha*x)^beta)),a,b)
```

```
}
```

```
goodness.fit(pdf=BW_pdf, cdf=BW_cdf,mle=c(90.7890747,
0.2360539 ,1.0548977, 0.8346324 ),
data = data,method = "BFGS",domain = c(0,1),
lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))
```

#####

#####BOLU

```
BOLU_pdf<-function(par,x){
```

```
  a=par[1]
```

```
  b=par[2]
```

```
  lambda=par[3]
```

```
  theta=par[4]
```

```
  (gamma(a+b)/(gamma(a)*gamma(b)))*((1-((lambda+(1-x/theta))/
```

```
  ((1+lambda)*(1-x/theta)))*exp(-lambda*x/(theta-x)))^(a-1))*
```

```
  (((lambda+(1-x/theta))/((1+lambda)*(1-x/theta)))*
```

```
  exp(-lambda*x/(theta-x)))^(b-1))*(lambda^2/(1+lambda))*
```

```
  (theta^2/(theta-x)^3)*exp(-lambda*x/(theta-x))
```

```
}
```

```
BOLU_cdf<-function(par,x){
```

```
  a=par[1]
```

```
  b=par[2]
```

```
  lambda=par[3]
```

```
  theta=par[4]
```

```
  pbeta(1-((lambda+(1-x/theta))/
```

```
  ((1+lambda)*(1-x/theta)))*
```

```
  exp(-lambda*x/(theta-x)),a,b,log=FALSE)
```

```
}
```

```
goodness.fit(pdf=BOLU_pdf, cdf=BOLU_cdf,mle=c( 3.8800e+00
```

```
,3.4012e+00 , 3.2368e+05, 5.1664e+06),
```

```
data = data,method = "BFGS",
```

```
domain = c(0,100), lim_inf = c(0,0,0,0),
```

```
lim_sup =c(10000000000000000,
```

```
10000000000000000,10000000000000000) )
```

#####

#####TTLW

```
TTLW_pdf<-function(par,x){
```

```
  a=par[1]
```

```

b=par[2]
alpha=par[3]
lambda=par[4]

2*alpha*b*(a^b)*(x^(b-1))*exp(-2*(a*x)^b)*
((1-exp(-2*(a*x)^b))^(alpha-1))*
(1+lambda-2*lambda*(1-exp(-2*(a*x)^b))^alpha)
}

```

```

TTLW_cdf<-function(par,x){

```

```

a=par[1]
b=par[2]
alpha=par[3]
lambda=par[4]

((1+lambda)*((1-exp(-2*(a*x)^b))))-lambda*
(1-exp(-2*(a*x)^b))^(2*alpha)
}

```

```

goodness.fit(pdf=TTLW_pdf, cdf=TTLW_cdf,mle=c(8.7261e+00,
2.9503e-01,1.6253e+03, 7.7947e-02 ),data = data,method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000,
10000000000000,10000000000000,10000000000000))

```

```

#####

```

```

####BGL

```

```

BGL_pdf<-function(parameter,x){
alpha=parameter[1]
theta=parameter[2]
a=parameter[3]
b=parameter[4]
(alpha*(theta^2)*(1+x)*(exp(-theta*x))/beta(a,b))*
((1-((1+theta+theta*x)/
(1+theta))*exp(-theta*x))^(a*alpha-1))*
(1-(1-((1+theta+theta*x)/

```

```

      (1+theta))*exp(-theta*x))^alpha)^(b-1)
}
# define BGL    cdf
BGL_cdf<-function(parameter,x){
  alpha=parameter[1]
  theta=parameter[2]
  a=parameter[3]
  b=parameter[4]
  (1/(beta(a,b)))*pbeta(((1-((1+theta+theta*x)/
  (1+theta))*
  exp(-theta*x))^(alpha))
  ,a,b,log=FALSE)
}

goodness.fit(pdf= BGL_pdf, cdf= BGL_cdf,mle=c(1.359631
,0.813021, 17.725004 , 0.177460 ), data = data,method = "BFGS",
      domain = c(0,100), lim_inf = c(0,0,0,0),
      lim_sup = c(1000000000000,
      1000000000000,1000000000000,1000000000000))
#####

```

Fitted Densities and Probability Plots

#####YARN DATASET AT 2.3% FITTED DENSITIES

```

par(mfrow=c(0,1))
x<-c(86, 146 ,251 ,653 ,98, 249 ,400 ,292, 131, 169, 175, 176 ,76 ,264,
  15, 364, 195, 262, 88, 264, 157, 220, 42 ,321, 180, 198 ,38 ,20 ,61,
  121, 282 ,224, 149, 180, 325, 250 ,196, 90, 229, 166, 38, 337, 65, 151
  ,341 ,40 ,40 ,135, 597 ,246, 211 ,180, 93 ,315, 353, 571, 124, 279, 81
  ,186, 497, 182, 423, 185, 229 ,400 ,338
  ,290, 398, 71, 246, 185 ,188 ,568, 55 ,55 ,61, 244 ,20 ,284 ,393, 396,
  203 ,829, 239, 236 ,286 ,194 ,277, 143 ,198, 264, 105
  ,203 ,124 ,137, 135, 350, 193 ,188
)

```

#####BTIIHLTLE

```

BTIIHLTLE_pdf<-function(alpha,beta,lambda,b,x){
  ((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  (((2*(1-exp(-(2*lambda*x)))^(b)))/
  (1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
  (((1-(1-exp(-(2*lambda*x)))^(b)))/

```

```

      ((1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
      ((4*b*lambda*(exp(-(2*lambda*x)))*)
      ((1-exp(-(2*lambda*x)))^(b-1)))/
      ((1+(1-exp(-(2*lambda*x)))^(b))^(2)))
    }

#####TIIHLW
TIIHLW_pdf<-function(a,lambda,delta,gamma,x){
2*a*lambda*delta*gamma*(x^(gamma-1))*(exp(-delta*x^gamma))*
((1-exp(-delta*x^gamma))^(lambda-1))*
(((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/
(1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
}

#####BW
BW_pdf<-function(a,b,alpha,beta,x){
(beta*(alpha^beta)/beta(a,b))*(x^(beta-1))*(exp(-b*(alpha*x)^beta))*
(1-exp(-(alpha*x)^beta))^(a-1)
}

#####TLW
TLW_pdf<-function(a,b,alpha,lambda,x){
2*alpha*b*(a^b)*(x^(b-1))*exp(-2*(a*x)^b)*((1-exp(-2*(a*x)^b))^(alpha-1))
*(1+lambda-2*lambda*(1-exp(-2*(a*x)^b))^alpha)
}

#####BGL
BGL_pdf<-function(alpha,theta,a,b,x){
(alpha*(theta^2)*(1+x)*(exp(-theta*x))/beta(a,b))*
((1-((1+theta+theta*x)/(1+theta))*exp(-theta*x))^(a*alpha-1))*
(1-(1-((1+theta+theta*x)/(1+theta))*exp(-theta*x))^alpha)^(b-1)
}

hist(x, prob=TRUE, br=9, main='Fitted Densities'
,xlab="Life time",ylab="Density",ylim=c(0,0.004))

curve(BTIIHLTLW_pdf(x, alpha= 0.30168084,
beta= 1.90987965,lambda=0.00205500,b=
6.28726198),lty=6, col=2, add=TRUE, lwd=3)
#par(new=TRUE)
curve(TIIHLW_pdf(x, a=1.8361e+03 ,

```

```

lambda=2.0530e+01,delta= 5.0765e-01,
  gamma= 1.4238e-01), lty=2, col=7, add=TRUE, lwd=3)
# par(new=TRUE)
curve(BW_pdf(x, a= 1.3623214 ,b=0.9761200
, alpha=0.0049185, beta= 1.3643917 ), lty=2,
  col=4, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TTLW_pdf(x, a=0.0029129, b=1.3625192,
alpha=1.3645608, lambda=0.000000 ),lty=2,
  col=3, add=TRUE,lwd=2)
#par(new=TRUE)
curve(BGL_pdf (x,alpha= 1.0049e-01 ,
theta=5.3027e-09,a=3.0009e-01, b=1.0100e+01)
,lty=2, col=6, add=TRUE,lwd=3)
#par(new=TRUE)

```

```

#par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,c
("BTIIHLTLE","TIIHLW","BW","TTLW","BGL"),
col=c(2,7,4,3,6),lwd=c(2,2,2),lty=c(6,2,2,2))

```

MONTHLY ACTUAL TAXES REVENUE FITTED DENSITIES

```

par(mfrow=c(0,1))
x<-c(5.9, 20.4, 14.9, 16.2, 17.2,
7.8, 6.1, 9.2, 10.2, 9.6, 13.3, 8.5, 21.6,
18.5,5.1,6.7, 17, 8.6, 9.7, 39.2,
35.7, 15.7, 9.7, 10, 4.1, 36, 8.5,
8, 9.2, 26.2,21.9,16.7,
21.3, 35.4, 14.3, 8.5, 10.6, 19.1,
20.5, 7.1, 7.7, 18.1, 16.5, 11.9, 7,
8.6,12.5, 10.3, 11.2,
6.1, 8.4, 11, 11.6, 11.9, 5.2, 6.8, 8.9, 7.1, 10.8)

```

#####BTIIHLTLE

```

BTIIHLTLE_pdf<-function(alpha,beta,lambda,b,x){
  ((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*
  (((2*(1-exp(-(2*lambda*x)))^(b)))/
  (1+(1-exp(-(2*lambda*x)))^(b)))^(alpha -1))*
  (((1-(1-exp(-(2*lambda*x)))^(b)))/
  (1+(1-exp(-(2*lambda*x)))^(b)))^(beta -1))*
  ((4*b*lambda*(exp(-(2*lambda*x)))*
  ((1-exp(-(2*lambda*x)))^(b-1)))/(
  (1+(1-exp(-(2*lambda*x)))^(b))^2)))
}

#####TIIHLW_LL
TIIHLW_pdf<-function(a,lambda,delta,gamma,x){
  2*a*lambda*delta*gamma*(x^(gamma-1))*(exp(-delta*x^gamma))*
  ((1-exp(-delta*x^gamma))^(lambda-1))*
  (((1-(1-exp(-delta*x^gamma))^(lambda))^(a-1))/
  (1+(1-exp(-delta*x^gamma))^(lambda))^(a+1))
}

#####BW_LL
BW_pdf<-function(a,b,alpha,beta,x){
  (beta*(alpha^beta)/beta(a,b))*(x^(beta-1))*
  (exp(-b*(alpha*x)^beta))*(1-exp(-(alpha*x)^beta))^(a-1)
}

#####TTLW_LL
TTLW_pdf<-function(a,b,alpha,lambda,x){
  2*alpha*b*(a^b)*(x^(b-1))*exp(-2*(a*x)^b)*
  ((1-exp(-2*(a*x)^b))^(alpha-1))*(1+lambda-2*lambda
  *(1-exp(-2*(a*x)^b))^alpha)
}

#####BGL
BGL_pdf<-function(alpha,theta,a,b,x){
  (alpha*(theta^2)*(1+x)*(exp(-theta*x))/beta(a,b))*
  ((1-((1+theta+theta*x)/(1+theta))*
  exp(-theta*x))^(a*alpha-1))*(1-(1-((1+theta+theta*x)/
  (1+theta))*exp(-theta*x))^alpha)^(b-1)
}

####BOLU

```

```

BOLU_pdf<-function(a,b,lambda,theta,x){(gamma(a+b)/
(gamma(a)*gamma(b)))*((1-((lambda+(1-x/theta))/
((1+lambda)*(1-x/theta))))*exp(-lambda*x/(theta-x)))^(a-1))
*(((lambda+(1-x/theta))/((1+lambda)*(1-x/theta))))*
exp(-lambda*x/(theta-x)))^(b-1))*(lambda^2/(1+lambda))*
(theta^2/(theta-x)^3)*exp(-lambda*x/(theta-x))}

```

```

hist(x, prob=TRUE, br=9, main='Fitted Densities'
,xlab="Life time",ylab="Density",ylim=c(0,0.12))

```

```

curve(BTIIHLTLTLE_pdf(x, alpha= 2.8040e-01 ,
beta= 2.9689e-01,lambda= 1.9223e-01
,b=7.6388e+01),lty=6, col=2, add=TRUE, lwd=2)
#par(new=TRUE)
curve(TIIEHLW_pdf(x, a=0.36261 ,
lambda=27.45624,delta= 0.49560,
gamma=0.91597), lty=2, col=7, add=TRUE, lwd=3)
# par(new=TRUE)
curve(BW_pdf(x, a= 90.7890747 ,
b= 0.2360539 , alpha=1.0548977,
beta= 0.8346324), lty=2, col=4, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TTLW_pdf(x, a=8.7261e+00,
b=2.9503e-01, alpha=1.6253e+03,
lambda=7.7947e-02 ),lty=2, col=3, add=TRUE,lwd=3)
#par(new=TRUE)
curve(BOLU_pdf (x, a= 3.8800e+00
,b=3.4012e+00 ,lambda=3.2368e+05,
theta=5.1664e+06),lty=2, col=6, add=TRUE,lwd=3)
#par(new=TRUE)

```

```

#par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,c("BTIIHLTLTLE",
"TIIEHLW", "BW", "TTLW", "BOLU"),col=c(2,7,4,3,6)

```

```
\,lwd=c(2,2,2),lty=c(6,2,2,2))
```

```
#####
```

```
##### YARN DATASEAT 2.3 % PROBABILITY PLOTS
```

```
data=c(
```

```
86, 146 ,251 ,653 ,98, 249 ,400 ,292, 131, 169, 175, 176 ,76 ,264,  
15, 364, 195, 262, 88, 264, 157, 220, 42 ,321, 180, 198 ,38 ,20 ,61,  
121, 282 ,224, 149, 180, 325, 250 ,196, 90, 229, 166, 38, 337, 65, 151  
,341 ,40 ,40 ,135, 597 ,246, 211 ,180, 93 ,315, 353, 571, 124, 279, 81  
,186, 497, 182, 423, 185, 229 ,400 ,338  
,290, 398, 71, 246, 185 ,188 ,568, 55 ,55 ,61, 244 ,20 ,284 ,393, 396,  
203 ,829, 239, 236 ,286 ,194 ,277, 143 ,198, 264,  
105 ,203 ,124 ,137, 135, 350, 193 ,188)
```

```
data<-sort(data)
```

```
n=length(data)
```

```
#####
```

```
#####BTIIHLTLE
```

```
alpha1=0.30168084
```

```
beta1= 1.90987965
```

```
lambda1=0.00205500
```

```
b1= 6.28726198
```

```
#####TIIHLW_LL
```

```
a2=1.8361e+03
```

```
lambda2= 2.0530e+01
```

```
delta2=5.0765e-01
```

```
gamma2=1.4238e-01
```

```
#####BW_LL
```

```
a3= 1.3623214
```

```
b3= 0.9761200
```

```
alpha3= 0.0049185
```

```
beta3= 1.3643917
```

```
#####TTLW_LL
```

```
a4= 0.0029129
```

```
b4=1.3625192
```

```
alpha4=1.3645608
```

```
lambda4=0.000000
```

```
#####BGL
```

```
alpha5=1.0049e-01
```

```
theta5=5.3027e-09
```

```

a5=3.0009e-01
b5=1.0100e+01

F_observed<-c(rep(0,n))

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

require(gsl)
require(stats)
data=c(
86, 146 ,251 ,653 ,98, 249 ,400 ,292, 131, 169, 175, 176 ,76 ,264,
15, 364, 195, 262, 88, 264, 157, 220, 42 ,321, 180, 198 ,38 ,20 ,61,
121, 282 ,224, 149, 180, 325, 250 ,196, 90, 229, 166, 38, 337, 65, 151
,341 ,40 ,40 ,135, 597 ,246, 211 ,180, 93 ,315, 353, 571, 124, 279, 81
,186, 497, 182, 423, 185, 229 ,400 ,338
,290, 398, 71, 246, 185 ,188 ,568, 55 ,55 ,61, 244 ,20 ,284 ,393, 396,
203 ,829, 239, 236 ,286 ,194 ,277, 143 ,198, 264,
105 ,203 ,124 ,137, 135, 350, 193 ,188)

data<-sort(data)
n=length(data)

```

```
BTIIHLTLE_cdf<-function(data,alpha,beta,lambda,b,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-###pbeta((1-((1-(1-exp(-(2*lambda*x)))^(b))
      /(1+(1-exp(-(2*lambda*x)))^(b))))),alpha,beta)
pbeta(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
  }

  return(F)
}
```

```
TIIEHLW_cdf<-function(data,a,lambda,delta,gamma,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1-(((1-((1-exp(-delta*x^gamma))^(lambda))))/
      (1+(1-exp(-delta*x^gamma))^(lambda))))^a
  }

  return(F)
}
```

```
BW_cdf<-function(data,a,b,alpha,beta,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-pbeta(1-(exp(-(alpha*x)^beta)),a,b)
  }

  return(F)
}
```

```
TTLW_cdf<-function(data,a,b,alpha,lambda,n){
  F<-c(rep(0,n))
```

```

    for (i in 1:n)
    {
        x<-data[i]
        F[i]<- ((1+lambda)*((1-exp(-2*(a*x)^b))))-lambda*
            (1-exp(-2*(a*x)^b))^(2*alpha)
    }

    return(F)
}

BGL_cdf<-function(data,alpha,theta,a,b,n){
    F<-c(rep(0,n))
    for (i in 1:n)
    {
        x<-data[i]
        F[i]<- (1/(beta(a,b)))*pbeta(((1-((1+theta+theta*x)/
            (1+theta))*exp(-theta*x))^(alpha)),a,b,log=FALSE)
    }
    return(F)
}

F1<-BTIIHLE_cdf(data,alpha1,beta1,lambda1,b1,n)
F1
F2<-TIIHLW_cdf(data,a2,lambda2,delta2,gamma2,n)
F2
F3<-BW_cdf(data,a3,b3,alpha3,beta3,n)
F3
F4<-TTLW_cdf(data,a4,b4,alpha4,lambda4,n)
F4
F5<-BGL_cdf(data,alpha5,theta5,a5,b5,n)
F5

SS<-function(F,F_line,n){
    SS<-0
    for (i in 1:n)
    {
        SS<-SS+(F[i]-F_line[i])^2
    }
    return(SS)
}

```

```

SS1<-SS(F1,F_observed,n)
SS1
SS2<-SS(F2,F_observed,n)
SS2
SS3<-SS(F3,F_observed,n)
SS3
SS4<-SS(F4,F_observed,n)
SS4
SS5<-SS(F5,F_observed,n)
SS5

```

```

op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l',
main="Probability Plots"
,xlab="Observed probability",ylab="Expected probability")
lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
legend("bottomright",cex=0.7,c("BTIIHLTLE(SS=0.0755)",
"TIIEHLW(SS=0.0934)","BW(SS=0.0906)",
"TTLW(SS=0.9386)","BGL(SS=7.7091)"),
,lty=c(1,1,1,1,1,1),
col=c("red","blue","purple","orange",
"green"),lwd=c(1,1,1,1,1,1))

```

MONTHLY REVENUE DATA

```

data=c(
5.9, 20.4, 14.9, 16.2, 17.2, 7.8,
6.1, 9.2, 10.2, 9.6, 13.3, 8.5, 21.6,
18.5,5.1,6.7, 17, 8.6, 9.7, 39.2,
35.7, 15.7, 9.7, 10, 4.1, 36, 8.5, 8, 9.2, 26.2,21.9,16.7,
21.3, 35.4, 14.3, 8.5, 10.6, 19.1,
20.5, 7.1, 7.7, 18.1, 16.5, 11.9,
7,8.6,12.5, 10.3, 11.2,
6.1, 8.4, 11, 11.6, 11.9, 5.2, 6.8, 8.9, 7.1, 10.8
)

```

```

data<-sort(data)
n=length(data)

#####
#####BTIIHLE
alpha1=2.8040e-01
beta1= 2.9689e-01
lambda1=1.9223e-01
b1= 7.6388e+01
#####TIEHLW_LL
a2=0.36261
lambda2= 27.45624
delta2=0.49560
gamma2=0.91597
#####BW_LL
a3= 90.7890747
b3= 0.2360539
alpha3= 1.0548977
beta3= 0.8346324
#####TTLW_LL
a4= 8.7261e+00
b4= 2.9503e-01
alpha4=1.6253e+03
lambda4=7.7947e-02
#####BOLU
a5=3.8800e+00
b5=3.4012e+00
lambda5=3.2368e+05
theta5=5.1664e+06

F_observed<-c(rep(0,n))

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

```

```

require(gsl)
require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

require(gsl)
require(stats)
data=c(
5.9, 20.4, 14.9, 16.2, 17.2, 7.8, 6.1, 9.2,
10.2, 9.6, 13.3, 8.5, 21.6,
18.5,5.1,6.7, 17, 8.6, 9.7, 39.2, 35.7, 15.7,
9.7, 10, 4.1, 36, 8.5, 8, 9.2, 26.2,21.9,16.7,
21.3, 35.4, 14.3, 8.5, 10.6, 19.1, 20.5, 7.1,
7.7, 18.1, 16.5, 11.9, 7,8.6,12.5, 10.3, 11.2,
6.1, 8.4, 11, 11.6, 11.9, 5.2, 6.8, 8.9, 7.1, 10.8
)

data<-sort(data)
n=length(data)

BTIIHLTLE_cdf<-function(data,alpha,beta,lambda,b,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-###pbeta((1-((1-(1-exp(-(2*lambda*x)))^(b)))/
      (1+(1-exp(-(2*lambda*x)))^(b))))),alpha,beta)
pbeta(((2*(1-exp(-(2*lambda*x)))^(b)))/
(1+(1-exp(-(2*lambda*x)))^(b))),alpha,beta)
  }
}

```

```

    return(F)
}

TIIHLW_cdf<-function(data,a,lambda,delta,gamma,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-1-(((1-((1-exp(-delta*x^gamma))^(lambda))))/
      (1+(1-exp(-delta*x^gamma))^(lambda))))^a
  }

  return(F)
}

BW_cdf<-function(data,a,b,alpha,beta,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-pbeta(1-(exp(-(alpha*x)^beta)),a,b)
  }

  return(F)
}

TTLW_cdf<-function(data,a,b,alpha,lambda,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<- ((1+lambda)*((1-exp(-2*(a*x)^b))))-lambda*
      (1-exp(-2*(a*x)^b))^(2*alpha)
  }

  return(F)
}

BOLU_cdf<-function(data,a,b,lambda,theta,n){
  F<-c(rep(0,n))
  for (i in 1:n)

```

```

{
  x<-data[i]
  F[i]<- pbeta(1-((lambda+(1-x/theta))/((1+lambda)*
    (1-x/theta))))*exp(-lambda*x/(theta-x)),a,b,log=FALSE)
}
  return(F)
}

```

```
F1<-BTIIHLE_cdf(data,alpha1,beta1,lambda1,b1,n)
```

```
F1
```

```
F2<-TIIHLW_cdf(data,a2,lambda2,delta2,gamma2,n)
```

```
F2
```

```
F3<-BW_cdf(data,a3,b3,alpha3,beta3,n)
```

```
F3
```

```
F4<-TTLW_cdf(data,a4,b4,alpha4,lambda4,n)
```

```
F4
```

```
F5<-BOLU_cdf(data,a5,b5,lambda5,theta5,n)
```

```
F5
```

```

SS<-function(F,F_line,n){
  SS<-0
  for (i in 1:n)
  {
    SS<-SS+(F[i]-F_line[i])^2
  }
  return(SS)
}

```

```
SS1<-SS(F1,F_observed,n)
```

```
SS1
```

```
SS2<-SS(F2,F_observed,n)
```

```
SS2
```

```
SS3<-SS(F3,F_observed,n)
```

```
SS3
```

```
SS4<-SS(F4,F_observed,n)
```

```
SS4
```

```
SS5<-SS(F5,F_observed,n)
```

```
SS5
```

```

op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l',
main="Probability Plots"
,xlab="Observed probability",ylab="Expected probability")
lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
legend("bottomright",cex=0.7,c("BTIIHLETLT(SS=0.0245)",
"TIIEHLW(SS=0.0400)","BW(SS=0.0369)",
"TTLW(SS=23.6407)","BOLU(SS=0.1959)"),
lty=c(1,1,1,1,1,1),
col=c("red","blue","purple","orange",
"green"),lwd=c(1,1,1,1,1,1))

```

CHAPTER 4 R CODES

Pdf and Hazard Plots

```

##TIIHLETLT-EXPONENTIAL DISTRIBUTION
TIIHLETLT_cdf<-function(alpha,beta,lambda,b,x){
(2*(1-(1-(1-exp(-(alpha*lambda*x)))^(beta))^(2))^(b))/
(1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))

}

TIIHLETLT_pdf<-function(alpha,beta,lambda,b,x){
(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))
}

x=seq(0,1.2,by=0.001)
y1=TIIHLETLT_pdf(2.2, 4.9, 2.5, 10.6, x)
plot(x,y1,ylim=c(0,006.0),col=2,'l',lwd=2
,xlab="x",ylab="pdf")
y2=TIIHLETLT_pdf(1.3, 0.9, 2.5, 2.6,x)
lines(x,y2,col=3,lwd=2)

```

```

y3=TIIHLTLEGE_pdf(2.95, 1.0, 1.5, 0.7,x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLEGE_pdf(1.5, 9.9, 2.5, 10.7, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLEGE_pdf(0.45, 8.2, 4.8, 8.02,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=2.2,', beta,'=4.9,
    ',lambda,'=2.5' ,b,'=10.6')),
  expression(paste(alpha,'=1.3,', beta,'=0.9,
    ',lambda,'=2.5' ,b,'=2.6')),
  expression(paste(alpha,'=2.95,', beta,'=1.0,
    ',lambda,'=1.5' ,b,'=0.7')),
  expression(paste(alpha,'=1.5,', beta,'=9.9,
    ',lambda,'=2.5' ,b,'=10.7')),
  expression(paste(alpha,'=0.45,', beta,'=8.2,
    ',lambda,'=4.8',b,'=8.02'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

## define TIIHLTLEGE hazard
TIIHLTLEGE_hazard <-function(alpha,beta,lambda,b,x){
  TIIHLTLEGE_pdf(alpha,beta,lambda,b,x)/
  (1 -TIIHLTLEGE_cdf(alpha,beta,lambda,b,x))
}

x=seq(0,3.5,by=0.001)
y1=TIIHLTLEGE_hazard(0.7, 9.65, 3.9,8.0, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2
, xlab="x",ylab="hrf")
y2=TIIHLTLEGE_hazard(1.6, 5.9, 0.8,7.1, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLEGE_hazard(1.0, 1.0, 1.4,3.0, x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLEGE_hazard(0.5, 1.9, 0.9,0.19, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLEGE_hazard(2.0, 5.9, 0.8,0.01, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=0.7,', beta,'=6.65,
    ',lambda,'=3.9', b,'=8.0')),
  expression(paste(alpha,'=1.6,', beta,'=5.9,

```

```

',lambda,'=0.8', b,'=7.1')),
expression(paste(alpha,'=1.0,', beta,'=1.0,
',lambda,'=1.4', b,'=3.0')),
expression(paste(alpha,'=0.5,', beta,'=1.9,
',lambda,'=0.9', b,'=0.19')),
expression(paste(alpha,'=2.0,', beta,'=5.9,
',lambda,'=0.8',b,'=0.01'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

##TIIHLEGLLoG LOGISTIC DISTRIBUTION

```

TIIHLEGLLoG_pdf<-function(alpha,beta,c,b,x)
{(4*b*c*alpha*beta*(x^(c-1))*((1+x^(c))^(-(2)))*
(1-(1-(1+x^(c))^(-(alpha))))^(beta))*((1+x^(c))^(1-alpha))*
((1-(1+x^(c))^(-(alpha))))^(beta-1))*
((1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^((2))^(b-1)))/
((1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^((2))^(b))^((2)))}

```

```

TIIHLEGLLoG_cdf<-function(alpha,beta,c,b,x){(2*
(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^((2))^(b))/
(1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^((2))^(b))}

```

```

x=seq(0,1.2,by=0.001)
y1=TIIHLEGLLoG_pdf(9.1, 1.4, 3.0,0.4, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLEGLLoG_pdf(3.1, 10.9, 2.3, 1.5,x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLEGLLoG_pdf(1.2, 8.9, 9.9, 3.5,x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLEGLLoG_pdf(0.9, 1.9, 1.2, 0.25, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLEGLLoG_pdf(2.9, 1.2, 1.9, 5.0,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
expression(paste(alpha,'=9.1,',beta,'=1.4,
',c,'=3.0',b,'=0.4')),
expression(paste(alpha,'=3.1,',beta,'=10.9,
',c,'=2.3',b,'=1.5')),
expression(paste(alpha,'=1.2,',beta,'=8.9,
',c,'=9.9',b,'=3.5'))),

```

```

expression(paste(alpha,'=0.9','beta','=1.9','
,c','=1.2','b','=0.25')),
expression(paste(alpha,'=2.9','beta','=1.2','
,c','=1.9','b','=5.0'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

## define TIIHLTLEGE hazard
TIIHLTLEGLLoG_hazard <-function(alpha,beta,b,c,x){
  TIIHLTLEGLLoG_pdf(alpha,beta,c,b,x)/
  (1 -TIIHLTLEGLLoG_cdf(alpha,beta,b,c,x))
}

x=seq(0,3.0,by=0.001)
y1=TIIHLTLEGLLoG_hazard(1.6, 4.9, 4.0,1.0, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',lwd=2,xlab="x",ylab="hrf")
y2=TIIHLTLEGLLoG_hazard(1.29, 1.0, 2.8, 2.8 ,x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLEGLLoG_hazard(1.0, 0.75, 6.0, 0.1,x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLEGLLoG_hazard(1.0, 2.8, 0.05, 2.7,x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLEGLLoG_hazard(0.6, 2.0, 2.0,7.9, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=1.6','beta','=4.9,
',c','=4.0','b','=1.0')),
  expression(paste(alpha,'=1.29','beta','=1.0,
',c','=2.8','b','=2.8')),
  expression(paste(alpha,'=1.0','beta','=0.75,
',c','=6.0','b','=0.1')),
  expression(paste(alpha,'=1.0','beta','=2.8,
',c','=0.05','b','=2.7')),
  expression(paste(alpha,'=0.6','beta','=2.0,
',c','=2.0','b','=7.9'))),
col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

##TIIHLTLEG-WEIBULL DISTRIBUTION
TIIHLTLEGW_pdf<-function(alpha,beta,b,x)
{(4*b*alpha*(beta^(2))*(x^(beta-1))*
(exp(-(alpha*x^(beta)))))*

```

$$\frac{((1-\exp(-(\alpha x^{\beta})))^{\beta-1}) * ((1-(1-\exp(-(\alpha x^{\beta}))))^{\beta}) * ((1-(1-(1-\exp(-(\alpha x^{\beta}))))^{\beta}))^2)^{b-1}}{((1+(1-(1-(1-\exp(-(\alpha x^{\beta}))))^{\beta}))^2)^b}^b$$

```
TIIHLEGTLEGW_cdf<-function(alpha,beta,b,x){
(2*(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))/
(1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))}

x=seq(0,1.2,by=0.001)
y1=TIIHLEGTLEGW_pdf(2.0, 1.3, 3.6, x)
plot(x,y1,ylim=c(0,6.0),col=2,'l',lwd=2,xlab="x",ylab="pdf")
y2=TIIHLEGTLEGW_pdf(6.9, 1.11, 20.9,x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLEGTLEGW_pdf(1.0, 0.96, 1.5,x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLEGTLEGW_pdf(0.9,0.9, 15.98, x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLEGTLEGW_pdf(18.0, 0.9, 2.55,x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=2.0,',
    beta,'=1.3,', b,'=3.6')),
  expression(paste(alpha,'=6.9,',
    beta,'=1.11,', b,'=20.9')),
  expression(paste(alpha,'=1.0,',
    beta,'=0.96,', b,'=1.5')),
  expression(paste(alpha,'=0.9,',
    beta,'=0.9,', b,'=15.98')),
  expression(paste(alpha,'=18.0,',
    beta,'=0.9,',b,'=2.55'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))
```

```
## define TIIHLEGTLEGE hazard
TIIHLEGTLEGW_hazard <-function(alpha,beta,b,x){
  TIIHLEGTLEGW_pdf(alpha,beta,b,x)/
  (1 -TIIHLEGTLEGW_cdf(alpha,beta,b,x))
}
```

```
x=seq(0,3.5,by=0.001)
y1=TIIHLEGTLEGW_hazard(1.1, 0.2, 0.27, x)
plot(x,y1,ylim=c(0,4.0),col=2,'l',
```

```

lwd=2,xlab="x",ylab="hrf")
y2=TIIHLTLEGW_hazard(0.9, 0.9, 1.9, x)
lines(x,y2,col=3,lwd=2)
y3=TIIHLTLEGW_hazard(2.3, 1.0, 10.0,x)
lines(x,y3,col=4,lwd=2)
y4=TIIHLTLEGW_hazard(1.0, 0.95, 0.19,x)
lines(x,y4,col=5,lwd=2)
y5=TIIHLTLEGW_hazard(0.9, 1.5, 1.0, x)
lines(x,y5,col=6,lwd=2)
legend("topright",c(
  expression(paste(alpha,'=1.1,',
    beta,'=0.2,', b,'=0.27')),
  expression(paste(alpha,'=0.9,',
    beta,'=0.9,', b,'=1.9')),
  expression(paste(alpha,'=2.3,',
    beta,'=1.0,', b,'=10.0')),
  expression(paste(alpha,'=1.0,',
    beta,'=0.95,', b,'=0.19')),
  expression(paste(alpha,'=0.9,',
    beta,'=1.5,', b,'=1.0'))),
  col=c(2,3,4,5,6),lwd=c(2,2,2,2,2))

```

Quantiles,Moments and Simulation

```

rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)
library(zipfR)

##define pdf TIIHLTLEGE

TIIHLTLEGE_pdf<-function(alpha,beta,lambda,b,x)
{
  (4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
    ((exp(-(lambda*x)))^(alpha-1))*
    ((1-exp(-(lambda*x*alpha)))^(beta-1))*
    (1-(1-exp(-(lambda*x*alpha)))^(beta))*
    ((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1))))/

```

```
((1+(1-(1-(1-exp(-(lambda*x*alpha))))^(beta))^(2))^(b))^(2))
}
```

```
##define cdf TIIHLTLEGE
TIIHLTLEGE_cdf=function(alpha,beta,lambda,b,x){
  y= (2*(1-(1-(1-exp(-(alpha*lambda*x))))^(beta))^(2))^(b))
  /(1+(1-(1-(1-exp(-(lambda*alpha*x))))^(beta))^(2))^(b))
}
```

```
TIIHLTLEGE_neglogl=function(alpha,beta,lambda,b,x){
  -sum(log(TIIHLTLEGE_pdf(alpha,beta,lambda,b,x)))}
```

```
TIIHLTLEGE_hazard=function(alpha,beta,lambda,b,x){
  f=TIIHLTLEGE_pdf(alpha,beta,lambda,b,x)
  F=TIIHLTLEGE_cdf(alpha,beta,lambda,b,x)
  y=f/(1-F)
  return(y)}
```

```
##-----
TIIHLTLEGE_quantile=function(parameter,u){
  alpha=parameter[1]
  beta=parameter[2]
  lambda=parameter[3]
  b=parameter[4]
  f=function(x){
    TIIHLTLEGE_cdf(alpha,beta,lambda,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=1000000000000,tol=0.0001))
  return(x)
}
```

```
TIIHLTLEGE_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0('(',paste0(x,collapse=', '),")")})}
```

```

Table_Quantile
for(iter in 1:size){
  parameter=parameter_matrix[iter,]
  for(i in 1:length(u)){
    Table_Quantile[i,iter]=TIIHLTLEGE_quantile(parameter,u[i])
  }
}
return(Table_Quantile)
}

```

```

#-----

```

```

## table of quantile

```

```

parameter_matrix=as.matrix(rbind(
  par1=c(1.5,1.5,1.5,1.5),
  par2=c(1.5,1.0,1.5,1.5),
  par3=c(1.5,0.5,1.5,1.0),
  par4=c(1.5,1.5,1.0,1.5),
  par5=c(1.0,1.5,1.0,0.5)

```

```

))

```

```

TIIHLTLEGE_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLEGE_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLEGE_QuantileTable(parameter_matrix))
sink()

```

```

#-----

```

```

#define pdf TIIHLTLEGE

```

```

TIIHLTLEGE_pdf=function(alpha,beta,lambda,b,x){
  y=(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
    ((exp(-(lambda*x)))^(alpha-1))*
    ((1-exp(-(lambda*x*alpha)))^(beta-1))*
    (1-(1-exp(-(lambda*x*alpha)))^(beta))*
    ((1-(1-(1-exp(-(lambda*x*alpha))))^(
    (beta))^(2))^(b-1))))/
  ((1+(1-(1-(1-exp(-(lambda*x*alpha))))^(beta))^(2))^(b))^(2))
}

```

```

TIIHLTLEGE_moments=function(alpha,beta,lambda,b,n){
  f=function(alpha,beta,lambda,b,n,x){
    (x^n)*(TIIHLTLEGE_pdf(alpha,beta,lambda,b,x))
  }
  y=integrate(f,lower=0,upper=1,subdivisions=10
    ,alpha=alpha,beta=beta,lambda=lambda,b=b,n=n)
  return(y$value)
}

### table of moments
parameter_matrix_Moments=matrix(c(1.8,0.8,0.7,0.5,2.0,2.5
,0.9,0.5,0.5,2.0,1.5,1.5,0.9,
5.0,1.8,0.5,0.8,4.5,1.6,0.4),ncol=4,byrow=T)
parameter_matrix_Moments
list_Moments=c('EX','EX2','EX3','EX4','EX5','EX6'
,'SD','CV','CS','CK')
Table_Moments=matrix(0,nrow=length(list_Moments),
ncol=dim(parameter_matrix_Moments)[1])
row.names(Table_Moments)=list_Moments
Table_Moments

for(iter in 1:dim(parameter_matrix_Moments)[1]){
  parameter=parameter_matrix_Moments[iter,]
  for(i in 1:10){
    Table_Moments[i,iter]=TIIHLTLEGE_moments(
      parameter[1],parameter[2]
      ,parameter[3],parameter[4],i)
  }
}

sink("C:/Users/DELL/Desktop/SIMULATIONS1/TIIHLTLEGE_Moments_table.txt")
print(Table_Moments)
###moments end

##-----
#####

TIIHLTLEGE_simulation=function(size=c(25,50,
100,200,400,800,1000),samp,par){
  set.seed(1000)
  K=length(par)

```

```

Mean=vector()
RMSE=vector()
Bias=vector()
samplesize=as.vector(t(mapply(rep,size,K)))
for(iter_size in 1:length(size)){
  coef1=matrix(NA,samp,K)
  colnames(coef1)=c('alpha','beta','lambda','b')
  for(nsamp in 1:samp){
    tryCatch(
      {
        x1_TIIHLTLEGE=NULL
        q=runif(size[iter_size],0,1)
        x1=sapply(q,TIIHLTLEGE_quantile,parameter=par)
        # MLE
        x1_TIIHLTLEGE=mle2(TIIHLTLEGE_neglogl,
          start=list(alpha=par[1],beta=par[2],
            lambda=par[3],b=par[4]),
          method="L-BFGS-B",data=list(x=x1),
          lower=c(alpha=0,beta=0,lambda=0,b=0),
          upper=c(alpha=Inf,beta=Inf,lambda=Inf,b=Inf)
          ,use.ginv=TRUE)
        coef1[nsamp,]=coef(x1_TIIHLTLEGE)
      },error=function(e){})
  }
  Mean[length(size)*(0:(K-1))+iter_size]=
  apply(coef1,2,mean,na.rm=TRUE)
  RMSE[length(size)*(0:(K-1))+iter_size]=
  apply((coef1-matrix(rep(par,nsamp),ncol=K,byrow=T))^2
    ,2,function(x){sqrt(mean(x,na.rm=TRUE))})
}
Bias=as.vector(sapply(1:K,function(x)
  {Bias[(length(size)*(x-1)+1):(length(size)*x)]
  =Mean[(length(size)*(x-1)+1):(length(size)*x)]
  -par[x]})) return(cbind\
(samplesize,Mean,RMSE,Bias))
}

```

#-----

```

sink("C:/Users/DELL/Desktop/SIMULATIONS1/TIIHLTLEGE_simula

tion_new_full1.txt",append=F,split=F)
par=c(1.5,1.0,0.5)
par_p=c(3.0)
for(i in 1:length(par)){
  for(j in 1:length(par)){
    for(p in 1:length(par)){
      for(q in 1:length(par_p)){
        print(c(par[i],par[j],par[p],par_p[q]))
        print(TIIHLTLEGE_simulation(samp=1000
          ,par=c(par[i],par[j],par[p],par_p[q])))
      }
    }
  }
}

sink()
-
-----

#####TIIHLTLEG_LLoG
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)
library(zipfR)

##define pdf TIIHLTLEG_LLoG

TIIHLTLEG_LoG_pdf<-function(alpha,beta,c,b,x)
{(4*b*c*alpha*beta*(x^(c-1))*
((1+x^(c))^(-(2))))*
(1-(1-(1+x^(c))^(-(alpha))))^(beta))*((1+x^(c))^(1-alpha))*
((1-(1+x^(c))^(-(alpha))))^(beta-1))*
((1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))^(2))}

##define cdf TIIHLTLEG_LLoG

```

```

TIIHLTLEG_Log_cdf<-function(alpha,beta,c,b,x)
{(2*(1-(1-(1-(1+x^(c))^(-(alpha)))^(beta))^(2))^(b))/
(1+(1-(1-(1-(1+x^(c))^(-(alpha)))^(beta))^(2))^(b))}

TIIHLTLEG_Log_neglogl=function(alpha,beta,c,b,x){
  -sum(log(BTIIHRTL_Log_pdf(alpha,beta,c,b,x)))}

TIIHLTLEG_Log_hazard=function(alpha,beta,c,b,x){
  f=TIIHLTLEG_Log_pdf(alpha,beta,c,b,x)
  F=TIIHLTLEG_Log_cdf(alpha,beta,c,b,x)
  y=f/(1-F)
  return(y)}

##-----
## define TIIHLTLEG_LLoG quantile
TIIHLTLEG_Log_quantile =function(parameter,u){
  alpha=parameter[1]
  beta=parameter[2]
  c=parameter[3]
  b=parameter[4]

  f=function(x){
    TIIHLTLEG_Log_cdf(alpha,beta,c,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=10,tol=0.0001))
  return(x)
}

TIIHLTLEG_Log_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1,
  function(x){paste0(' ',paste0(x,collapse=' '),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHLTLEG_Log_quantile
      (parameter,u[i])
    }
  }
}

```

```

    }
  }
  return(Table_Quantile)
}
#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(1.8,1.8,1.7,1.5),
  par2=c(2.0,2.5,0.9,0.5),
  par3=c(0.5,0.5,1.5,1.5),
  par4=c(0.9,1.0,1.8,0.5),
  par5=c(1.8,1.5,1.9,1.4)

))

#TIIHLTLEG_Log_QuantileTable(parameter_matrix)
#print(parameter_matrix)
print(TIIHLTLEG_Log_QuantileTable(parameter_matrix))
TIIHLTLEG_Log_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLEG_Log_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLEG_Log_QuantileTable(parameter_matrix))
sink()

##End of Quantile
#-----

##define pdf TIIHLTLEG_LLoG
TIIHLTLEG_Log_pdf=function(alpha,beta,c,b,x){
  y=(4*b*c*alpha*beta*(x^(c-1))*((1+x^(c))^(-(2)))*
  (1-(1-(1+x^(c))^(-(alpha))))^(beta))*((1+x^(c))^(1-alpha))*
  ((1-(1+x^(c))^(-(alpha))))^(beta-1))*
  ((1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b-1)))/
  ((1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))^(2))
}

TIIHLTLEG_Log_moments=function(alpha,beta,c,b,n){
  f=function(alpha,beta,c,b,n,x){
    (x^n)*(TIIHLTLEG_Log_pdf(alpha,beta,c,b,x))
  }
}

```

```

y=integrate(f,lower=0,upper=1,subdivisions=10000,
alpha=alpha,beta=beta,c=c,b=b,n=n)
return(y$value)
}

### table of moments
parameter_matrix_Moments=matrix(c(0.8,1.3,1.9,1.4,0.9,0.5,
1.6,1.5,0.5,2.0,1.5,1.5,0.9,0.9,1.8,0.5,0.8,1.5,1.6,1.4)
,ncol=4,byrow=T)
parameter_matrix_Moments
list_Moments=c('EX','EX2','EX3','EX4','EX5','EX6',
'SD','CV','CS','CK')
Table_Moments=matrix(0,nrow=length(list_Moments),
ncol=dim(parameter_matrix_Moments)[1])
row.names(Table_Moments)=list_Moments
Table_Moments

for(iter in 1:dim(parameter_matrix_Moments)[1]){
  parameter=parameter_matrix_Moments[iter,]
  for(i in 1:10){
    Table_Moments[i,iter]=TIIHLTLEG_Log_moments(
      parameter[1],parameter[2],parameter[3],parameter[4],i)
  }
}
print(Table_Moments)
###moments end
-----

#####TIIHLTLEGW
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(rootSolve)

##define TIIHLTLEGW pdf
TIIHLTLEGW_pdf<-function(alpha,beta,b,x){(4*b*alpha*
(beta^(2))*(x^(beta-1))*(exp(-(alpha*x^(beta)))))*
((1-exp(-(alpha*x^(beta))))^(beta-1))*
((1-(1-exp(-(alpha*x^(beta))))^(beta)))*
((1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))))^

```

```
(beta))^(2))^(b))^(2))}
```

```
##define TIIHLTLEGW cdf
TIIHLTLEGW_cdf<-function(alpha,beta,b,x){(2*
(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))/
(1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))}
```

```
## define TIIHLTLEGW hazard
TIIHLTLEGW_hazard <-function(alpha,beta,b,x){
  TIIHLTLEGW_pdf(alpha,beta,b,x)/
  (1 -TIIHLTLEGW_cdf(alpha,beta,b,x))
}
```

```
TIIHLTLEGW_neglogl=function(alpha,beta,b,x){
  -sum(log(TIIHLTLEGW_pdf(alpha,beta,b,x)))
}
```

```
TIIHLTLEGW_hazard2=function(alpha,beta,b,x){
  f=TIIHLTLEGW_pdf(alpha,beta,b,x)
  F=TIIHLTLEGW_cdf(alpha,beta,b,x)
  y=f/(1-F)
  return(y)}
```

```
#-----
TIIHLTLEGW_quantile=function(parameter,u){
  alpha=parameter[1]
  beta=parameter[2]
  b=parameter[3]
  f=function(x){
    TIIHLTLEGW_cdf(alpha,beta,b,x)-u
  }
  x=min(uniroot.all(f,lower=0,upper=
1000000000000,tol=0.0001))
  return(x)
}
```

```

TIIHLTLEGW_QuantileTable=function(parameter_matrix){
  u=seq(0.1,0.9,0.1)
  size=dim(parameter_matrix)[1]
  Table_Quantile=matrix(NA,nrow=length(u),ncol=size)
  row.names(Table_Quantile)=u
  colnames(Table_Quantile)=apply(parameter_matrix,1
  ,function(x){paste0('(',paste0(x,collapse=','),")")})
  Table_Quantile
  for(iter in 1:size){
    parameter=parameter_matrix[iter,]
    for(i in 1:length(u)){
      Table_Quantile[i,iter]=TIIHLTLEGW_quantile(parameter,u[i])
    }
  }
  return(Table_Quantile)
}

```

```

#-----
## table of quantile
parameter_matrix=as.matrix(rbind(
  par1=c(0.6,1.0,0.5),
  par2=c(0.9,1.0,1.0),
  par3=c(0.9,0.7,1.8),
  par4=c(0.9,1.0,0.5),
  par5=c(0.9,0.9,1.1)

))
TIIHLTLEGW_QuantileTable(parameter_matrix)
sink("C:/Users/DELL/Desktop/SIMULATIONS1/
TIIHLTLEGW_quantile_table.txt")
print(parameter_matrix)
print(TIIHLTLEGW_QuantileTable(parameter_matrix))
sink()
#-----

```

Skewness and Kurtosis Plots

```

TIIHLTLOG<-function(alpha,beta,c,b)
{(4*b*c*alpha*beta*(x^(c-1))*((1+x^(c))^(-(2)))*
(1-(1-(1+x^(c))^(-(alpha))))^(beta))*((1+x^(c))^(1-alpha))*
((1-(1+x^(c))^(-(alpha))))^(beta-1))*

```

```

((1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))^(2))}

TIIHLTLEG_Log_cdf<-function(alpha,beta,c,b)
{(2*(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))/
(1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))}

#####TIIHLTLEG_LLoG
library(plot3Drgl)
library(zipfR)
F=function(x,par){
  alpha=par[1];beta =par[2];c= par[3];b=par[4]
  C =(2*(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))/
  (1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))
  return(C)
}
#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2];c= par[3];b=par[4]
  D =(4*b*c*alpha*beta*(x^(c-1))*((1+x^(c))^(-(2)))*
  (1-(1-(1+x^(c))^(-(alpha))))^(beta))*((1+x^(c))^(1-alpha))*
  ((1-(1+x^(c))^(-(alpha))))^(beta-1))*
  ((1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b-1)))/
  ((1+(1-(1-(1-(1+x^(c))^(-(alpha))))^(beta))^(2))^(b))^(2))

  return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
beta=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(beta,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c( 6.9,beta,3.6,b))
#Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
#return(Kurtosis)}

Skewness=skewness(U)
return(Skewness)}

```

```

#Skew
G_kurt<-Vectorize(G_kurt, c('beta', 'b'))
#Kurtosis<-outer(beta,b,G_kurt)
Skewness<-outer(beta,b,G_kurt)

persp3D(beta,b, Skewness,main=
expression(paste(TIIHLEGTLLoG(6.9,beta,3.6, b)))
,xlab="beta",ylab="b",zlab="Skewness")

#####
TIIHLEGTLLoG_pdf<-function(alpha,beta,b){(4*b*alpha*
(beta^(2))*(x^(beta-1))
*(exp(-(alpha*x^(beta))))*((1-exp(-(alpha*x^(
(beta))))^(beta-1))*
((1-(1-exp(-(alpha*x^(beta))))^(beta))*
((1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(
(2))^(b-1)))/((1+(1-(1-(1-exp(-(alpha*x^(beta))))^(
(beta))^(2))^(b))^(2))})

TIIHLEGTLLoG_cdf<-function(alpha,beta,b)
{(2*(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))/
(1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))}
####TIIHLEGTLLoG_W
library(plot3Drgl)
library(zipfR)
F=function(x,par){
  alpha=par[1];beta =par[2];b= par[3]
  C =(2*(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))/
  (1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))
return(C)
}
#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2];b= par[3]
  D =(4*b*alpha*(beta^(2))*(x^(beta-1))*
  (exp(-(alpha*x^(beta))))*((1-exp(-(alpha*x^(beta))))^(beta-1))*
  ((1-(1-exp(-(alpha*x^(beta))))^(beta))*
  ((1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b-1)))/
  ((1+(1-(1-(1-exp(-(alpha*x^(beta))))^(beta))^(2))^(b))^(2))

return(D)
}

```

```

#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
beta=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(beta,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.5,beta,b))
Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
return(Kurtosis)}

#Skewness=skewness(U)
#return(Skewness)}

#Skew
G_kurt<-Vectorize(G_kurt, c('beta', 'b'))
Kurtosis<-outer(beta,b,G_kurt)
#Skewness<-outer(beta,b,G_kurt)

persp3D(beta,b, Kurtosis,main=expression
(paste(TIIHLTLEG_W(0.5,beta, b)))
,xlab="beta",ylab="b",zlab="Kurtosis")

#####
TIIHLTLEGE_pdf <- function(alpha,beta,lambda,b)
{-sum(log((4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))))))}

TIIHLTLEGE_cdf<-function(par,x){

  alpha=par[1]
  beta=par[2]
  lambda=par[3]
  b=par[4]

(2*(1-(1-(1-exp(-(alpha*lambda*x)))^(beta))^(2))^(b))/

```

```

(1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))

}

#####TIIHLEGTLEG_E
library(plot3Drgl)
library(zipfR)
F=function(x,par){
  alpha=par[1];beta =par[2]; lambda= par[3];b=par[4]
  C =(2*(1-(1-(1-exp(-(alpha*lambda*x)))^(beta))^(2))^(b))/
  (1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))

return(C)
}
#PDF function:
f = function(x,par){
  alpha=par[1];beta =par[2]; lambda= par[3];b=par[4]
  D =(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*(
  ((exp(-(lambda*x)))^(alpha-1))*
  ((1-exp(-(lambda*x*alpha)))^(beta-1))*
  (1-(1-exp(-(lambda*x*alpha)))^(beta))*
  ((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1))))/
  ((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))
return(D)
}
#Hazard function:
h = function(x,par) f(x,par)/(1-F(x,par))
#The program is developed in R to obtain the value of skewness:
library(moments)
lambda=seq(1,7,.1)
b=seq(1,7,.1)
G_kurt<- function(lambda,b) {
  x <- seq(0, 10, le = 2*10^4)
  U=f(x,c(0.9,3.6,lambda,b))
Kurtosis= kurtosis(U) # or kurt =Kurtosis(U)
return(Kurtosis)}

#Skewness=skewness(U)
#return(Skewness)}

```

```

#Skew
G_kurt<-Vectorize(G_kurt, c('lambda', 'b'))
Kurtosis<-outer(lambda,b,G_kurt)
#Skewness<-outer(lambda,b,G_kurt)

persp3D(lambda,b, Kurtosis,main=
expression(paste(TIIHLTLEG_E
(0.9,3.6,lambda,b))),xlab="lambda"
,ylab="b",zlab="Kurtosis")

```

Applications and Goodness of Fit

```

#####TIIHLTLEGE APPLICATION1 (FAIULRE TIMES OF AC SYSTEM)
rm(list=ls())
library(stats4)
library(bbmle)
library(stats)
library(numDeriv)
library(zipfR)

#Failure times of the air conditioning system of an airplane data set
x<-c(
1, 3, 5, 7, 11, 11, 11, 12, 14, 14, 14,
16, 16, 20, 21, 23, 42, 47 ,52, 62 ,71,
71 ,87, 90 ,95 ,120 ,120 ,225, 246, 261
)

TIIHLTLEGE_pdf <- function(alpha,beta,lambda,b)
{-sum(log((4*b*lambda*alpha*beta*(exp(-(lambda*x))))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))))}

mle.result<-mle2(TIIHLTLEGE_pdf ,hessian = NULL,
start=list(alpha=0.00117, beta=0.000019,
lambda=0.008, b=2.689),optimizer="nlminb",lower=0)
summary(mle.result)
vcov(mle.result)
output1<-vcov(mle.result)
1.96*sqrt(diag(output1))

```



```
#####EHLPGWL
```

```
EHLPGWL_LL<-function(alpha,beta,delta,c){-sum(log(
(2*delta*alpha*beta*
c*(x^(c-1))*((1+x^c)^(-2))*
((1-(1+x^c)^(-1))^(alpha-1))*((1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^(beta-1))*
(exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^(beta))*
((1-exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^(beta))
^(delta-1)))/
(((1+x^c)^(-alpha-1))*
((1+exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^(beta))
^(delta+1))))))}
mle.result<-mle2(EHLPGWL_LL,hessian = NULL,
start=list(alpha=2.000000000902175,
beta=0.95190 ,delta=1.5,c=.178770),
optimizer="nlminb",lower=0)
summary(mle.result)
```

```
#####HLBXII
```

```
HLBXII_LL<-function(alpha,lambda,a,b){-sum(log((2*alpha*lambda*a
*b*(x^(a-1))*(exp(lambda*((1-(1+x^a)^b))))*
(1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1))/
(((1+x^a)^(-b+1))*(1+exp(lambda*(1-(1+x^a)^b)))*(alpha+1))))}
mle.result<-mle2(HLBXII_LL,hessian =
NULL,start=list(alpha=.009008219,
lambda=10.004121,a=.000000000000000005887,
b=.987259),optimizer="nlminb",lower=0)
summary(mle.result)
```

```
#####(TIIHLTLEGE APLPLICATION2(SURVIVAL TIMES OF GUINEA PIGS)
```

```
#Survival Times of Guinea Pigs
```

```
x<-c(
12, 15, 22, 24, 24, 32, 32, 33, 34,
38, 38, 43, 44, 48, 52, 53, 54, 54,
55, 56, 57, 58, 58, 59, 60,
60, 60, 60, 61, 62, 63, 65, 65, 67,
68, 70, 70, 72, 73, 75, 76, 76, 81,
83, 84, 85, 87,
91, 95, 96, 98, 99, 109, 110, 121,
127, 129, 131, 143, 146, 146, 175,
175, 211, 233, 258, 258, 263, 297, 341, 341, 376
```

)

```
TIIHLTLEGE_pdf <- function(alpha,beta,lambda,b)
{-sum(log((4*b*lambda*alpha*beta*(exp(-(lambda*x)))*)
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(
(beta))^(2))^(b-1))))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(
(beta))^(2))^(b))^(2))))})
mle.result<-mle2(TIIHLTLEGE_pdf ,
hessian = NULL,
start=list(alpha=0.00117,
beta=0.000019,lambda=0.008,
b=2.689),optimizer="nlminb",lower=0)
summary(mle.result)
vcov(mle.result)
output1<-vcov(mle.result)
1.96*sqrt(diag(output1))

#####TLWLx
TLWLx_LL<-function(a,b,alpha,theta)
{-sum(log(2*theta*alpha*a*
b*((1+b*x)^(a*alpha-1))*
(((1-(1+b*x)^(-a))^(alpha-1)))*)
(exp(-2*(((1-(1+b*x)^(-a)))/
(1+b*x)^(-a))^alpha)))*)
(1-exp(-2*(((1-(1+b*x)^(-a)))/
(1+b*x)^(-a))^alpha)))^(theta-1))})
mle.result<-mle2(TLWLx_LL,hessian = NULL,
start=list(a=1,b=1.6,alpha=.891,theta=0.1),
optimizer="nlminb",lower=0)
summary(mle.result)

#####TIIGIE-Lx
TIIGIE_Lx_pdf<- function(lambda,alpha,a,b) {
-sum(log(lambda*alpha*(a/b)*((1+(x/b))^(-(a+1)))*
((1+(x/b))^(a*(alpha+1)))*exp(lambda*(1-(1+(x/b))^(a*alpha)))))
```

```

))
}

m.result<-mle2(TIIGIE_Lx_pdf,hessian = NULL,start=list(
lambda=3.999999999999999909801,alpha=1.00010,
a=0.1,b=0.0000000000000000001),optimizer="nlminb",lower=0)
summary(m.result)

#####TII-EHL-TL-WL

TIIEHLTLWL_pdf<- function(a,b,theta, lambda) {
  -sum(log(((4*a*b*theta*lambda*(x^(lambda-1))*
    (exp(-x^lambda))*((1-exp(-2*x^lambda))^(b-1))*
    (exp(-x^lambda)))/((1+(1-exp(-2*x^lambda))^b)^(a+1)))*
    ((1-(1-exp(-2*x^lambda))^b)^(a-1))*((1-theta*
    ((1-(1-exp(-2*x^lambda))^b)/
    ((1+(1-exp(-2*x^lambda))^b)))^a)^(-1))/(-log(1-theta)))
  )))}

mai.result<-mle2(TIIEHLTLWL_pdf,hessian =
  NULL,start=list(a=0.001910290,
  b=.00000000098020,theta=0.000000000010,
  lambda=.0000001),optimizer="nlminb",lower=0)
summary(mai.result)

#####EHLPGWL

EHLPGWL_LL<-function(alpha,beta,delta,c)
{-sum(log((2*delta*alpha*beta*c*
(x^(c-1))*((1+x^c)^(-2))*((1-(1+x^c)^(-1))^(alpha-1))*
((1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^(beta-1))*
(exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))*
((1-exp(1-(1+((1-(1+x^c)^(-1)))/
(1+x^c)^(-1))^alpha)^beta))^(delta-1)))/
(((1+x^c)^(-alpha-1))*((1+exp(1-(1+((1-(1+x^c)^(-1)))/
(1+x^c)^(-1))^alpha)^beta))^(delta+1)))))}

mle.result<-mle2(EHLPGWL_LL,hessian = NULL,start=list
(alpha=2.000000000902175, beta=0.95190 ,delta=1.5,
c=.178770),optimizer="nlminb",lower=0)
summary(mle.result)

#####HLBXII_LL

HLBXII_LL<-function(alpha,lambda,a,b)
{-sum(log((2*alpha*lambda*a*b*(x^(a-1))*

```

```

(exp(lambda*((1-(1+x^a)^b))))*
(1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1)/(((1+x^a)^(-b+1))*
(1+exp(lambda*(1-(1+x^a)^b))*(alpha+1))))}
mle.result<-mle2(HLBXII_LL,hessian = NULL,start=list
(alpha=.009008219,lambda=10.004121,a=.000000000000000005887,
b=.987259),optimizer="nlminb",lower=0)
summary(mle.result)
#####
library(AdequacyModel)
library(zipfR)
##### SURVIVAL TIMES OF GUINEA PIGS GOODNESS OF FIT
data=c(12, 15, 22, 24, 24, 32, 32, 33, 34, 38, 38, 43, 44,
48, 52, 53, 54, 54, 55, 56, 57, 58, 58, 59, 60, 60, 60, 60, 61,
62, 63, 65, 65, 67, 68, 70, 70, 72, 73, 75, 76, 76, 81, 83, 84,
85, 87, 91, 95, 96, 98, 99, 109, 110, 121, 127, 129, 131,
143, 146, 146, 175, 175, 211, 233, 258, 258, 263, 297, 341, 341, 376)

TIIHLTLEGE_pdf<-function(par,x){

  alpha=par[1]
  beta=par[2]
  lambda=par[3]
  b=par[4]
(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))
}
TIIHLTLEGE_cdf<-function(par,x){

  alpha=par[1]
  beta=par[2]
  lambda=par[3]
  b=par[4]

(2*(1-(1-(1-exp(-(alpha*lambda*x)))^(beta))^(2))^(b))/
(1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))

```

```

}

goodness.fit(pdf=TIIHLTLEGE_pdf, cdf=TIIHLTLEGE_cdf,mle=c
( 2.4828e+01,2.2748e-02, 1.9383e-04 ,1.5640e+03),
data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),

      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

#####TWLx
TLWLx_pdf<-function(par,x){

  a=par[1]
  b=par[2]
  alpha=par[3]
  theta=par[4]

  2*theta*alpha*a*b*((1+b*x)^(a*alpha-1))*
  (((1-(1+b*x)^(-a))^(alpha-1)))*
  (exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha)))*
  (1-exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha)))^(theta-1)
}

TLWLx_cdf<-function(par,x){

  a=par[1]
  b=par[2]
  alpha=par[3]
  theta=par[4]

  (1-exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha)))^theta
}

goodness.fit(pdf=TLWLx_pdf, cdf=TLWLx_cdf,mle=c(
0.135823 ,3.070082,2.413841,8.141779),data = data,method = "BFGS",

```

```

        domain = c(0,100), lim_inf = c(0,0,0,0),
        lim_sup = c(1000000000000,
        1000000000000,1000000000000,1000000000000))
#####
####TIIGIE-Lx
TIIGIE_Lx_pdf<-function(parameter,x){
  lambda=parameter[1]
  alpha=parameter[2]
  a=parameter[3]
  b=parameter[4]
  lambda*alpha*(a/b)*((1+(x/b))^(-(a+1)))*((1+(x/b))^(a*(alpha+1)))*
  exp(lambda*(1-(1+(x/b))^(a*alpha)))

}

TIIGIE_Lx_cdf<-function(parameter,x){
  lambda=parameter[1]
  alpha=parameter[2]
  a=parameter[3]
  b=parameter[4]
  1-exp(lambda*(1-(1+(x/b))^(a*alpha)))

}

goodness.fit(pdf=TIIGIE_Lx_pdf,cdf= TIIGIE_Lx_cdf,mle=c(
0.040843 , 1.891761 , 0.428481 , 1.000007 ),
data=data,method = "BFGS",domain =
c(0,100),lim_inf = c(0,0,0,0),
        lim_sup = c(1000000000000,1000000000000
        ,1000000000000,1000000000000))

#####
####TII-EHL-TL-WL
TIIHLTLWL_pdf<-function(par,x){
  a=par[1]
  b=par[2]
  theta=par[3]
  lambda=par[4]

  ((4*a*b*theta*lambda*(x^(lambda-1))*(exp(-x^lambda))*
  ((1-exp(-2*x^lambda))^(b-1))*(exp(-x^lambda)))/
  ((1+(1-exp(-2*x^lambda))^b)^(a+1)))*

```

```

((1-(1-exp(-2*x^lambda))^b)^(a-1))*((1-theta*
((1-(1-exp(-2*x^lambda))^b)/
((1+(1-exp(-2*x^lambda))^b)))^a)^(-1))/(-log(1-theta)))
}

TIIHLLTLWL_cdf<-function(parameter,x){
  a=parameter[1]
  b=parameter[2]
  theta=parameter[3]
  lambda=parameter[4]

  1-((-log(1-theta*((1-(1-exp(-2*x^lambda))^b)/
((1+(1-exp(-2*x^lambda))^b)))^a))/(-log(1-theta)))

}

goodness.fit(pdf=TIIHLLTLWL_pdf,cdf=TIIHLLTLWL_cdf,mle=c(
1.0019e+00,9.9876e-01,1.3301e-09,4.9717e-02),
data=data,method = "BFGS",domain = c(0,100),lim_inf = c(0,0,0,0),
      lim_sup = c(100000000000000,
      100000000000000,100000000000000,100000000000))

#####
HLBXII_pdf<-function(par,x){

                                alpha=par[1]
                                lambda=par[2]
                                a=par[3]

  b=par[4]
  (2*alpha*lambda*a*b*(x^(a-1))*(exp(lambda*((1-(1+x^a)^b))))*
  (1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1))/(((1+x^a)^(-b+1))*
  (1+exp(lambda*(1-(1+x^a)^b)))*(alpha+1))
}

HLBXII_cdf<-function(par,x){

```

```

alpha=par[1]
lambda=par[2]
a=par[3]
b=par[4]
((1-exp((lambda*(1-(1+x^a)^b))))/(1+exp((lambda*(1-(1+x^a)^b))))^alpha
}

```

```

goodness.fit(pdf=HLBXII_pdf, cdf=HLBXII_cdf,
mle=c( 8.0381e-01 ,1.9748e-05,1.8769e-01 e-02
,8.8644e+00),data = data,method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

```

```
#####
```

```

EHLPGWL_pdf<-function(par,x){

alpha=par[1]
beta=par[2]
delta=par[3]
c=par[4]

(2*delta*alpha*beta*c*(x^(c-1))*((1+x^c)^(-2))*
((1-(1+x^c)^(-1))^(alpha-1))*
((1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^(beta-1))*
(exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))*
((1-exp(1-(1+((1-(1+x^c)^(-1))/(
(1+x^c)^(-1))^alpha)^beta))^(delta-1)))/
(((1+x^c)^(-alpha-1))*((1+exp(1-(1+((1-(1+x^c)^(-1))/(
(1+x^c)^(-1))^alpha)^beta))^(delta+1)))
}

```

```

EHLPGWL_cdf<-function(par,x){

```

```

alpha=par[1]

```

```

beta=par[2]
delta=par[3]
c=par[4]

((1-exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))/
(1+exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta)))^delta
}

goodness.fit(pdf=EHLPGWL_pdf, cdf=EHLPGWL_cdf,mle=c(0.54993
, 0.59521 , 16.98723 , 1.09348 ),data = data,method = "BFGS",
domain = c(0,1), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,10000000000000000
,10000000000000000,10000000000000000))

#####FAILURE TIMES OF AC SYSTEM GOODNESS OF FIT
data=c(1, 3, 5, 7, 11, 11, 11, 12, 14, 14, 14, 16, 16,
20, 21, 23, 42, 47 ,52, 62 ,71, 71 ,87, 90 ,95 ,120
,120 ,225, 246, 261
)
TIIHLEGE_pdf<-function(par,x){

alpha=par[1]
beta=par[2]
lambda=par[3]
b=par[4]
(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))
*((1-(1-(1-exp(-(lambda*x*alpha)))^(beta)))^
(2))^(b-1)))/((1+(1-(1-(1-exp(-(lambda*x*alpha)))^
(beta))^(2))^(b))^(2))

```

```

}
TIIHLTLEGE_cdf<-function(par,x){

  alpha=par[1]
  beta=par[2]
  lambda=par[3]
  b=par[4]

  (2*(1-(1-(1-exp(-(alpha*lambda*x)))^(beta))^(2))^(b))/
  (1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))

}

goodness.fit(pdf=TIIHLTLEGE_pdf, cdf=TIIHLTLEGE_cdf,mle=c( 2.4828e+01,
2.2748e-02,1.9383e-04 , 1.5640e+03  ),data = data,method = "BFGS",
      domain = c(0,1), lim_inf = c(0,0,0,0),
      lim_sup = c(10000000000000000,
10000000000000000,10000000000000000))

#####TWLx
TLWLx_pdf<-function(par,x){

  a=par[1]
  b=par[2]
  alpha=par[3]
  theta=par[4]

  2*theta*alpha*a*b*((1+b*x)^(a*alpha-1))*
  (((1-(1+b*x)^(-a))^(alpha-1)))*(exp(-2*(((1-(1+b*x)^
  (-a))/
  (1+b*x)^(-a))^alpha)))*(1-exp(-2*(((1-(1+b*x)^(-a))/
  (1+b*x)^(-a))^alpha)))^(theta-1)
}

TLWLx_cdf<-function(par,x){

```

```

a=par[1]
b=par[2]
alpha=par[3]
theta=par[4]

(1-exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha)))^theta
}

goodness.fit(pdf=TLWLx_pdf, cdf=TLWLx_cdf,mle=c( 0.14561
,0.74365,3.01451,0.65476),data = data,method = "BFGS",
          domain = c(0,100), lim_inf = c(0,0,0,0),
          lim_sup = c(1000000000000,1000000000000,
          10000000000000,1000000000000))
#####
####TIIGIE-Lx
TIIGIE_Lx_pdf<-function(parameter,x){
  lambda=parameter[1]
  alpha=parameter[2]
  a=parameter[3]
  b=parameter[4]
lambda*alpha*(a/b)*((1+(x/b))^(-(a+1)))*
((1+(x/b))^(a*(alpha+1)))*
exp(lambda*(1-(1+(x/b))^(a*alpha)))

}

TIIGIE_Lx_cdf<-function(parameter,x){
  lambda=parameter[1]
  alpha=parameter[2]
  a=parameter[3]
  b=parameter[4]
1-exp(lambda*(1-(1+(x/b))^(a*alpha)))

}

goodness.fit(pdf=TIIGIE_Lx_pdf,cdf=
TIIGIE_Lx_cdf,mle=c(127.67891,
0.22325 , 0.11067 , 134.65135 ),
data=data,method =
"BFGS",domain = c(0,100),

```

```

lim_inf = c(0,0,0,0),
lim_sup = c(1000000000000,1000000000000,
1000000000000,1000000000000))

#####EHLPGWL
EHLPGWL_pdf<-function(par,x){

  alpha=par[1]

  beta=par[2]
  delta=par[3]
  c=par[4]

  (2*delta*alpha*beta*c*(x^(c-1))*((1+x^c)^(-2))*
  ((1-(1+x^c)^(-1))^(alpha-1))*((1+((1-(1+x^c)^(-1))/
  (1+x^c)^(-1))^alpha)^(beta-1))*
  (exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))*
  ((1-exp(1-(1+((1-(1+x^c)^(-1))/
  (1+x^c)^(-1))^alpha)^beta))^(delta-1)))/
  (((1+x^c)^(-alpha-1))*
  ((1+exp(1-(1+((1-(1+x^c)^(-1))/
  (1+x^c)^(-1))^alpha)^beta))^(delta+1)))
  }

EHLPGWL_cdf<-function(par,x){

  alpha=par[1]

  beta=par[2]
  delta=par[3]
  c=par[4]

  ((1-exp(1-(1+((1-(1+x^c)^(-1))/
  (1+x^c)^(-1))^alpha)^beta))/
  (1+exp(1-(1+((1-(1+x^c)^(-1))/
  (1+x^c)^(-1))^alpha)^beta)))^delta
  }

```

```

goodness.fit(pdf=EHLPGWL_pdf, cdf=EHLPGWL_cdf,mle=c(2.512513
, 0.442807 , 2.658286 , 0.284138 ),
data = data,method = "BFGS",
domain = c(0,1), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,10000000000000000,
10000000000000000,10000000000000000))
#####
HLBXII_pdf<-function(par,x){

alpha=par[1]
lambda=par[2]
a=par[3]

b=par[4]
(2*alpha*lambda*a*b*(x^(a-1))*
(exp(lambda*((1-(1+x^a)^b))))*
(1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1))/
(((1+x^a)^(-b+1))*
(1+exp(lambda*(1-(1+x^a)^b)))*(alpha+1))
}

HLBXII_cdf<-function(par,x){

alpha=par[1]
lambda=par[2]
a=par[3]

b=par[4]
((1-exp((lambda*(1-(1+x^a)^b))))/(1+exp((lambda*(1-(1+x^a)^b))))^alpha
}

goodness.fit(pdf=HLBXII_pdf, cdf=HLBXII_cdf,
mle=c(8.0381e-01

```

```
,1.9748e-05,1.8769e-01 , 8.8644e+00 ),
data = data,method = "BFGS",
domain = c(0,100), lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,10000000000000000,
10000000000000000,10000000000000000))
#####
####TII-EHL-TL-WL
TIIHLTLWL_pdf<-function(par,x){
  a=par[1]
  b=par[2]
  theta=par[3]
  lambda=par[4]

  ((4*a*b*theta*lambda*(x^(lambda-1))*(exp(-x^lambda))*
  ((1-exp(-2*x^lambda))^(b-1))*(exp(-x^lambda)))/
  ((1+(1-exp(-2*x^lambda))^b)^(a+1)))*
  ((1-(1-exp(-2*x^lambda))^b)^(a-1))*
  (((1-theta*((1-(1-exp(-2*x^lambda))^b)/
  ((1+(1-exp(-2*x^lambda))^b)))^a)^(-1))/(-log(1-theta)))
}

TIIHLTLWL_cdf<-function(parameter,x){
  a=parameter[1]
  b=parameter[2]
  theta=parameter[3]
  lambda=parameter[4]

  1-((-log(1-theta*((1-(1-exp(-2*x^lambda))^b)/
  ((1+(1-exp(-2*x^lambda))^b)))^a))/(-log(1-theta)))

}
goodness.fit(pdf=TIIHLTLWL_pdf,cdf=TIIHLTLWL_cdf,
mle=c( 1.0019e+00,
9.9876e-01,1.3301e-09,4.9717e-02)
,data=data,method = "BFGS",domain = c(0,100),lim_inf = c(0,0,0,0),
lim_sup = c(10000000000000000,10000000000000000,
10000000000000000,10000000000000000))
#####
```

Fitted Densities and Probability Plots

```
##### AC SYSTEM FAILURE DATA FITTED DENSITIES
```

```

par(mfrow=c(0,1))
x<-c(1, 3, 5, 7, 11, 11, 11, 12, 14, 14, 14, 16,
    16, 20, 21, 23, 42, 47 ,52, 62 ,71, 71 ,87, 90
    ,95 ,120 ,120 ,225, 246, 261
)

#####TIIHLTLEGE
TIIHLTLEGE_pdf<-function(alpha, beta,lambda,b,x){
(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta))
^2))^b-1))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^2))^b-1))
}

#####HLBXII
HLBXII_pdf<-function(alpha,lambda,a,b,x){
(2*alpha*lambda*a*b*(x^(a-1))*(exp(lambda*((1-(1+x^a)^b))))*
(1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1))/(((1+x^a)^(-b+1))*
(1+exp(lambda*(1-(1+x^a)^b)))*(alpha+1))
}

#####TIIHSLTLWL
TIIHSLTLWL_pdf<-function(a,b,theta,lambda,x){
((4*a*b*theta*lambda*(x^(lambda-1))*(exp(-x^lambda))*
((1-exp(-2*x^lambda))^b-1)*(exp(-x^lambda)))/
((1+(1-exp(-2*x^lambda))^b)^(a+1)))*
((1-(1-exp(-2*x^lambda))^b)^(a-1))*
(((1-theta*((1-(1-exp(-2*x^lambda))^b)/
((1+(1-exp(-2*x^lambda))^b)))^a)^(-1))/(-log(1-theta)))
}

#####TLWLx
TLWLx_pdf<-function(a,b,alpha,theta,x){
2*theta*alpha*a*b*((1+b*x)^(a*alpha-1))*
(((1-(1+b*x)^(-a))^(alpha-1)))*
(exp(-2*(((1-(1+b*x)^(-a)))/(
1+b*x)^(-a))^alpha)))*(1-exp(-2*(((1-(1+b*x)^(-a)))/(1+b*x)^(-a))^alpha)))^(theta-1)
}

```

```
#####EHLPGWL
```

```
EHLPGWL_pdf<-function(alpha,beta,delta,c,x){
(2*delta*alpha*beta*c*(x^(c-1))*((1+x^c)^(-2))*
((1-(1+x^c)^(-1))^(alpha-1))*((1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^(beta-1))*
(exp(1-(1+((1-(1+x^c)^(-1)))/
(1+x^c)^(-1))^alpha)^beta))*
((1-exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))^(delta-1)))/
(((1+x^c)^(-alpha-1))*
((1+exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))^(delta+1)))
}
```

```
####TIIGIE-Lx
```

```
TIIGIE_Lx_pdf<-function(lambda,alpha,a,b,x){
lambda*alpha*(a/b)*((1+(x/b))^(-(a+1)))*
((1+(x/b))^(a*(alpha+1)))*exp(lambda*(1-(1+(x/b))^(a*alpha)))
}
```

```
hist(x, prob=TRUE, br=9, main='Fitted Densities',
,xlab="Life time",ylab="Density",ylim=c(0,0.015))
```

```
curve(TIIHLTLEGE_pdf(x, alpha=2.5654e+00,
beta= 9.8618e-02,lambda= 1.3277e-03,
b=2.5564e+01),lty=6, col=2, add=TRUE, lwd=2)
#par(new=TRUE)
curve(HLBXII_pdf(x, alpha=1.5476e+00,
lambda=1.3104e-05,a= 2.5273e-02,
b=1.5738e+01 ), lty=2, col=7, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TIIEHLTLWL_pdf(x, a=1.0019e+00
,b= 9.9876e-01 , theta=1.3301e-09,
lambda=4.9717e-02), lty=2, col=4, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TLWLx_pdf(x, a=0.14561, b=0.74365,
alpha=3.01451, theta=0.65476 )
,lty=2, col=3, add=TRUE,lwd=3)
#par(new=TRUE)
#curve(EHLPGWL_pdf (x,alpha= 2.512513
,beta=0.442807,delta=2.658286,
```

```

c= 0.284138),lty=2, col=6, add=TRUE,lwd=3)
#par(new=TRUE)
curve(TIIGIE_Lx_pdf(x, lambda=0.040843
,alpha=1.891761 ,a=0.428481,
b=1.000007 ),lty=2, col=1, add=TRUE,lwd=3)
#par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,c("TIIHLTLEGE",
"HLBXII","TIIHLLTLWL","TLW_Lx","TIIGIE_Lx"),
col=c(2,7,4,3,1),lwd=c(2,2,2),lty=c(6,2,2,2,2))

##### SURVIVAL TIMES OF GUINEA PIGS DATA FITTED DENSITIES
par(mfrow=c(0,1))
x<-c(12, 15, 22, 24, 24, 32, 32, 33, 34, 38,
38, 43, 44, 48, 52, 53, 54, 54, 55, 56, 57,
58, 58, 59, 60,
60, 60, 60, 61, 62, 63, 65, 65, 67, 68, 70,
70, 72, 73, 75, 76, 76, 81, 83, 84, 85, 87,
91, 95, 96, 98, 99, 109, 110, 121, 127, 129,
131, 143, 146, 146, 175, 175, 211, 233, 258,
258, 263, 297, 341, 341, 376
)

#####TIIHLTLEGE
TIIHLTLEGE_pdf<-function(alpha, beta,lambda,b,x){
(4*b*lambda*alpha*beta*(exp(-(lambda*x)))*
((exp(-(lambda*x)))^(alpha-1))*
((1-exp(-(lambda*x*alpha)))^(beta-1))*
(1-(1-exp(-(lambda*x*alpha)))^(beta))*
((1-(1-(1-exp(-(lambda*x*alpha)))^(beta)))
^(2))^(b-1)))/
((1+(1-(1-(1-exp(-(lambda*x*alpha)))^(beta))^(2))^(b))^(2))
}

#####HLBXII
HLBXII_pdf<-function(alpha,lambda,a,b,x){

```

```

(2*alpha*lambda*a*b*(x^(a-1))*(exp(lambda*
((1-(1+x^a)^b))))*(1-exp(lambda*(1-(1+x^a)^b)))^(alpha-1))/
(((1+x^a)^(-b+1))*(1+exp(lambda*(1-(1+x^a)^b)))*(alpha+1))
}

```

#####TIIEHLTLWL

```

TIIEHLTLWL_pdf<-function(a,b,theta,lambda,x){
((4*a*b*theta*lambda*(x^(lambda-1))*(exp(-x^lambda))*
((1-exp(-2*x^lambda))^(b-1))*(exp(-x^lambda)))/
((1+(1-exp(-2*x^lambda))^b)^(a+1)))*
((1-(1-exp(-2*x^lambda))^b)^(a-1))*
(((1-theta*((1-(1-exp(-2*x^lambda))^b)/
((1+(1-exp(-2*x^lambda))^b)))^a)^(-1))/(-log(1-theta)))
}

```

#####TLWLx

```

TLWLx_pdf<-function(a,b,alpha,theta,x){
2*theta*alpha*a*b*((1+b*x)^(a*alpha-1))*(((1-(1+b*x)^(-a))^(alpha-1)))*
(exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha)))*
(1-exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha))^(theta-1))
}

```

#####EHLPGWL

```

EHLPGWL_pdf<-function(alpha,beta,delta,c,x){
(2*delta*alpha*beta*c*(x^(c-1))*((1+x^c)^(-2))*
((1-(1+x^c)^(-1))^(alpha-1))*
((1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^(beta-1))*
(exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^beta))*
((1-exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^beta))^(delta-1)))/
(((1+x^c)^(-alpha-1))*((1+exp(1-(1+((1-(1+x^c)^(-1))/
(1+x^c)^(-1))^alpha)^beta))^(delta+1)))
}

```

####TIIGIE-Lx

```

TIIGIE_Lx_pdf<-function(lambda,alpha,a,b,x){
lambda*alpha*(a/b)*((1+(x/b))^(-(a+1)))*

```

```

((1+(x/b))^(a*(alpha+1)))*
exp(lambda*(1-(1+(x/b))^(a*alpha)))
}

hist(x, prob=TRUE, br=9, main='Fitted Densities',
xlab="Life time", ylab= "Density",ylim=c(0,0.015))

curve(TIIHLTLEGE_pdf(x, alpha=2.4828e+01,
beta= 2.2748e-02,lambda=1.9383e-04,
b= 1.5640e+03),lty=6, col=2, add=TRUE, lwd=3)
#par(new=TRUE)
curve(HLBXII_pdf(x, alpha=8.0381e-01,
lambda=1.9748e-05,a= 1.8769e-01,
b=8.8644e+00 ), lty=2, col=7, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TIIEHLTLWL_pdf(x, a= 0.0657806
,b= 0.1422881 , theta=0.0046556,
lambda=0.4321893), lty=2, col=4, add=TRUE, lwd=3)
# par(new=TRUE)
curve(TLWLx_pdf(x, a=0.135823,
b=3.070082, alpha=2.413841,
theta=8.141779 ),lty=2, col=3, add=TRUE,lwd=2)
#par(new=TRUE)
#curve(EHLPGWL_pdf (x,alpha= 0.54993
,beta=0.59521,delta=16.98723,
c= 1.09348),lty=2, col=6, add=TRUE,lwd=2)
#par(new=TRUE)
curve(TIIGIE_Lx_pdf(x, lambda=1.3567e-03,
alpha=9.8949e+00 ,a=1.4154e-01,
b=1.0001e+00 ),lty=2, col=1, add=TRUE,lwd=3)
#par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
par(new=TRUE)
legend("topright",pch=1,cex=0.8,
c("TIIHLTLEGE",
HLBXII","TIIEHLTLWL","TLW_Lx","TIIGIE_Lx"),col=c(2,7,4,3,1),
lwd=c(2,2,2),lty=c(6,2,2,2,2))

```

SURVIVAL TIMES OF GUINEA PIGS PROBABILITY PLOTS

```
data=c(
12, 15, 22, 24, 24, 32, 32, 33, 34, 38, 38, 43, 44,
48, 52, 53, 54, 54, 55, 56, 57, 58, 58, 59, 60,
60, 60, 60, 61, 62, 63, 65, 65, 67, 68, 70, 70, 72,
73, 75, 76, 76, 81, 83, 84, 85, 87,
91, 95, 96, 98, 99, 109, 110, 121, 127, 129, 131, 143,
146, 146, 175, 175, 211, 233, 258, 258, 263, 297, 341, 341, 376
)
```

```
data<-sort(data)
n=length(data)
```

#####TIIHLTLEGE

```
alpha1=2.4828e+01
beta1= 2.2748e-02
lambda1=1.9383e-04
b1= 1.5640e+03
```

#####HLBXII

```
alpha2=8.0381e-01
lambda2=1.9748e-05
a2= 1.8769e-01
b2=8.8644e+00
```

#####TIIEHLTLWL

```
a3= 0.0657806
b3= 0.1422881
theta3=0.0046556
lambda3=0.4321893
```

#####TLWLx

```
a4=0.135823
b4=3.070082
alpha4=2.413841
theta4=8.141779
```

#####EHLPGWL

```
alpha5= 0.54993
beta5=0.59521
delta5=16.98723
c5= 1.09348
```

#####TIIGIE

```
lambda6=1.3567e-03
```

```

alpha6=9.8949e+00
a6=1.4154e-01
b6=1.0001e+00

F_observed<-c(rep(0,n))

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

require(gsl)
require(stats)

data=c(
12, 15, 22, 24, 24, 32, 32, 33, 34, 38, 38,
  43, 44, 48, 52, 53, 54, 54, 55, 56, 57, 58, 58, 59, 60,
60, 60, 60, 61, 62, 63, 65, 65, 67, 68, 70,
  70, 72, 73, 75, 76, 76, 81, 83, 84, 85, 87,
91, 95, 96, 98, 99, 109, 110, 121, 127, 129,
  131, 143, 146, 146, 175, 175, 211, 233, 258,
  258, 263, 297, 341, 341, 376
)

```

```
(data,alpha,lambda,a,b,n){
```

```
(data,alpha,lambda,a,b,n){
```

```
(data,alpha,lambda,a,b,n){
```

```
(data,alpha,lambda,a,b,n){
```

```
}
```

```
  return(F)
```

```
}
```

```
TLWLx_cdf<-function(data,a,b,alpha,theta,n){
```

```
  F<-c(rep(0,n))
```

```
  for (i in 1:n)
```

```
  {
```

```
    x<-data[i]
```

```
    F[i]<- (((1-exp(-2*(((1-(1+b*x)^(-a))/  
      (1+b*x)^(-a))^alpha))))^theta)
```

```
  }
```

```
  return(F)
```

```
}
```

```
EHLPGWL_cdf<-function(data,alpha,beta,delta,c,n){
```

```
  F<-c(rep(0,n))
```

```
  for (i in 1:n)
```

```
  {
```

```
    x<-data[i]
```

```
    F[i]<- (((1-exp(1-(1+((1-(1+x^c)^(-1))/  
      (1+x^c)^(-1))^alpha)^beta))/  
      (1+exp(1-(1+((1-(1+x^c)^(-1))/  
      (1+x^c)^(-1))^alpha)^beta)))^delta)
```

```
  }
```

```
  return(F)
```

```
}
```

```
TIIGIE_Lx_cdf<-function(data,lambda,alpha,a,b,x){
```

```
  F<-c(rep(0,n))
```

```
  for (i in 1:n)
```

```
  {
```

```
    x<-data[i]
```

```
    F[i]<-(1-exp(lambda*(1-(1+(x/b))^(a*alpha))))
```

```
}
```

```

    return(F)
}

F1<-TIIHLEGE_cdf(data,alpha1,beta1,lambda1,b1,n)
F1
F2<-HLBXII_cdf(data,alpha2,lambda2,a2,b2,n)
F2
F3<-TIIHLEGE_cdf(data,a3,b3,theta3,lambda3,n)
F3
F4<-TLWLx_cdf(data,a4,b4,alpha4,theta4,n)
F4
F5<-EHLPGWL_cdf(data,alpha5,beta5,delta5,c5,n)
F5
F6<-TIIHLEGE_Lx_cdf(data,lambda6,alpha6,a6,b6,x)
F6

SS<-function(F,F_line,n){
  SS<-0
  for (i in 1:n)
  {
    SS<-SS+(F[i]-F_line[i])^2
  }
  return(SS)
}

SS1<-SS(F1,F_observed,n)
SS1
SS2<-SS(F2,F_observed,n)
SS2
SS3<-SS(F3,F_observed,n)
SS3
SS4<-SS(F4,F_observed,n)
SS4
SS5<-SS(F5,F_observed,n)
SS5
SS6<-SS(F6,F_observed,n)
SS6

```

```

op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l'
,main="Probability Plots"
,xlab="Observed probability",ylab="Expected probability")
lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
lines(F_observed,F6,col='maroon',type='l')
legend("bottomright",cex=0.7,c("TIIHLTLEGE(SS=0.1321)",
"HLBXII(SS=0.88254)","TIIHLLTLWL(SS=4.4962)",
"TLW_Lx(SS=1328)","EHLPGW_LLoG(SS=0.1238)",
"TIIGIE_Lx(SS=0.4295)"),lty=c(1,1,1,1,1,1),
col=c("red","blue","purple","orange","green",
"maroon"),lwd=c(1,1,1,1,1,1))

#####AC SYSTEM DATA  PROBABILITY PLOTS

data=c(
1, 3, 5, 7, 11, 11, 11, 12, 14, 14, 14,
16, 16, 20, 21, 23, 42, 47 ,52, 62 ,71,
71 ,87, 90 ,95 ,120 ,120 ,225, 246, 261
)

data<-sort(data)
n=length(data)

#####TIIHLTLEGE
alpha1=2.5654e+00
beta1=9.8618e-02
lambda1=1.3277e-03
b1= 2.5564e+01
#####HLBXII
alpha2=1.5476e+00
lambda2=1.3104e-05
a2= 2.5273e-02
b2=1.5738e+01
#####TIIHLLTLWL
a3= 1.0019e+00

```

```

b3= 9.9876e-01
theta3=1.3301e-09
lambda3=4.9717e-02
#####TLWLx
a4=0.14561
b4=0.74365
alpha4=3.01451
theta4=0.65476
#####EHLPGWL
alpha5=2.512513
beta5=0.442807
delta5=2.658286
c5= 0.284138
#####TIIGIE
lambda6=0.040843
alpha6=1.891761
a6=0.428481
b6=1.000007

F_observed<-c(rep(0,n))

m=n+0.25

for (i in 1:n)

{

  F_observed[i]<-(i-0.375)/m
}

require(gsl)
require(stats)

F_observed<-c(rep(0,n))
m=n+0.25
for (i in 1:n)
{
  F_observed[i]<-(i-0.375)/m
}

require(gsl)

```

```

require(stats)

require(gsl)
require(stats)

data=c(
1, 3, 5, 7, 11, 11, 11, 12, 14, 14, 14, 16, 16,
20, 21, 23, 42, 47 ,52, 62 ,71, 71 ,87, 90 ,95
,120 ,120 ,225, 246, 261)

data<-sort(data)
n=length(data)

TIIHLTLEGE_cdf<-function(data,alpha,beta,lambda,b,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-(2*(1-(1-(1-exp(-(alpha*lambda*x)))^
(beta))^(2))^(b))/(1+(1-(1-(1-exp(-(lambda*alpha*x)))^(beta))^(2))^(b))
  }

  return(F)
}

HLBXII_cdf<-function(data,alpha,lambda,a,b,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-((((1-exp((lambda*(1-(1+x^a)^b))))/
(1+exp((lambda*(1-(1+x^a)^b))))))^alpha
  }
  return(F)
}

```

```

TIIHLLTLWL_cdf<-function(data,a,b,theta,lambda,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<-(1-((-log(1-theta*((1-(1-exp(-2*x^lambda))^b)/
      ((1+(1-exp(-2*x^lambda))^b)))^a))/(-log(1-theta))))
  }

  return(F)
}

```

```

TLWLx_cdf<-function(data,a,b,alpha,theta,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<- ((1-exp(-2*(((1-(1+b*x)^(-a))/(1+b*x)^(-a))^alpha))))^theta)
  }
  return(F)
}

```

```

EHLPGWL_cdf<-function(data,alpha,beta,delta,c,n){
  F<-c(rep(0,n))
  for (i in 1:n)
  {
    x<-data[i]
    F[i]<- (((1-exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta))/
      (1+exp(1-(1+((1-(1+x^c)^(-1))/(1+x^c)^(-1))^alpha)^beta)))^delta)
  }
  return(F)
}

```

```

TIIGIE_Lx_cdf<-function(data,lambda,alpha,a,b,x){
  F<-c(rep(0,n))

```

```

for (i in 1:n)
{
  x<-data[i]
  F[i]<-(1-exp(lambda*(1-(1+(x/b))^(a*alpha))))
}

return(F)
}

```

```

F1<-TIIHLEGE_cdf(data,alpha1,beta1,lambda1,b1,n)
F1
F2<-HLBXII_cdf(data,alpha2,lambda2,a2,b2,n)
F2
F3<-TIIHLETLWL_cdf(data,a3,b3,theta3,lambda3,n)
F3
F4<-TLWLx_cdf(data,a4,b4,alpha4,theta4,n)
F4
F5<-EHLPGWL_cdf(data,alpha5,beta5,delta5,c5,n)
F5
F6<-TIIGIE_Lx_cdf(data,lambda6,alpha6,a6,b6,x)
F6

```

```

SS<-function(F,F_line,n){
  SS<-0
  for (i in 1:n)
  {
    SS<-SS+(F[i]-F_line[i])^2
  }
  return(SS)
}

```

```

SS1<-SS(F1,F_observed,n)
SS1
SS2<-SS(F2,F_observed,n)
SS2
SS3<-SS(F3,F_observed,n)
SS3
SS4<-SS(F4,F_observed,n)

```

SS4

```
SS5<-SS(F5,F_observed,n)
```

SS5

```
SS6<-SS(F6,F_observed,n)
```

SS6

```
op <- par(mfrow=c(1,1))
plot(F_observed,F_observed,type='l',
main="Probability Plots",
xlab="Observed probability",ylab="Expected probability")
lines(F_observed,F1,col='red',type='l')
lines(F_observed,F2,col='blue',type='l')
lines(F_observed,F3,col='purple',type='l')
lines(F_observed,F4,col='orange',type='l')
lines(F_observed,F5,col='green',type='l')
lines(F_observed,F6,col='maroon',type='l')
legend("bottomright",cex=0.7,c("TIIHLTLEGE(SS=0.0755)",
"HLBXII(SS=0.7365)", "TIIHLTLWL(SS=8.4136)"
,"TLW_Lx(SS=0.0764)", "EHLPGW_LLoG(SS=0.0774)",
"TIIGIE_Lx(SS= 0.0835)"),
lty=c(1,1,1,1,1,1),
col=c("red", "blue", "purple", "orange",
"green", "maroon"),lwd=c(1,1,1,1,1,1))
```