INPUT MODELING

by

Başak Pınar

B.S., Chemical Engineering, Yıldız Technical University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2012

ACKNOWLEDGEMENTS

First and foremost I offer my sincerest gratitude to my thesis supervisor, Assoc. Prof. Wolfgang Hörmann who has supported me throughout my thesis with his valuable support, patience and knowledge.

I would like to thank Assist. Prof. Aybek Korugan and Assoc. Prof. Aslı Sencer for their interests and joining my thesis committee.

In addition, I would also like to thank to my friend; Sıla Halulu for her support, motivation and sincerity throughout the days spent in Boğaziçi University.

Finally, I appreciate my family for supporting me since my born. I dedicate this thesis to the most influential person in my life; to my mother.

I thankfully acknowledge the support of TUBITAK - Turkish Technological and Scientific Research Institute during my master.

ABSTRACT

INPUT MODELING

In this study input modeling is executed by using the most efficient statistical principles and a new input modeling tool, the Fit All function, is developed in the statistical computing environment R. Moreover a practical comparison between this function and commercial input modeling softwares such as the Arena Input Analyzer and EasyFit is made. The Fit All function automatically tries all distributions for the input data and recommends the most proper model to the user. Twenty-five continuous distributions are included to this function and the maximum likelihood method (MLE) is utilized to estimate the parameters for all these distributions. For model selection, Akaike's Information Criterion (AIC) is used which is a theoretically sound and popular criterion for model selection. The main advantage of utilizing AIC is that it is based on calculating the log-likelihood of the candidate distribution, therefore it is consistent with the MLE. To assess the goodness of fit for the best fitting distribution, the Chi-Square test is executed. If a proper fit can be found, this model is recommended to the user. Otherwise the empirical model is recommended. To compare the models recommended by the Fit All function, the Arena Input Analyzer and EasyFit; random samples from common distributions are generated. The results of the simulation study show that the Fit All function identifies the correct distribution clearly more successfully than the other softwares. This means that AIC performs much better than the other model selection procedures. Moreover the parameter estimates that are presented by the Fit All function are closer to the original parameters. This also shows that the MLE obtains high quality parameter estimates.

ÖZET

GİRDİ MODELLEME

Bu çalışmada en etkili istatistiksel ilkeler kullanılarak girdi modelleme uygulanmaktadır ve istatistiksel hesaplama programı R'da yeni bir girdi modelleme aracı olan Fit All fonksiyonu geliştirilmektedir. Ayrıca, bu fonksiyon ile Arena Girdi Modelleme ve EasyFit gibi ticari programların pratik bir karşılaştırması yapılmaktadır. Fit All fonksiyonu girdi verileri için tüm dağılımları otomatik olarak dener ve en uygun modeli kullanıcıya önerir. Bu fonksiyona yirmi beş sürekli dağılım dahil edilmiştir ve bu dağılımların parametrelerini tahmin etmek için maksimum olabilirlik metodu (MLE) uygulanmıştır. Model seçimi için; teorik olarak güvenilir ve popüler bir kriter olan Akaike'nin bilgi kriteri (AIC) kullanılmıştır. AIC'nin temel avantajı; aday fonksiyonun olabilirlik fonksiyonunun logaritmasına dayalı hesaplama yapması ve bu nedenle MLE ile uyumlu olmasıdır. En iyi dağılımın uygunluğunu değerlendirmek için Ki-Kare testi uygulanmıştır. Yeterli uygunluk bulunursa, bu model kullanıcıya önerilmektedir. Aksi takdirde deneysel model önerilmektedir. Fit All fonksiyonu, Arena Girdi Modelleme ve EasyFit'in önerdiği modellerin karşılaştırmasını yapmak için genel dağılımlardan rastgele örnekler üretilmiştir. Simülasyon çalışmasının sonuçları; doğru dağılımı belirlemede Fit All fonksiyonunun diğer programlardan açıkça daha başarılı olduğunu göstermiştir. Bu sonuç AIC'nin diğer model seçim yöntemlerinden üstün olduğu anlamına gelmektedir. Ayrıca Fit All fonksiyonunun sunduğu parametre tahminleri gerçek parametre değerlerine daha yakındır. Bu sonuç da MLE'nin yüksek kaliteli parametre tahminleri yapabildiğini göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

B	Beta function
I	Indicator function
K	Kernel function
L	Likelihood function
W	Probability density function of random noise distribution
$x_{(1)}$	Smallest order statistic
$x_{(n)}$	Largest order statistic
γ	Location parameter
Γ	Gamma function
Θ	Parameter set
Λ	Log-likelihood function
χ^2	Chi-Square test statistic
Ψ	Derivative of the log gamma function

LIST OF ACRONYMS/ABBREVIATIONS

AIC	Akaike's Information Criterion
CDF	Cumulative Density Function
CS	Chi-Square Test
IID	Independent and Identically Distributed
KDE	Kernel Density Estimate
KS	Kolmogorov-Smirnov Test
MLE	Maximum Likelihood Estimate
P-P	Probability-Probability Plot
PDF	Probability Density Function
Q-Q	Quantile-Quantile Plot

1. INTRODUCTION

Input modeling is the practice of selecting probability distributions to represent the random input of a simulation. If the random inputs are represented by a proper model, one can remove some uncertainties from the system hence draw a realistic picture of the behaviour of the real system. To make well-grounded decisions on the current and future performance of the process, input modeling is necessary.

There are many software products available which automatically fit distribution functions to data and recommend the best fitting model to the user. Two popular examples are the Arena Input Analyzer and EasyFit. The user is capable of finding the most proper model via using these softwares even though he does not have statistical background knowledge. In studying the details of the statistical methods used for the modeling process in these two softwares, it turns out that no generally accepted statistical criterion to select the best fitting distribution is used; it even turns out that the utilized selection criterion for both softwares does not take the number of parameters of the fitted distribution into account.

The aim of this thesis is therefore to design a similar software tool that is using well accepted statistical principles to select the best fitting distribution. It should allow the user to determine automatically and accurately which probability distribution best represents a data set. To execute parameter estimation, the MLE is utilized, to rank distributions from best to worst, AIC is performed and to asses if the best fitting distribution has a good fit, the Chi-Square test is applied. We decided to code our Fit All function in the statistical computing environment R.

The second main aim of this thesis is to make a practical comparison between our Fit All function and commercial input modeling softwares such as the Arena Input Analyzer and EasyFit. We hope to be able to see that the use of well accepted statistical principles improves the input modeling.

2. INPUT MODELING

Almost all real systems contain random inputs. In order to carry out a successful stochastic simulation and obtain true outputs, we need a tool to model the random inputs. This method is called input modeling which is an vital step in the design of stochastic simulation [1].

The times between arrivals in a call center, the processing time for a manufacturing process and the demand per unit time of a product are examples of random inputs which essentially represent uncertainties in the process. By implementing input models, we can obtain proper models for the random inputs. It is obvious that without considering the uncertainty, one can not make well-grounded decisions on the current and future performance of the process. Hence, input modeling is necessary.

A brief definition of input modeling is: The practice of selecting and fitting a probability distribution to data in order to represent randomness and uncertainty that is inherent in the system [2]. In this thesis, we deal with independent and identically distributed data coming from continuous distributions as they constitute the most important models in practice.

In this section, we start with a literature study. Secondly, we mention softwares available on input modeling. At last, we shortly explain our ideas on input modeling.

2.1. Input Modeling in the Literature

The aim of input modeling is to present the underlying process by including the key characteristics of random inputs as there is no clear-cut input model for any stochastic input. Therefore we should not expect a perfect match of the fitted distribution and the real data. Moreover some events, which may happen in reality, may not be reflected during input modeling [1].

Most of the softwares that carry out input modeling, fit standard distributions (e.g: normal, gamma, beta, etc.) to independent and identically distributed data. As it was stated before, we deal with independent and identically distributed data coming from continuous distributions. If the process exhibits dependence or non-stationary behaviour, special models (e.g: multivariate input models, non-homogeneous Poisson systems) should be implemented which is out of the scope of this thesis. Biller and Gunes [3] identified situations in which the standard input models fail to adequately represent the available input data and make recommendations for each case.

The mainframe of input modeling consists of four stages (see Banks [2] and Chung [4]): As a first stage, if physical characteristics of the process data are known, candidate probability distributions are selected. However in many cases, we might not have this information. For this reason, most of the software packages fit all the distributions that are included in the distribution pool to the data.

Parameter estimation is the second prominent stage of input modeling. The criteria for determining which parameter estimation method should be implemented is generally based on obtaining estimates with good statistical properties. The most common methods for parameter estimation are the least square error method, the method of moments, the method of percentiles and the maximum likelihood estimation method. All simulation input modeling packages estimate the parameters of the fitted input model using either one or a couple of these methods.

Third stage is assessing the goodness of fit of the distributions. The Chi-Square, the Kolmogorov-Smirnov and the Anderson-Darling tests which are built on hypothesis testing, can be applied in order to check how well a fit captures the key characteristics of the input data. Additionally, graphical analysis such as the density-histogram plots, the probability-probability (P-P) and the quantile-quantile (Q-Q) plots can be performed complementary to these goodness of fit tests. Based upon the results of the goodness of fit, the best fitting distribution is selected.

Finally, in the fourth stage, if no proper distribution is found, an empirical input model can be recommended. This means that instead of trying to solve the problem in a parametric approach by estimating the parameters of a standard distribution, a non-parametric approach is applied to estimate the unknown distribution.

Briefly, the execution of input modeling in the literature can be summarized as below:

- Select candidate or all available probability distributions.
- For these distributions, determine the values of the input model parameters.
- Via statistical tests and graphical analysis, check the goodness of fit. Then select the best fitting distribution as an input model.
- If none of the standard input models has a good fit to the given data, recommend an empirical input model.

After presenting the input modeling procedure, one question may arise: What about always using a non-parametric model? To answer this question, we will compare parametric and non-parametric models briefly.

Non-parametric models are based on collecting data and reusing them in order to drive simulation. In fact, it is hard to contain all of the system characteristics in that particular data set especially when the sample size is small. On the other hand, parametric models try to fill in gaps and smooth the data, thus inference can be executed regardless of the perfect reflection of the system characteristics by the data. Moreover by insuring the tail behaviour, they may predict extreme conditions which reduces risk against events that occur rarely. Another advantage of using parametric models is their simplicity. If one wants to see the reaction of the system to the changes in one input variable, it is enough to change the value of that parameter without the necessity of resampling. In addition, if the sample exhibits dependence or non-stationary behaviour, the data set can not be naively resampled. These situations can be reflected by a parametric model rather than reusing data itself. So performing the parametric

approach has important advantages. Non-parametric models are recommended if no proper parametric model is found [1].

2.1.1. Random Variate Generation from Kernel Density Estimate

When we complete the parameter estimation and check the goodness of fit, it may happen that no distribution fits well. In this case, we apply a non-parametric approach.

The most common non-parametric approach is to use an empirical distribution by simply resampling the observed data itself. If data are drawn from a continuous process, there may be gaps among observed data points. In order to overcome this problem, Law and Kelton [5] suggest using a linear interpolation of the empirical CDF as the CDF for generating random variates for continuous random variables: $(X_1, X_2, \ldots X_n)$. The empirical CDF, F is defined by first sorting the X_i's in increasing order. Let $X_{(i)}$ denote the ith smallest of the X_i's, so that $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$. Then F is given by

$$F(x) = \begin{cases} 0 & \text{if } x < X_{(1)} \\ \frac{i-1}{n-1} + \frac{x - X_i}{(n-1)(X_{(i+1)} - X_{(i)})} & \text{if } X_{(i)} \leq x < X_{(i+1)} \text{ for } i = 1, 2, \ldots, n-1 \\ 1 & \text{if } X_{(n)} \leq x. \end{cases} \quad (2.1)$$

However this process can be shown to have a poor approximation due to its fine structure. As sample size $n \to \infty$, the PDF of the empirical distribution does not converge to the true distribution (see Devroye [6] for further details).

An almost unknown simple alternative to constructing an empirical cumulative distribution is that of kernel density estimation. Hörmann and Bayar [7] pointed out that the choice of the input distribution is often hidden behind a procedure that is called generating random variates from empirical distributions. They showed that resampling with noise is well suited for modeling input distributions. Before explaining

their algorithm, we will briefly define kernel density estimation.

The kernel density estimate $\hat{f}_Y(x)$ of the unknown distribution is a sum of bumps placed at the observations given by the formula

$$\hat{f}_Y(x) = \frac{1}{nb} \sum_{i=1}^{n} K\left(\frac{x - x_i}{b}\right) ,$$

where $K(.)$ is a kernel function and $b > 0$ is a smoothing parameter called bandwidth. While the kernel function determines the shape of the bumps, the bandwidth; b determines their width.

The choice of the kernel function is an important issue. If a smooth kernel function is used, then a smooth density estimate will be obtained. A range of kernel functions are commonly used: Uniform, triangular, biweight, triweight, Epanechnikov, normal and others. Due to its convenient mathematical properties, we prefer utilizing the normal kernel: $K(x) = \Phi(x)$ where Φ is the standard normal density function.

Bandwidth selection is the other important issue in kernel density estimation. If a small bandwidth is selected, the variance will be too high. Conversely, if a large bandwidth is selected, then the bias will be too high. Thus, there is always a trade-off between the bias of the density estimator and its variance. By minimizing the mean integrated squared error, the optimal choice for b is obtained. If we use a Gaussian kernel, the optimal value of b is

$$b = 1.06 \ \min(s, R/1.34) \ n^{-1/5} , \tag{2.2}$$

where s is the sample standard deviation, R is the interquartile range and n is the sample size [8].

Hörmann and Bayar [7] suggested random variate generation without the necessity of computing the estimated density. They argued that to simulate observations

from the kernel density estimate, it is enough to resample the data and to add to each of the resampled observations some noise, i.e a continuous random variable with 0 expectation and small variance. This process is called smoothed bootstrap in the statistical literature. The density of the random noise distribution presents the kernel function K(.) that is denoted by W. Their general algorithm for kernel density estimation (KDE) is:

- Choose the smoothing parameter, b.
- Generate a random integer I uniformly distributed on (1,2,. . . ,n).
- Generate a random variate W from the noise distribution.
- Return $Y = X_I + bW$.

R code for generating random variates from kernel density of the data is given in Appendix A. We use the standard normal distribution $N(0, 1)$ as our random noise distribution. To calculate the value of the bandwidth b, we use Equation 2.2.

Algorithm KDE samples from the kernel density estimate and it guarantees that the density function of the empirical distribution approximates the density of the unknown true distribution as good as possible with respect to the mean integrated square error of the density estimate.

2.2. Softwares for Input Modeling

There are various simulation softwares that implement input modeling. EasyFit, the Arena Input Analyzer, BestFit, Vose, ExpertFit and Statistica are a few examples. Once the data are imported, they automatically fit distributions to the data and rank the distributions with respect to the selected goodness of fit measure. In this thesis, we want to compare our Fit All function which is developed in the statistical software R, with the Arena Input Analyzer and EasyFit. Therefore, we briefly introduce the functionalities and capabilities of the Arena Input Analyzer and EasyFit.

2.2.1. ARENA Input Analyzer

The Input Analyzer is part of the ARENA simulation software package available from Rockwell software. The software is available through www.arenasimulation.com. Arena input analyzer has the capability to:

- Examine a total of 15 distributions for data fitting
- Perform the Chi-Square and the Kolmogorov-Smirnov tests
- Generate graph of the best fitting distribution superimposed on a histogram of the empirical data

For parameter estimation the Arena Input Analyzer implements the method of moments, the maximum likelihood estimation method and the least square error method for different distributions in its distribution pool. It ranks all the distributions with respect to a measure of the goodness of fit which is called square error. The square error (e^2) is the sum of squared discrepancies between empirical histogram frequencies and best fitting distribution frequencies. It is clear that smaller value of e^2 represents better fit. Its formula is given by

$$e^2 = \sum_{j=1}^{J} [\hat{p}_j - p_j]^2 \tag{2.3}$$

where J is the number of cells in the empirical histogram, $\hat{p}_j$ is the relative frequency of the j'th cell in the empirical histogram and p_j is the best fitting distribution's probability of the corresponding interval.

Fit All Summary option lists all fitted distributions and associated square errors, e^2 and selects the one which has the minimum square error [9]. It is important to note that this ranking depends on the chosen histogram intervals. Moreover, the square error measure does not take into account the number of parameters that the model includes. A screen capture from the Arena Input Analyzer is given in Figure 2.1.

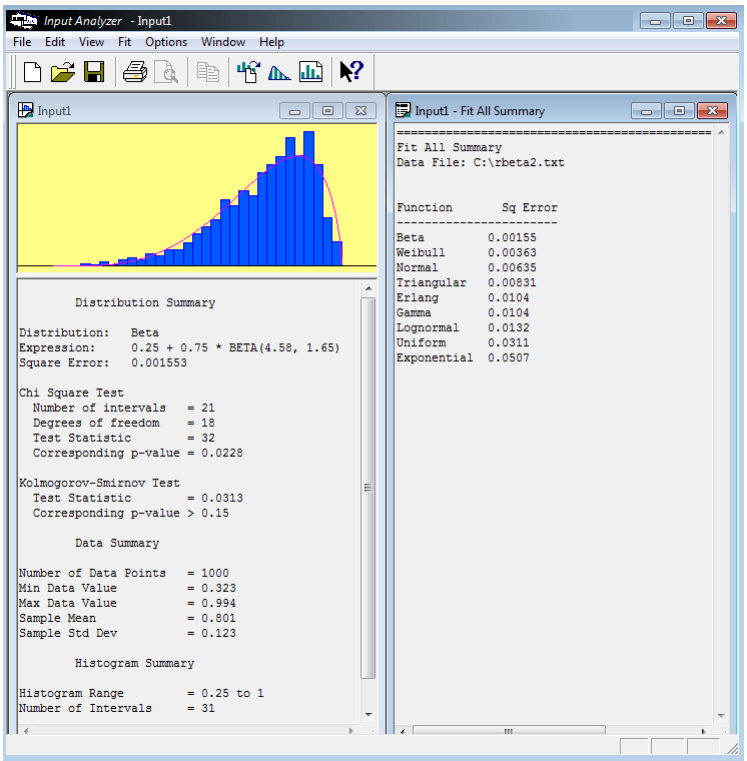

Figure 2.1. A screen capture from the Arena Input Analyzer.

2.2.2. EasyFit

EasyFit is available from Mathwave software through www.mathwave.com. This software has the capability to:

- Examine a total of 40 distributions for data fitting
- Perform the Chi-Square, the Kolmogorov-Smirnov and the Anderson-Darling tests
- Generate histograms of distributions, P-P, Q-Q and probability difference graphs

EasyFit performs different parameter estimation methods for different probability distributions in order to balance between the accuracy and the speed of calculations. The method of moments is used for the models that allow an analytical solution. However, for many other distributions, the method of moments does not yield closed-form expressions for parameter estimates and in such cases EasyFit uses the maximum

likelihood estimation method. The Least square method and the method of L-moments are the other parameter estimation tools that EasyFit utilizes [10]. A screen capture from EasyFit is presented in Figure 2.2.

Goodness of Fit - Summary

#	Distribution	Kolmogorov Smirnov		Anderson Darling		Chi-Squared	
		Statistic	Rank	Statistic	Rank	Statistic	Rank
5	Logistic	0,41786	1	146,1	1	494,54	4
7	Normal	0,42338	2	150,8	2	510,29	8
1	Beta	0,42355	3	150,8	3	510,29	9
6	Lognormal (3P)	0,42856	4	150,91	4	506,19	5
4	Gamma (3P)	0,42862	5	151,06	5	506,2	6
2	Erlang (3P)	0,43111	6	151,09	6	494,02	3
10	Uniform	0,43464	7	208,39	11	N/A	
9	Triangular	0,44336	8	154,12	8	519,22	10
11	Weibull (3P)	0,44902	9	153,39	7	506,39	7
8	Rayleigh (2P)	0,5648	10	181,43	9	334,06	2
3	Exponential (2P)	0,59303	11	195,72	10	285,81	1
12	Erlang	No fit (data min < 0)					
13	Exponential	No fit (data min < 0)					
14	Gamma	No fit (data min < 0)					
15	Lognormal	No fit (data min < 0)					
16	Rayleigh	No fit (data min < 0)					
17	Student's t	No fit					
18	Weibull	No fit (data min < 0)					

Fitting Results

#	Distribution	Parameters
1	Beta	α_1=44616,0 α_2=21971,0 a=-204,09 b=100,51
2	Erlang (3P)	m=345 β=0,03024 γ=-10,433
3	Exponential (2P)	λ=0,39935 γ=-2,5041
4	Gamma (3P)	α=316,12 β=0,03165 γ=-9,9999
5	Logistic	σ=0,30564 μ=-1,2268E-5
6	Lognormal (3P)	σ=0,02256 μ=3,2066 γ=-24,702
7	Normal	σ=0,55437 μ=-1,2268E-5
8	Rayleigh (2P)	σ=1,8292 γ=-2,5268
9	Triangular	m=-3,8049E-8 a=-2,5742 b=2,5754
10	Uniform	a=-0,96021 b=0,96018
11	Weibull (3P)	α=4,8101 β=3,2791 γ=-3,0676
12	Erlang	No fit (data min < 0)
13	Exponential	No fit (data min < 0)
14	Gamma	No fit (data min < 0)
15	Lognormal	No fit (data min < 0)
16	Rayleigh	No fit (data min < 0)
17	Student's t	No fit
18	Weibull	No fit (data min < 0)

Figure 2.2. A screen capture from EasyFit.

EasyFit can rank distributions with respect to three different measures of goodness of fit: The test statistics of the Chi-Square, the Kolmogorov-Smirnov and the Anderson-Darling tests. The user has the flexibility to order the distributions from best to worst with respect to each of these three measures.

2.3. Our Ideas on Input Modeling

As it was mentioned before, parameter estimation is a prominent part of input modeling. Like most softwares, the Arena Input Analyzer and EasyFit perform different parameter estimation methods for different types of distributions in order to gain acceleration and easiness of estimation. As we think that obtaining accurate estimates is more important, we concentrate on choosing a strong method that produces good quality estimates. Therefore we prefer the maximum likelihood method as it has many

attractive features (see Section 3.2).

Assessing the goodness of fit is the second issue that should be reviewed. Most softwares consider the Chi-Square test statistic as the primary measure. However it strongly depends on how the data are arranged into classes. In fact there is also no standard rule in terms of selecting the number of classes. The square error criterion has the same problem. As different number of classes can lead to different best fitting distribution, it is likely to have a poor fit. For this reason the Arena Input Analyzer is not generally capable to find the best fitting distribution. The Kolmogorov-Smirnov and the Anderson-Darling tests that are used by EasyFit have problems related to the critical values (see Sections 5.2 and 5.3). It is also important to consider that all these three statistics were never suggested as a criterion for model selection.

Another problem related to these measures is that of over-fitting. Over-fitting implies preferring a complex model that consists of many parameters to a simple model that includes fewer parameters [11]. Suppose we generate a sample coming from the uniform distribution. When we fit all the distributions to that sample, it is likely that we obtain the best fit for the four-parameter beta distribution rather than for the two-parameter uniform distribution. It is obvious that a distribution with four parameters may match the data better because it has more flexibility in shape than a two-parameter distribution. In order to overcome this problem, a measure of goodness of fit that considers the number of parameters is necessary.

When we consider all these problems of assessing the goodness of fit, we decided to use Akaike's Information Criterion (AIC) which is a popular criterion for model selection and is defined by

$$AIC = -2\log(L(\hat{\theta})) + 2k \tag{2.4}$$

where $\hat{\theta}$ is the maximum likelihood estimate, $L(.)$ is the likelihood function and k is the number of parameters.

The best model is the one with the smallest AIC value. AIC is congruent with the maximum likelihood estimation method since the main ingredient is the value of the log-likelihood at the maximum likelihood estimate. As the value of the log-likelihood function increases, we obtain smaller AIC values and better fits. The term $2k$ penalizes distributions for the number of parameters [12].

It is important to note that AIC does not give any warning in case there is no good fit. This is not surprising since AIC does not provide a test model in the sense of hypothesis testing. In fact, AIC ranks all distributions from best to worst. In order to learn whether the best model which has the minimum AIC, represents the data well or not, a goodness of fit test can be applied.

Our Fit All function implements the MLE for all distributions and ranks all the distributions from best to worst with respect to their AIC values. The distribution which has the smallest AIC value is selected as the best fitting distribution. It is still possible that this best fitting distribution does not fit well. To check that the P-values of the Chi-Square test are computed for all distributions. If the P-value that is computed for the best fitting distribution, exceeds the level of significance; α, then it is confirmed that the data are presented quite well by this distribution. Conversely if this P-value is smaller than the level of significance, we conclude that the data are not presented well by this distribution. In this case the Fit All function recommends the non-parametric model: "resampling with noise" (see Chapter 6).

3. MAXIMUM LIKELIHOOD ESTIMATION

There are many methods to estimate the parameters of distributions. The most common ones are the least square error method, the method of moments, the method of percentiles and the maximum likelihood estimation method. However, the maximum likelihood estimates (MLEs) have many desirable features often not possessed by the other methods mentioned here [5]. Therefore, we consider and apply the maximum likelihood method in this thesis.

3.1. Implementation of Maximum Likelihood Estimation

The maximum likelihood method is the procedure of finding the value of one or more parameters for a given model such that the likelihood function is maximized. The likelihood function is defined by

$$L(\Theta|x) = L(\theta_1, \theta_2, \ldots, \theta_k|x_1, x_2, \ldots, x_n) = L = \prod_{i=1}^{n} f(x_i, \theta_1, \theta_2, \ldots, \theta_k) \ , \tag{3.1}$$

where x represents the observations and Θ denotes the parameter vector.

For a fixed set of data, we aim to find the values of the parameters that maximize the likelihood function

$$L(\hat{\theta}) = \max_{\theta \in \Theta} L(\theta) \ . \tag{3.2}$$

For many applications, it is more convenient to work in terms of the natural logarithm of the likelihood function, called the log-likelihood, than in terms of the likelihood function itself. As the logarithm of a function achieves its maximum value at the same points as the function itself, the log-likelihood can be used in place of the

likelihood in the maximum likelihood estimation method.

$$\Lambda = \log L(\theta) = \sum_{i=1}^{n} \log f(x_i, \theta_1, \theta_2, \ldots, \theta_k) \tag{3.3}$$

For some distributions, the MLEs can be accomplished simply by an analytical solution. However, for many distributions, a numerical solution is required to find the MLEs [13].

3.1.1. Analytical Solution

Analytical solution refers that the MLEs can be obtained as an explicit function of the observed data. Firstly the log-likelihood function is differentiated with respect to the parameter vector. Then the resulting gradient vector is set to zero and the system of Equations is solved to find the MLEs.

The MLEs of $\theta_1, \theta_2, \ldots, \theta_k$ are the simultaneous solutions of the k equations such that:

$$\Lambda^{'} = \frac{d\Lambda}{d\theta_j} = 0 \ \text{ for } \ j = 1, 2, \ldots, k$$

For uniqueness we have to check the concavity of the log-likelihood. This can be performed by calculating the second derivative of the log-likelihood function with respect to the each parameter and showing whether they are all negative at $\theta_i = \hat{\theta}_i$ [14]:

$$\frac{d^2\Lambda}{d\theta_j^2} < 0 \ \text{ for } \ j = 1, 2, \ldots, k.$$

If the log-likelihood is not concave we have to check other local maxima. If possible, a plot of the likelihood is very useful for the problems containing not more

than two parameters [12, 14].

3.1.2. Numerical Solution

In practice, the analytical solution is not usually possible for many models. Especially when the distribution involves many parameters and/or it is highly nonlinear, a numerical solution is required to find the MLEs [14].

All the numerical solutions are iterative, starting with a tentative θ_0 and updating it through various iterations until an optimal value $\hat{\theta}$, is found. Essentially, by giving the computer the log-likelihood function and a set of starting values, the MLEs can be obtained easily by a hill climbing algorithm.

The numerical methods are separated by their search mechanism (e.g: gradient-based and direct search methods). As a typical guide search, gradient methods (e.g: the Newton's method and quasi-Newton method: the BFGS) apply analytic derivatives of the objective function. However, direct search methods (e.g: the Nelder-Mead method) do not use derivative information. These methods tend to converge more slowly, but can be more tolerant to the presence of noise in the function and constraints [15].

Finding reasonable starting values θ_0 is an important part of the numerical optimization. If the surface of the objective function is quadratic, it has only one minimum, and thus, all starting points will lead the minimization routine to the same optimal solution $\hat{\theta}$. In practice, however there may be many local minima. The best way to avoid false convergence in a local minimum is to start the routine at various locations or to start as close as possible to the optimal solution. To achieve this, heuristic estimates can be developed, often based on the first few moments of the data [16].

In this thesis, to find starting values for the numerical optimization problems, we use the method of moments estimators if they are in closed-form. If not, we use simple formulas that are suggested to obtain good starting values for the specified distribution.

We use the `optim`() and `optimize`() functions that allow numerical optimization in the statistical software R.

For the multi-dimensional optimization problems, we use the Nelder-Mead method which is the default algorithm of the `optim`() function. Although, it can be considerably slower than derivative based methods, it is less sensitive to discontinuities or noise in the likelihood surface, since it does not try to use fine-scale derivative information to navigate across the likelihood surface [17].

It is important to note that the Nelder-Mead method is not reliable for one-dimensional optimization problems. For this reason, if we have a one-dimensional numerical optimization, we apply different methods. When we know the parameter interval where the MLE is being searched, we use the `optimize`() function that implements the Brent method. Otherwise, we use the quasi-Newton method; BFGS in the `optim`() function.

3.1.2.1. Profile Log-likelihood. When the log-likelihood function is differentiated, it is possible to obtain some of the estimates in explicit and some of them in implicit forms. For this case, the concentrated or profile log-likelihood function can be utilized which is the commonly used simplification in maximum likelihood estimation.

Firstly, let the parameter vector be partitioned into two subsets α and β. Then the log-likelihood function can be written with two arguments:

$$\Lambda = \log L(\theta) = \log L(\alpha, \beta) \ .$$

Now suppose that, for a given value of β, the MLEs for the subset α could be found as a function of β; that is: $\hat{\alpha} = \hat{\alpha}(\beta)$. Then the profile log-likelihood function

(Λ_P) is a function only of β:

$$\Lambda_P = \log L_P(\beta) = \log L(\hat{\alpha}(\beta), \beta) \ . \tag{3.4}$$

Clearly, maximizing Λ_P for β will maximize Λ with respect to both α and β. The main advantage is a reduction in the dimension of the optimization problem. The greatest simplifications occur when α does not functionally depend on β [18].

For the gamma, Weibull, three-parameter lognormal and two-parameter Rayleigh distributions, we utilize the profile log-likelihood functions, hence reduce the computational burden significantly (see Chapter 4).

3.2. Properties of the MLEs

We have already mentioned that the MLEs have many desirable properties, so in this section we would like to briefly address them. Under very general and not restrictive conditions the MLEs have properties such as:

- For most of the common distributions, the MLE is unique. That is; $L(\hat{\theta})$ is strictly greater than $L(\theta)$ for any other value of θ.
- The MLEs may not be unbiased in small samples. However, in general they are asymptotically unbiased: (as $n \to \infty$, $E[\hat{\theta}] = \theta$).
- The MLEs converge more closely to the true value as the sample size increases. For this reason, they are strongly consistent.
- The MLEs are asymptotically normally distributed. That is, the asymptotic distribution of the MLEs is standard multivariate normal regardless of the distribution assumed in the model itself. This feature allows application of the central limit theorem, hypothesis testing and confidence intervals very easily.
- The MLEs are the optimal estimates among all unbiased estimators and they are minimum variance unbiased estimates. It means that it is impossible to have an

unbiased estimate with a smaller error than the MLEs [5, 12].

However it is important to mention that for many distributions finding the MLEs can be problematic. Especially, when the standard form of the distributions are shifted by adding a location and/or scale parameters, the likelihood function may not have a global maximum [5].

For example the MLEs are not well defined for the three parameter-gamma, three parameter-Weibull and three parameter-lognormal distributions. This is due to the fact that the log-likelihood function can become infinite as the location parameter equals to the smallest order statistic, $\gamma = x_{(1)}$ which results in inadmissible values for the other parameters. Thus, the MLEs may not be well defined for some cases (e.g the three-parameter gamma and the three-parameter Weibull distributions with shape parameter $\alpha < 1$) or numerical difficulties may occur [19–21].

4. PARAMETER ESTIMATION FOR COMMON PROBABILTY DISTRIBUTIONS

In this part of this thesis, we examine the parameter estimation for common continuous distributions. In our Fit All function, there are twelve distributions which are in their standard form. Since these distributions are not always capable of representing the data, we introduce their flexible forms by adding location and/or scale parameters to them. The names of the distributions are respectively; the beta, uniform, triangular, gamma, Erlang, Weibull, exponential, normal, Student's t, lognormal, logistic and Rayleigh. For all these distributions, the maximum likelihood method is utilized to execute the parameter estimation.

4.1. Beta Distribution

The beta distribution is widely used to model probability distributions of variables or project parameters in many areas of operational research. It is specifically used in risk analysis for strategic planning, finance and marketing. An attractive feature of this distribution is its allowance of both negative and positive skewness. As it is extremely versatile, a variety of uncertainties can be modeled by using it [22]. The beta distribution is defined in the interval $(0, 1)$ and parameterized by two shape parameters in the standard form. If we add a lower and an upper bound parameters (γ_1 and γ_2) to the standard form, the flexible form of the beta distribution is obtained which is defined in the interval (γ_1, γ_2) and parameterized by two shape and two boundary parameters.

4.1.1. Two-Parameter Beta Distribution

The two-parameter PDF of the beta distribution with shape parameters α_1 and α_2 is given by

$$f(x|\alpha_1, \alpha_2) = \frac{x^{\alpha_1 - 1}(1-x)^{\alpha_2 - 1}}{B(\alpha_1, \alpha_2)} \quad 0 \leq x \leq 1 \text{ and } \alpha_1, \alpha_2 > 0 \tag{4.1}$$

where the beta function $B(\alpha_1, \alpha_2)$ is represented by the following formula:

$$B(\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1)\,\Gamma(\alpha_2)}{\Gamma(\alpha_1, \alpha_2)}. \tag{4.2}$$

The beta function, $B(\alpha_1, \alpha_2)$, can be considered as a normalization constant to ensure that the total probability integrates to unity [23].

The likelihood function for n IID observations $(x_1, ..., x_n)$ is:

$$L(\alpha_1, \alpha_2|x_i) = \prod_{i=1}^{n} \frac{x_i^{\alpha_1-1}(1-x_i)^{\alpha_2-1}}{B(\alpha_1, \alpha_2)} \text{ for } i = 1, \ldots, n\ .$$

The log-likelihood function Λ, is in the following form:

$$\begin{aligned} \log L(\alpha_1, \alpha_2|x_i) = \Lambda &= \sum_{i=1}^{n} \{(\alpha_1 - 1)\log x_i + (\alpha_2 - 1)\log(1 - x_i) \\ &\quad - \log(B(\alpha_1, \alpha_2))\} \\ \Lambda &= (\alpha_1 - 1)\sum_{i=1}^{n} \log x_i + (\alpha_2 - 1)\sum_{i=1}^{n} \log(1 - x_i) \\ &\quad -n\ \log(B(\alpha_1, \alpha_2))\ . \end{aligned} \tag{4.3}$$

The log-likelihood function can be rearranged by substituting Equation 4.2 into 4.3 as in the following form:

$$\begin{aligned} \Lambda &= (\alpha_1 - 1)\sum_{i=1}^{n} \log x_i + (\alpha_2 - 1)\sum_{i=1}^{n} \log(1 - x_i) \\ &\quad +n \log\Gamma(\alpha_1, \alpha_2) - n \log\Gamma(\alpha_1) - n \log\Gamma(\alpha_2). \end{aligned} \tag{4.4}$$

To solve the MLEs of α_1 and α_2, we take the partial derivatives of the log-likelihood and set them equal to zero.

$$\begin{aligned}\frac{d\Lambda}{\alpha_1} &= \sum_{i=1}^{n} \log x_i + n\left(\Psi(\alpha_1,\alpha_2) - \Psi(\alpha_1)\right) = 0 \\ \frac{d\Lambda}{\alpha_2} &= \sum_{i=1}^{n} \log(1-x_i) + n\left(\Psi(\alpha_2,\alpha_2) - \Psi(\alpha_2)\right) = 0 \end{aligned} \tag{4.5}$$

$\Psi(.)$ is the digamma function representing the derivative of the log gamma function.

The system of Equations 4.5 shows that the MLEs of α_1 and α_2 can not be obtained in closed-form, hence have to be computed numerically.

In order to initiate the parameter estimation, we need starting values. Since the method of moments estimators of the beta distribution are in closed-form and consistent, we can obtain reasonable starting values by using them [24].

The method of moments estimators are given by

$$\begin{aligned}{\alpha_1}^0 &= \bar{x}^2\left(\frac{1-\bar{x}}{\bar{s}^2} - \frac{1}{\bar{x}}\right) \\ {\alpha_2}^0 &= {\alpha_1}^0\left(\frac{1}{\bar{x}} - 1\right) \end{aligned} \tag{4.6}$$

where $\bar{x}$ is the sample mean and $\bar{s}^2$ is the sample variance [25].

4.1.2. Four-Parameter Beta Distribution

The four-parameter PDF of the beta distribution is given by

$$f(x|\alpha_1,\alpha_2,\gamma_1,\gamma_2) = \frac{(x-\gamma_1)^{\alpha_1-1}(\gamma_2-x)^{\alpha_2-1}}{(\gamma_2-\gamma_1)^{\alpha_1+\alpha_2-1}B(\alpha_1,\alpha_2)} \quad \gamma_1 \le x \le \gamma_2 \text{ and } \alpha_1,\alpha_2 > 0 \tag{4.7}$$

where α_1 and α_2 are the shape and γ_1 and γ_2 are the lower and upper bound parameters respectively [26].

The likelihood function for n IID observations $(x_1, ..., x_n)$ is:

$$L(\alpha_1, \alpha_2, \gamma_1, \gamma_2 | x_i) = \prod_{i=1}^{n} \frac{(x_i - \gamma_1)^{\alpha_1 - 1}(\gamma_2 - x_i)^{\alpha_2 - 1}}{(\gamma_2 - \gamma_1)^{\alpha_1 + \alpha_2 - 1} B(\alpha_1, \alpha_2)} \text{ for } i = 1, \ldots, n.$$

We obtain the log-likelihood in the following form:

$$\begin{aligned} \log L(\alpha_1, \alpha_2, \gamma_1, \gamma_2 | x_i) = \Lambda &= \sum_{i=1}^{n} \{(\alpha_1 - 1)\log(x_i - \gamma_1) \\ &\quad + (\alpha_2 - 1)\log(\gamma_2 - x_i) \\ &\quad - (\alpha_1 + \alpha_2 - 1)\log(\gamma_2 - \gamma_1) \\ &\quad - \log(B(\alpha_1, \alpha_2))\} \end{aligned}$$

$$\begin{aligned} \Lambda &= (\alpha_1 - 1)\sum_{i=1}^{n} \log(x_i - \gamma_1) + (\alpha_2 - 1)\sum_{i=1}^{n} \log(\gamma_2 - x_i) \\ &\quad - n\ (\alpha_1 + \alpha_2 - 1)\log(\gamma_2 - \gamma_1) - n\ \log(B(\alpha_1, \alpha_2). \end{aligned} \tag{4.8}$$

The log-likelihood function is rearranged by substituting Equation 4.2 into 4.8:

$$\begin{aligned} \Lambda &= (\alpha_1 - 1)\sum_{i=1}^{n} \log(x_i - \gamma_1) + (\alpha_2 - 1)\sum_{i=1}^{n} \log(\gamma_2 - x_i) \\ &\quad - n\ (\alpha_1 + \alpha_2 - 1)\log(\gamma_2 - \gamma_1) \\ &\quad + n \log\Gamma(\alpha_1, \alpha_2) - n \log\Gamma(\alpha_1) - n \log\Gamma(\alpha_2). \end{aligned} \tag{4.9}$$

To solve the MLEs of $\alpha_1, \alpha_2, \gamma_1$ and γ_2, we take the partial derivatives of the log-likelihood and set them equal to zero.

$$
\begin{aligned}
\frac{d\Lambda}{\alpha_1} &= \sum_{i=1}^{n} \log(x_i - \gamma_1) - n \sum_{i=1}^{n} \log(\gamma_2 - \gamma_1) + n\left(\Psi(\alpha_1, \alpha_2) - \Psi(\alpha_1)\right) = 0 \\
\frac{d\Lambda}{\alpha_2} &= \sum_{i=1}^{n} \log(\gamma_2 - x_i) - n \sum_{i=1}^{n} \log(\gamma_2 - \gamma_1) + n\left(\Psi(\alpha_2, \alpha_2) - \Psi(\alpha_2)\right) = 0 \\
\frac{d\Lambda}{\gamma_1} &= -\sum_{i=1}^{n} \frac{(\alpha_1 - 1)}{(x_i - \gamma_1)} + \frac{n(\alpha_1 + \alpha_2 - 1)}{\gamma_2 - \gamma_1} = 0 \\
\frac{d\Lambda}{\gamma_2} &= \sum_{i=1}^{n} \frac{(\alpha_2 - 1)}{(\gamma_2 - x_i)} - \frac{n(\alpha_1 + \alpha_2 - 1)}{\gamma_2 - \gamma_1} = 0
\end{aligned} \tag{4.10}
$$

The system of Equations 4.10 shows that the MLEs can not be obtained in closed-form, hence have to be computed numerically [27].

We need starting values to initiate the parameter estimation. To find initial estimates of boundary parameters, the distribution support ($\gamma_1 \leq x \leq \gamma_2$) can be considered.

The initial estimate for the lower bound parameter ${\gamma_1}^0$, is taken slightly less than the smallest order statistic.

$$
{\gamma_1}^0 = \begin{cases} x_{(1)}\left(\frac{n+1}{n}\right) & \text{if } x_{(1)} < 0, \\ \\ x_{(1)}\left(\frac{n}{n+1}\right) & \text{if } x_{(1)} > 0. \end{cases} \tag{4.11}
$$

The initial estimate for the upper bound parameter ${\gamma_2}^0$ is taken slightly greater than the largest order statistic.

$$
{\gamma_2}^0 = \begin{cases} x_{(n)}\left(\frac{n}{n+1}\right) & \text{if } x_{(n)} < 0, \\ \\ x_{(n)}\left(\frac{n+1}{n}\right) & \text{if } x_{(n)} > 0. \end{cases} \tag{4.12}
$$

Starting values of the shape parameters (α_1 and α_2), can be obtained by using the method of moments estimators as for the two-parameter beta distribution.

Note that for the four-parameter beta distribution, mean and variance values are changed when considering the boundary parameters:

$$\begin{aligned} \bar{X} &= \frac{\bar{x} - \gamma_1}{(\gamma_2 - \gamma_1)} \\ \bar{S}^2 &= \frac{\bar{s}^2}{(\gamma_2 - \gamma_1)^2} \end{aligned} \tag{4.13}$$

where $\bar{x}$ is the sample mean and $\bar{s}^2$ is the sample variance [28].

The method of moments estimators 4.6 that enable us finding the starting values of the shape parameters for the two-parameter beta distribution are rearranged for the four-parameter beta distribution. By substituting $\bar{X}$ and $\bar{S}^2$ that are given in the system of Equations 4.13 instead of $\bar{x}$ and $\bar{s}^2$ values in 4.6, the initial estimates of the shape parameters are found as in the following form [24].

$$\begin{aligned} {\alpha_1}^0 &= \bar{X}^2 \left(\frac{1 - \bar{X}}{\bar{S}^2} - \frac{1}{\bar{X}} \right) \\ {\alpha_2}^0 &= {\alpha_1}^0 \left(\frac{1}{\bar{X}} - 1 \right) \end{aligned} \tag{4.14}$$

R code for finding the MLEs of the two forms of beta distribution is given in Appendix A. Firstly, the log-likelihood functions of the four and two-parameter beta distributions that are presented by Equations 4.9 and 4.4 respectively, are coded.

Then the optimization function, `MLEbeta(x, npar)` is created to execute the parameter estimation of the two forms of the beta distribution. The starting values are presented inside that function. For the four-parameter beta distribution, Equations 4.11 to 4.14 are utilized to find the initial solution. For the two-parameter beta distribution, the system of Equations 4.6 is utilized to find the starting values.

The optimization function, minimizes the negative of the log-likelihood functions: `lbeta4(x)` and `lbeta2(x)` by initiating the optimization with the starting values which are stored in the `starting.beta` vector. To carry out the optimization, the `optim()` function with its default method Nelder-Mead is called.

It is necessary to consider the quality of the MLEs of the beta distribution: The MLEs of the two-parameter beta distribution are not necessarily unbiased, and the degree of bias will be greater for small samples. For several parameter values, large sample sizes are necessary for the MLEs to become accurate [29].

The MLEs of the four-parameter beta distribution are not easily estimated, especially when the boundary parameters are unknown. Cheng and Iles [30] pointed out that the log-likelihood function has a local maximum when (α_1 and $\alpha_2 \geq 1$) and the MLEs are consistent. However for (α_1 and $\alpha_2 < 1$) the local maximum does dot exist and the MLE fails. Moreover, only for large samples ($n \geq 1000$) does the bias in the MLEs become small [27]. Besides all these limitations the MLE is, in general, considered as better than other parameter estimation methods [26].

In order to see the practical relevance of all these results, we did some simulation experiments. Firstly we can easily see that, as the number of observations increases, the bias becomes smaller. Moreover, we tried to find the MLEs of the four-parameter beta distribution when (α_1 and $\alpha_2 < 1$). The results showed that we may have convergence problems that indicate the iteration limits of the Nelder-Mead method have been reached.

4.2. Uniform Distribution

The uniform distribution is also called the rectangular distribution because of the shape of its PDF. It is widely used as the basis for the generation of random numbers for other statistical distributions [31].

The uniform distribution is defined in the interval $[0, b]$ and parameterized by the upper bound parameter in the standard form. If we add the lower bound parameter a, to the standard form, the flexible form of the uniform distribution is obtained which is defined in the interval $[a, b]$ and parameterized by two boundary parameters.

4.2.1. One-Parameter Uniform Distribution

The one-parameter PDF of the uniform distribution is given by

$$f(x|b) = \frac{1}{b}\, I_{0\leq x\leq b} \quad \text{for } 0 \leq x \leq b \text{ and } b \in R. \tag{4.15}$$

The indicator function $I_{0\leq x\leq b}$, refers that the PDF will be equal to 0 if x_i is outside of the allowed interval $(0, b)$ [32].

For n IID observations; let the order statistics be $x_{(1)} < x_{(2)} < \ldots < x_{(n)}$. The likelihood function $L(b|x)$ is in the following form:

$$\begin{aligned} L(b|x_i) &= \prod_{i=1}^{n} \frac{1}{b}\, I_{0\leq x_i\leq b} \text{ for } i = 1, \ldots, n \\ L(b|x_i) &= b^{-n}\, I_{0\leq x_i\leq b}\ . \end{aligned}$$

Therefore, the log-likelihood function is:

$$\log L(b|x_i) = \Lambda = -n \log b + \log(I_{0\leq x_i}) + \log(I_{b\geq x_i})\ . \tag{4.16}$$

To find the MLE for b, we take the derivative of the log-likelihood and set it equal to zero.

$$\frac{d(\Lambda)}{db} = -\frac{n}{b} < 0$$

It means that for $b > 0$ the log-likelihood is a decreasing function of b. Therefore the maximum of the log-likelihood occurs at the discontinuity point [33]. So, the MLE of the one-parameter uniform distribution is found to be

$$\hat{b} = x_{(n)} \ , \tag{4.17}$$

where $x_{(n)}$ represents the largest order statistic.

Since we have a closed-form solution that is presented in Equation 4.17, we can directly utilize it to find the MLE of the one-parameter uniform distribution.

4.2.2. Two-Parameter Uniform Distribution

The two-parameter PDF of the uniform distribution is given by

$$f(x|a,b) = \frac{1}{b-a} \ I_{a \leq x \leq b} \quad \text{for } a \leq x \leq b \text{ and } a, b \in R \tag{4.18}$$

where a is the lower bound and b is the upper bound parameters. Parameter a represents the location and $(b-a)$ denotes the scale of the two-parameter uniform distribution. The indicator function $I_{a \leq x \leq b}$, implies again that the PDF will be equal to 0 if x_i is outside of the allowed interval (a, b) [32].

For n IID observations; let the order statistics be $x_{(1)} < x_{(2)} < \ldots < x_{(n)}$. From Equation 4.18, it follows that for $a \leq x_{(1)}$ and $b \geq x_{(n)}$, the likelihood function $L(a, b|x)$, equals to

$$\begin{aligned} L(a,b|x_i) &= \prod_{i=1}^{n} \frac{1}{b-a} \ I_{a \leq x_i \leq b} \ \text{ for } i = 1, \ldots, n \\ L(a,b|x_i) &= (b-a)^{-n} \ I_{a \leq x_i \leq b} \ . \end{aligned}$$

So, the log-likelihood function is:

$$\log L(a, b|x_i) = \Lambda = -n \log(b - a) + \log(I_{a \leq x_i}) + \log(I_{b \geq x_i}) \ . \tag{4.19}$$

To find the MLE of a, we take the partial derivative of the log-likelihood and set it equal to zero.

$$\frac{d(\Lambda)}{da} = \frac{n}{b - a} > 0 \tag{4.20}$$

We can immediately conclude that for $a \leq x_{(1)}$ and $b \geq x_{(n)}$ the log-likelihood function, Λ is a strictly increasing function in the variable a. Hence, from Equation 4.20 it follows that the MLE of a is given by

$$\hat{a} = x_{(1)}. \tag{4.21}$$

To find the MLE for b, we take the partial derivative of the log-likelihood and set it equal to zero.

$$\frac{d(\Lambda)}{db} = -\frac{n}{b - a} \tag{4.22}$$

We can immediately conclude that for the allowed interval, the log-likelihood function Λ is a strictly decreasing function in the variable b. Hence, from Equation 4.22 it follows that the MLE of b is given by

$$\hat{b} = x_{(n)} \ . \tag{4.23}$$

Since we have closed-form solutions for both parameters that are presented in Equations 4.21 and 4.23, we can directly utilize them to find the MLEs of the two-parameter uniform distribution [34].

R code for finding the MLEs of the two forms of uniform distribution is given in Appendix A. Firstly, the log-likelihood functions of the two and one-parameter uniform distributions which are presented by Equations 4.16 and 4.19 respectively, are coded in the statistical software R.

Then the function, `MLEuniform(x, npar)` is created to implement the parameter estimation of the two forms of the uniform distribution. Since we have closed-form solutions for both distributions, we directly utilize them to obtain the MLEs. We use Equations 4.21 and 4.23 for the two-parameter uniform distribution and we use Equation 4.17 for the one-parameter uniform distribution.

It is important to consider the quality of the MLEs of the uniform distribution. For both the one and two-parameter uniform distributions, the distribution support depends on the parameters ($(0, b)$ and (a, b)). So, they do not satisfy the regularity conditions. Therefore, the attractive features of the maximum likelihood estimation method are not valid for the uniform distribution [34].

4.3. Triangular Distribution

The triangular distribution is specifically used as a subjective description of a population when there are limited sample data available. It is mainly based on a knowledge of the minimum, maximum and the modal value [2].

The PDF of the triangular distribution is given by

$$f(x|a,b,c) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{if } a \leq x < c, \\ \\ \frac{2}{b-a} & \text{if } x = c, \\ \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{if } c < x \leq b. \end{cases} \tag{4.24}$$

where c is the shape, a is the lower bound and b is the upper bound parameters respectively. Parameter c represents the mode, parameter a represents the location and $(b-a)$ denotes the scale of the triangular distribution.

The likelihood function for n IID observations $(x_1, ..., x_n)$ is:

$$L(a,b,c|x_i) = \begin{cases} \prod_{i=1}^{n} \frac{2(x_i-a)}{(b-a)(c-a)} & \text{if } a \leq x_i < c, \\ \\ \prod_{i=1}^{n} \frac{2}{b-a} & \text{if } x_i = c, \\ \\ \prod_{i=1}^{n} \frac{2(b-x_i)}{(b-a)(b-c)} & \text{if } c < x_i \leq b. \end{cases} \tag{4.25}$$

Therefore the system of the log-likelihood equations is in the following form:

$$\Lambda = \begin{cases} \sum_{i=1}^{n} \log(2(x_i-a)) - n(\log(b-a) - \log(c-a)) & \text{if } a \leq x_i < c, \\ \\ n(\log 2 - \log(b-a)) & \text{if } x_i = c, \\ \\ \sum_{i=1}^{n} \log(2(b-x_i)) - n(\log(b-a) - \log(b-c)) & \text{if } c < x_i \leq b. \end{cases} \tag{4.26}$$

Let the order statistics be $x_{(1)} < x_{(2)} < \ldots < x_{(n)}$. For all values c such that

$$x_{(1)} < c < x_{(n)}$$

the likelihood $L(a, b, c|x_i) \to 0$, hence the log-likelihood $\Lambda \to -\infty$ as the parameter a approaches the smallest order statistic $x_{(1)}$ or the parameter b approaches the largest order statistic $x_{(n)}$. So, if a modal value can be observed in the data (e.g: via utilizing a histogram) that indicates the validity of the expression $x_{(1)} < c < x_{(n)}$, then the MLEs for a and b are not the order statistics $x_{(1)}$ and $x_{(n)}$ respectively. This is in contrast with the well-known fact that the MLEs of the uniform distribution with support (a, b) are the smallest order statistic $x_{(1)}$ and the largest one $x_{(n)}$.

Briefly, the MLE of parameter a is less than the smallest order statistic and the MLE of parameter b is greater than the largest order statistic when $x_{(1)} < c < x_{(n)}$ [35].

As we do not have closed-form solutions for the MLEs, we need to implement a numerical solution. In order to begin optimization, we need starting values of the parameters. For the lower bound parameter a, it is reasonable to define an initial estimate that is less than the smallest order statistic and for the upper bound parameter b, an initial estimate that is greater than the largest order statistic. For the shape parameter c, we can simply compute the mode value of the data set.

R code for finding the MLEs of the triangular distribution is given in Appendix A. Firstly, the log-likelihood function, which is presented by Equation 4.26 is coded.

Then the optimization function, `MLEtriangular(x)` is created to execute the parameter estimation of the triangular distribution. The starting values are presented inside this function as `starting.triangular`. Equations 4.11 and 4.12 are utilized as initial estimates of parameters a and b. For parameter c, we compute the mode value by using the function: `Mode(x)`.

To carry out optimization, the `optim`() function with its default Nelder-Mead method is utilized in the statistical software R. By initiating optimization with starting values: `starting.triangular`, the optimization function minimizes the negative of the log-likelihood function: `ltriangular`(`x`).

Similar to the uniform distribution, the triangular distribution does not satisfy the regularity conditions since the distribution support depends on its parameters ($a \leq x \leq b$) [35].

4.4. Gamma Distribution

The gamma distribution is widely used in many research areas such as engineering, hydrology and survival analysis. Specifically it may offer a good fit to some sets of failure data due to its flexibility [36].

The gamma distribution is defined in the interval $[0, \infty)$ and parameterized by the shape and scale parameters in the standard form. If we add the location parameter γ to the standard form, the flexible form of the gamma distribution is obtained which is defined in the interval $[\gamma, \infty)$.

4.4.1. Two-Parameter Gamma Distribution

The two-parameter PDF of the gamma distribution is given by

$$f(x|\alpha, \beta) = \frac{\beta^{-\alpha}(x)^{\alpha-1}e^{-(x)/\beta}}{\Gamma(\alpha)} \quad \text{for } x \geq 0 \text{ and } \alpha, \beta > 0 \tag{4.27}$$

where α is the shape and β is the scale parameters respectively.

The likelihood and log-likelihood functions for n IID observations $(x_1, ..., x_n)$ are:

$$\begin{aligned} L(\alpha, \beta|x_i) &= \prod_{i=1}^{n} \frac{\beta^{-\alpha} x_i^{\alpha-1} e^{-x_i/\beta}}{\Gamma(\alpha)} \text{ for } i = 1, \ldots, n \\ \log L(\alpha, \beta|x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ -\alpha \log \beta + (\alpha - 1) \log x_i - \frac{x_i}{\beta} - \log \Gamma(\alpha) \right\} . \end{aligned}$$

Therefore, the log-likelihood function can be simplified to

$$\Lambda = -n\alpha \log \beta + (\alpha - 1) \sum_{i=1}^{n} \log x_i - \sum_{i=1}^{n} x_i/\beta - n \log \Gamma(\alpha) . \tag{4.28}$$

The system of the log-likelihood equations is given by

$$\begin{aligned} \frac{d(\Lambda)}{d\alpha} &= -n \ \log \beta + \sum_{i=1}^{n} \log x_i - n \ \Psi(\alpha) = 0 \\ \frac{d(\Lambda)}{d\beta} &= -\frac{n\alpha}{\beta} + \frac{\sum_{i=1}^{n} x_i}{\beta^2} = 0 \end{aligned}$$

where $\Psi(.)$ is the digamma function representing the derivative of the log gamma function.

We rearrange the log-likelihood equations to isolate the parameters and get [37]:

$$\begin{aligned} \Psi(\alpha) &= -\log \beta + \left(\sum_{i=1}^{n} \log x_i \right) /n \\ \hat{\beta} &= \sum_{i=1}^{n} x_i/n\alpha . \end{aligned} \tag{4.29}$$

We clearly see in the system of Equation 4.29 that only the scale parameter β can be calculated in closed-form. Thus, if a value for α is available, then $\hat{\beta}$ is easily computed. For this reason, we first estimate α and then use the equation:

$\hat{\beta} = \sum_{i=1}^{n} x_i / n\alpha$ to obtain $\hat{\beta}$. If we rearrange Equation 4.28 by substituting the closed-form $\hat{\beta}$ estimate, we get the profile log-likelihood function (Λ_P) in the following form [38]:

$$\begin{aligned} \log L(\alpha, \hat{\beta}|x_i) = \Lambda_P &= -n\alpha \log\left(\frac{\sum_{i=1}^{n} x_i}{n\alpha}\right) + (\alpha - 1)\sum_{i=1}^{n} \log x_i \\ &\quad -n\alpha - n \log\Gamma(\alpha) \ . \end{aligned} \tag{4.30}$$

We require a numerical solution in order to maximize the profile log-likelihood function. Therefore we need a starting value for α. We can utilize the method of moments estimators of the gamma distribution as they are in closed-form. For the shape parameter α, the method of moments estimator is given as

$$\begin{aligned} \alpha &= \frac{E[x]^2}{Var[x]} \\ \alpha^0 &= \left(\frac{\bar{x}}{s}\right)^2 \end{aligned} \tag{4.31}$$

where $\bar{x}$ is sample mean and s is sample standard deviation [39].

4.4.2. Three-Parameter Gamma Distribution

The three-parameter PDF of the gamma distribution is given by

$$f(x|\alpha, \beta, \gamma) = \frac{\beta^{-\alpha}(x-\gamma)^{\alpha-1}e^{-(x-\gamma)/\beta}}{\Gamma(a)} \quad \text{for } x \geq \gamma \text{ and } \alpha, \beta > 0 \tag{4.32}$$

where α is the shape, β is the scale and γ is the location parameters respectively.

The likelihood and log-likelihood functions for n IID observations $(x_1, ..., x_n)$ are:

$$\begin{aligned} L(\alpha, \beta, \gamma | x_i) &= \prod_{i=1}^{n} \frac{\beta^{-\alpha}(x_i - \gamma)^{\alpha-1} e^{-(x_i-\gamma)/\beta}}{\Gamma(a)} \text{ for } i = 1, \ldots, n \\ \log L(\alpha, \beta, \gamma | x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ -\alpha \log \beta + (\alpha - 1) \log(x_i - \gamma) - \frac{(x_i - \gamma)}{\beta} - \log \Gamma(\alpha) \right\} . \end{aligned}$$

The log-likelihood function can be simplified to

$$\Lambda = -n\alpha \log \beta + (\alpha - 1) \sum_{i=1}^{n} \log(x_i - \gamma) - \sum_{i=1}^{n} (x_i - \gamma)/\beta - n \log \Gamma(\alpha) . \tag{4.33}$$

The system of the log-likelihood equations is given by

$$\begin{aligned} \frac{d(\Lambda)}{d\alpha} &= -n \log \beta + \sum_{i=1}^{n} \log(x_i - \gamma) - n\Psi(\alpha) = 0 \\ \frac{d(\Lambda)}{d\beta} &= -\frac{n\alpha}{\beta} + \frac{\sum_{i=1}^{n} (x_i - \gamma)}{\beta^2} = 0 \\ \frac{d(\Lambda)}{d\gamma} &= -(\alpha - 1) \sum_{i=1}^{n} (x_i - \gamma)^{-1} + \frac{n}{\beta} = 0 , \end{aligned}$$

where $\Psi(.)$ is the digamma function representing the derivative of the log gamma function.

We rearrange the log-likelihood equations to isolate the parameters and get:

$$\begin{aligned} \Psi(\alpha) &= -\log \beta + \left(\sum_{i=1}^{n} \log(x_i - \gamma) \right) / n \\ \hat{\beta} &= \sum_{i=1}^{n} (x_i - \gamma)/n\alpha \\ \sum_{i=1}^{n} (x_i - \gamma)^{-1} &= \frac{n}{\beta(\alpha - 1)} . \end{aligned} \tag{4.34}$$

As it is indicated in the system of Equations 4.34, the scale parameter β is the only parameter that can be obtained in closed-form. If we first estimate α and γ, then β can be computed using the equation, $\hat{\beta} = \sum_{i=1}^{n}(x_i - \gamma)/n\alpha$.

If we rearrange Equation 4.33 by substituting the closed-form $\hat{\beta}$, we get the profile log-likelihood function (Λ_P) as in the following form [38]:

$$\begin{aligned} \log L(\alpha, \gamma, \hat{\beta}|x_i) = \Lambda_P &= -n\alpha \log\left(\frac{\sum_{i=1}^{n}(x_i - \gamma)}{n\alpha}\right) + (\alpha - 1)\sum_{i=1}^{n}\log(x_i - \gamma) \\ &\quad -n\alpha - n\log\Gamma(\alpha)\ . \end{aligned} \tag{4.35}$$

We require a numerical solution in order to maximize the profile log-likelihood function. Therefore, we need starting values for γ and α.

The initial estimate of the location parameter γ^0, is taken as slightly less than the smallest order statistic. This is reasonable since $(x \geq \gamma)$ is a condition in the support of the distribution [19]. So, we use Equation 4.11 to find the starting value for γ.

The starting value for the shape parameter α can be found by utilizing the method of moments estimator as in for the two-parameter gamma distribution [39].

Note that for a three-parameter gamma distribution, mean and standard deviation values are rearranged considering the location parameter γ:

$$\begin{aligned} \bar{X} &= \frac{\sum_{i=1}^{n}(x - \gamma^0)}{n} \\ S &= \frac{\sum_{i=1}^{n}((x - \gamma^0) - \bar{X})^2}{n-1}\ , \end{aligned} \tag{4.36}$$

where $\bar{X}$ is the sample mean and S is the sample standard deviation.

The method of moments estimator given by Equation 4.31 is rearranged for the

three-parameter gamma distribution. By substituting $\bar{X}$ and $\bar{S}$ that are given in the system of Equations 4.36 instead of $\bar{x}$ and $\bar{s}$ values in 4.31, the initial estimate of the shape parameter α, is found.

R code for finding the MLEs of the two forms of gamma distribution is given in Appendix A. Firstly, the profile log-likelihood functions of the three and two-parameter gamma distributions that are presented by Equations 4.35 and 4.30 respectively, are coded.

Then the optimization function, `MLEgamma(x, npar)` is created to execute the parameter estimation of the two forms of gamma distribution. The starting values are presented inside this function. For the three-parameter gamma distribution, Equations 4.11, 4.31 and 4.36 are utilized to find the starting values and for the two-parameter gamma distribution, Equation 4.31 is used in order to find the initial solution as well. The optimization function minimizes the negative of the profile log-likelihood functions: `lgamma3(x)` and `lgamma2(x)` by initiating optimization with starting values: `starting.gamma`.

To carry out the optimization, the `optim()` function is utilized for both distributions in the statistical software R. For the three-parameter gamma distribution the Nelder-Mead method is used and for the two-parameter gamma distribution the BFGS is utilized. This is due to the fact that after we introduce the profile log-likelihood functions, dimension of the optimization is reduced by one. That means; for the two-parameter gamma distribution, we implement one dimensional optimization and for the three-parameter gamma distribution, we have two dimensional optimization. Since the Nelder-Mead method is not reliable for one-dimensional optimization, we prefer using the BFGS to find the MLEs.

For the two-parameter gamma distribution, after $\hat{\alpha}$ was found which is denoted by `res$par`, $\hat{\beta}$ is calculated by using the formula: $\hat{\beta} = \sum_{i=1}^{n} x_i/n\alpha$. Similarly for the three-parameter gamma distribution, after $\hat{\alpha}$ and $\hat{\gamma}$ were found which are denoted

`res$par[1]` and `res$par[2]` respectively, $\hat{\beta}$ is calculated by using the formula: $\hat{\beta} = \sum_{i=1}^{n}(x_i - \gamma)/n\alpha$.

It is important to mention the quality of the MLEs of the gamma distribution. Various estimators have been used in practice for the two-parameter gamma distribution. The MLE is a preferable choice due to its optimal asymptotic properties. The MLEs, $\hat{\alpha}$ and $\hat{\beta}$ can exhibit considerable percentage bias when the sample size is small (especially for $n < 100$). As the sample size increases, the biases and mean squared errors of the MLEs fall which reflect the consistency of these estimators [37].

Balakrishnan and Wang [19] noted that when the shape parameter, α is small (especially for $\alpha < 2.5$), the MLEs have problems of convergence in the parameter estimation of the three-parameter gamma distribution. However, if the shape parameter α is large, then the MLEs are asymptotically efficient and also will face no problems in convergence.

We did some experiments in order to see how our R-code for the three-parameter gamma distribution reacts when the shape parameter $\alpha < 2.5$. We observed that when $\alpha < 1$ the number of iterations increases and the MLEs may have problems of convergence.

4.5. Erlang Distribution

The Erlang distribution was developed as the distribution of waiting time and message length in telephone traffic. If the durations of individual calls are exponentially distributed, the duration of a succession of calls has an Erlang distribution [31].

The Erlang distribution is the gamma distribution with shape parameter α, an integer. The notes on parameter estimation for the gamma distribution apply to the Erlang distribution. Thus, in order to avoid repetition, we do not perform the same operations again. Moreover we only examine the standard form of the Erlang distri-

bution.

The log-likelihood function of the Erlang distribution has the same structure with the gamma distribution. The only difference is that the shape parameter must have an integer value. So we apply `as.integer()` function to the shape parameter α inside the log-likelihood function of gamma distribution. R code for finding the MLEs of the Erlang distribution is given in Appendix A.

Briefly, we find the MLEs of two-parameter gamma distributions at first. Then by using these estimates as starting values, we find the MLEs of the Erlang distribution.

To carry out optimization, the `optim()` function is utilized with its BFGS method. After $\hat{\alpha}$ is found which is denoted by `res$par`, $\hat{\beta}$ is calculated by using the formula: $\hat{\beta} = \sum_{i=1}^{n} x_i/n\alpha$.

As the Erlang distribution is a special case of the gamma distribution, see Section 4.4.1 to learn the quality of the MLEs.

4.6. Weibull Distribution

The Weibull distribution is widely used in reliability and life data analysis due to its versatility. It can be used to model a variety of life time behaviors depending on the values of the parameters [40].

The Weibull distribution is defined in the interval $[0, \infty)$ and parameterized by the shape and scale parameters in the standard form. If we add the location parameter γ to the standard form, the flexible form of the Weibull distribution is obtained which is defined in the interval $[\gamma, \infty)$.

4.6.1. Two-Parameter Weibull Distribution

The two-parameter PDF of the Weibull distribution is given by

$$f(x|\alpha,\beta) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}} \text{ for } x \geq 0 \text{ and } \alpha,\beta > 0 \tag{4.37}$$

where α is the shape and β is the scale parameters respectively.

The likelihood and log-likelihood functions for n IID observations $(x_1, ..., x_n)$ are:

$$\begin{aligned} L(\alpha,\beta|x_i) &= \prod_{i=1}^{n} \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x_i/\beta)^{\alpha}} \text{ for } i = 1,\ldots,n. \\ \log L(\alpha,\beta|x_i) = \Lambda &= \sum_{i=1}^{n}\left\{\log\alpha - \alpha\log\beta + (\alpha-1)\log x_i - \left(\frac{x_i}{\beta}\right)^{\alpha}\right\}. \end{aligned}$$

Therefore, the log-likelihood function can be simplified to

$$\Lambda = n\log\alpha - n\alpha\log\beta + (\alpha-1)\sum_{i=1}^{n}\log x_i - \beta^{-\alpha}\sum_{i=1}^{n}(x_i)^{\alpha}. \tag{4.38}$$

To find the MLEs of α and β, we take the partial derivatives of the log-likelihood and set them equal to zero.

$$\begin{aligned} \frac{d(\Lambda)}{d\beta} &= -\frac{n\alpha}{\beta} + \frac{\alpha}{\beta^{\alpha+1}}\sum_{i=1}^{n}(x_i)^{\alpha} = 0 \\ \hat{\beta} &= \left(\frac{\sum_{i=1}^{n}(x_i)^{\alpha}}{n}\right)^{1/\alpha} \end{aligned} \tag{4.39}$$

$$\frac{d(\Lambda)}{d\alpha} = \frac{n}{\alpha} - n\log\beta + \sum_{i=1}^{n}\log x_i + \frac{\log\beta}{\beta^{\alpha}}\sum_{i=1}^{n}(x_i)^{\alpha} - \frac{1}{\beta^{\alpha}}\sum_{i=1}^{n}(x_i)^{\alpha}\log x_i = 0 \tag{4.40}$$

As it can be seen in Equations 4.39 and 4.40, only the scale parameter β is calculated in closed-form. If we first estimate α, then β can be obtained through Equation 4.39.

If we rearrange Equation 4.38 by substituting the closed-form of $\hat{\beta}$, we get the profile log-likelihood function (Λ_P) [41]:

$$\begin{aligned}\log L(\alpha, \hat{\beta}|x_i) = \Lambda_P &= n\log\alpha - n\alpha\log\left[\left(\frac{\sum_{i=1}^{n}(x_i)^\alpha}{n}\right)^{1/\alpha}\right] + (\alpha-1)\sum_{i=1}^{n}\log x_i \\ &\quad - \left[\left(\frac{\sum_{i=1}^{n}(x_i)^\alpha}{n}\right)^{1/\alpha}\right]^{-\alpha}\sum_{i=1}^{n}(x_i)^\alpha\,. \end{aligned} \tag{4.41}$$

We require a numerical solution in order to maximize the profile log-likelihood function. Therefore, we need a starting value for α. As initial estimate α^0, we use

$$\alpha^0 = \frac{\bar{x}}{s}\,, \tag{4.42}$$

where $\bar{x}$ is the sample mean and s is the sample standard deviation [2].

4.6.2. Three-Parameter Weibull Distribution

The three-parameter PDF of the Weibull distribution is given by

$$f(x|\alpha,\beta,\gamma) = \alpha\beta^{-\alpha}(x-\gamma)^{\alpha-1}e^{-((x-\gamma)/\beta)^\alpha} \text{ for } x \geq \gamma \text{ and } \alpha,\beta > 0 \tag{4.43}$$

where α is the shape and β is the scale and γ is the location parameters respectively.

The likelihood and log-likelihood functions for n IID observations $(x_1, ..., x_n)$ are:

$$\begin{aligned} L(\alpha, \beta, \gamma | x_i) &= \prod_{i=1}^{n} \alpha \beta^{-\alpha} (x_i - \gamma)^{\alpha - 1} e^{-((x_i - \gamma)/\beta)^{\alpha}} \text{ for } i = 1, \ldots, n. \\ \log L(\alpha, \beta, \gamma | x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ \log \alpha - \alpha \log \beta + (\alpha - 1) \log (x_i - \gamma) - \left(\frac{x_i - \gamma}{\beta} \right)^{\alpha} \right\}. \end{aligned}$$

Therefore, the log-likelihood function can be simplified to

$$\Lambda = n \log \alpha - n\alpha \log \beta + (\alpha - 1) \sum_{i=1}^{n} \log (x_i - \gamma) - \beta^{-\alpha} \sum_{i=1}^{n} (x_i - \gamma)^{\alpha}. \tag{4.44}$$

To find the MLEs of α, β and γ, we take the partial derivatives of the log-likelihood and set them equal to zero. $\hat{\beta}$ is the only estimate which can be isolated on the left-hand side of the equation.

$$\begin{aligned} \frac{d(\Lambda)}{d\beta} &= -\frac{n\alpha}{\beta} + \frac{\alpha}{\beta^{\alpha+1}} \sum_{i=1}^{n} (x_i - \gamma)^{\alpha} = 0 \\ \hat{\beta} &= \left(\frac{\sum_{i=1}^{n} (x_i - \gamma)^{\alpha}}{n} \right)^{1/\alpha} \end{aligned} \tag{4.45}$$

Hence, if α and γ are estimated, $\hat{\beta}$ can easily be computed by utilizing Equation 4.45. Therefore we firstly estimate α and γ. If we rearrange Equation 4.44 by substituting the closed-form $\hat{\beta}$ estimator, we get the profile log-likelihood function (Λ_P) [42]:

$$\begin{aligned} \Lambda_P &= n \log \alpha - n\alpha \log \left[\left(\frac{\sum_{i=1}^{n} (x_i - \gamma)^{\alpha}}{n} \right)^{1/\alpha} \right] + (\alpha - 1) \sum_{i=1}^{n} \log (x_i - \gamma) \\ &\quad - \left[\left(\frac{\sum_{i=1}^{n} (x_i - \gamma)^{\alpha}}{n} \right)^{1/\alpha} \right]^{-\alpha} \sum_{i=1}^{n} (x_i - \gamma)^{\alpha} \ . \end{aligned} \tag{4.46}$$

We require a numerical solution in order to maximize the profile log-likelihood

function. Therefore, we need reasonable starting values for γ and α.

When the location parameter γ is unknown and must therefore be estimated from the sample data, the starting value for γ is selected less than the smallest order statistic: $\gamma^0 < x_{(1)}$ [20]. Hence, we take the initial estimate (γ^0) equal to

$$\gamma^0 = \begin{cases} x_{(1)} + 0.1x_{(1)} & \text{if } x_{(1)} < 0, \\ x_{(1)} - 0.1x_{(1)} & \text{if } x_{(1)} > 0, \end{cases} \tag{4.47}$$

where $x_{(1)}$ is the smallest order statistic.

The starting value for the shape parameter α, can be found in the same way as for the two-parameter Weibull distribution. We can again use Equation 4.42 to find the starting value of α.

Note that for a three-parameter Weibull distribution, mean and standard deviation formulas have to be changed by including the location parameter γ:

$$\begin{aligned} \bar{X} &= \frac{\sum_{i=1}^{n}(x-\gamma^0)}{n} \\ S &= \frac{\sum_{i=1}^{n}((x-\gamma^0)-\bar{X})^2}{n-1}\,. \end{aligned} \tag{4.48}$$

Here, $\bar{X}$ is the sample mean and S is the sample standard deviation. We substitute $\bar{X}$ and S that are given in the system of Equations 4.48 instead of the $\bar{x}$ and s values in Equation 4.42 [2].

R code for finding the MLEs of the two forms of Weibull distribution is given in Appendix A. Firstly, the profile log-likelihood functions of the three and two-parameter Weibull distributions that are presented by Equations 4.46 and 4.41 respectively, are coded.

Then the optimization function, `MLEweibull(x, npar)`, is written to execute the parameter estimation of both forms of the Weibull distribution. The starting values are presented inside this function as `starting.weibull`. For the three-parameter Weibull distribution, Equations 4.42, 4.47 and 4.48 are utilized to find the initial solution and for the two-parameter Weibull distribution, Equation 4.42 is required as well. The optimization function, minimizes negative of the profile log-likelihood functions: `lweibull3(x)` and `lweibull2(x)` by initiating optimization with starting values: `starting.weibull`.

To carry out optimization, the `optim()` function is utilized for both distributions in the statistical software R. For the three-parameter Weibull distribution the Nelder-Mead method is used and for the two-parameter Weibull distribution the BFGS is utilized. This is due to the fact that after we introduce the profile log-likelihood functions, dimension of the optimization is reduced by one. That means; for the two-parameter Weibull distribution, we implement a one dimensional optimization and for the three-parameter Weibull distribution, we have a two dimensional optimization problem. Since the Nelder-Mead method is not reliable for one-dimensional optimization, we prefer using the BFGS in that case.

For the two-parameter Weibull distribution, after $\hat{\alpha}$ is found which is denoted by `res$par`, $\hat{\beta}$ is calculated by using Equation 4.39. Similarly for the three-parameter Weibull distribution, after $\hat{\alpha}$ and $\hat{\gamma}$ are found which are denoted by `res$par[1]` and `res$par[2]` respectively, $\hat{\beta}$ is calculated by using Equation 4.45.

It is important to consider properties of the MLEs of the Weibull distribution. Cohen and Whitten [20] addressed the issue of existence of the estimates for the three-parameter Weibull distribution;

- When $\alpha < 1$, the distribution is reverse J-shaped and the likelihood function becomes infinite as $\gamma \to x_{(1)}$. Accordingly in this situation estimate of γ is $x_{(1)}$, but estimates of α and β would not exist.

- When $\alpha = 1$, the two-parameter exponential distribution emerges as a special case. The estimators are both unbiased and have minimum variance.
- When $\alpha > 1$, the Weibull distribution is bell-shaped and the MLEs exist. Computational problems are likely to be encountered when α is near to 1, even though α actually exceeds 1. Furthermore, the usual good asymptotic properties of MLE do not hold unless $\alpha > 2$.

We made some experiments to see how our R-code for the three-parameter Weibull distribution behaves when the shape parameter $\alpha < 1$. We observed that the number of iterations increases and there are problems with the convergence. However, when the shape parameter $\alpha \geq 1$, there were no problems.

4.7. Exponential Distribution

The exponential distribution is a special case of both the gamma and the Weibull distributions for shape parameter $\alpha = 1$. It is widely used for survival analysis.

The exponential distribution is defined in the interval $[0, \infty)$ and parameterized by the rate parameter in the standard form. If we add the location parameter γ, to the standard form, the flexible form of the exponential distribution is obtained which is defined in the interval $[\gamma, \infty)$.

4.7.1. One-Parameter Exponential Distribution

The one-parameter PDF of the exponential distribution is given by

$$f(x|\beta) = \beta e^{-\beta x} \text{ for } x \geq 0 \text{ and } \beta > 0 \tag{4.49}$$

where β is the rate or the inverse scale parameter.

The likelihood and log-likelihood functions for n IID observations $(x_1, ..., x_n)$ are:

$$\begin{aligned} L(\beta|x_i) &= \prod_{i=1}^{n} \beta e^{-\beta x_i} \text{ for } i = 1, \ldots, n \\ \log L(\beta|x_i) = \Lambda &= \sum_{i=1}^{n} (\log \beta - \beta x_i) \ . \end{aligned}$$

Therefore the log-likelihood function is found to be

$$\Lambda = n \log \beta - \beta \sum_{i=1}^{n} x_i \ . \tag{4.50}$$

The MLE of β is found by taking derivative of Λ with respect to β:

$$\begin{aligned} \frac{d(\Lambda)}{d\beta} &= \frac{n}{\beta} - \sum_{i=1}^{n} x_i = 0 \\ \hat{\beta} &= \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\bar{x}} \ , \end{aligned} \tag{4.51}$$

where $\bar{x}$ is the sample mean.

To make sure that this is the maximum likelihood value, we check the second derivative of Equation 4.50:

$$\frac{d^2(\Lambda)}{d\beta} = -\frac{n}{\beta^2} < 0$$

Since the second derivative is negative, $1/\bar{x}$ is the maximum likelihood value of β [43]. As we have a closed-form solution that is presented in Equation 4.51, we can directly utilize it to find the MLE of the one-parameter exponential distribution.

4.7.2. Two-Parameter Exponential Distribution

The two-parameter PDF of the exponential distribution is given by

$$f(x|\beta,\gamma) = \beta e^{-\beta(x-\gamma)} \text{ if } x \geq \gamma,\ \beta > 0. \tag{4.52}$$

where β is the rate and γ is the location parameters respectively.

For n IID observations $(x_1, \ldots, x_n)$, let the order statistics be

$$x_{(1)} < x_{(2)} < x_{(3)} < \ldots < x_{(n)} \ .$$

From Equation 4.52 it follows that for $\gamma \leq x_{(1)}$, the likelihood function $L(\beta,\gamma|x)$ equals to

$$\begin{aligned} L(\beta,\gamma|x_i) &= \prod_{i=1}^{n} \beta e^{-\beta(x_i-\gamma)} = \beta^n e^{\sum -\beta(x_i-\gamma)} \\ L(\beta,\gamma|x_i) &= \beta^n exp\left\{-\beta\sum_{i=1}^{n} x_i + n\beta\gamma\right\} \ . \end{aligned}$$

The log-likelihood function $\log L(\beta,\gamma|x) = \Lambda$, equals to

$$\log L(\beta,\gamma|x_i) = \Lambda = n\log\beta - \beta\sum_{i=1}^{n} x_i + n\beta\gamma \ . \tag{4.53}$$

Taking the partial derivative of Λ with respect to γ:

$$\frac{d\Lambda}{\gamma} = n\beta > 0 \ . \tag{4.54}$$

We can immediately conclude that for $\gamma \leq x_{(1)}$, Λ is a strictly increasing function

in the variable γ regardless of the value of β. Hence, Equations 4.53 and 4.54 confirm that the MLE of γ is given by

$$\hat{\gamma} = x_{(1)}. \tag{4.55}$$

Taking the partial derivative of Λ with respect to β:

$$\begin{aligned} \frac{d(\Lambda)}{d\beta} &= \frac{n}{\beta} - \sum_{i=1}^{n} x_i - n\gamma = 0 \\ \hat{\beta} &= \frac{n}{\sum_{i=1}^{n}(x_i - \gamma)} > 0 \, . \end{aligned} \tag{4.56}$$

So, we have closed-form solutions of the MLEs for the two-parameter exponential distribution [44].

R code for finding the MLEs of the two forms of exponential distribution is given in Appendix A. Firstly, the log-likelihood functions of the two and one-parameter exponential distributions, which are presented by Equations 4.53 and 4.50 respectively, are coded.

Then the function, `MLEexponential(x, npar)` is implemented for the parameter estimation of both forms of the exponential distribution. Since we have closed-form solutions for both distributions, we directly utilize these formulas to obtain the MLEs. We use Equations 4.55 and 4.56 for the two-parameter exponential distribution and we use Equation 4.51 for the one-parameter exponential distribution.

For the quality of the MLEs see Section 4.6.2.

4.8. Normal Distribution

The normal distribution is applicable to a very wide range of phenomena and is the most widely used distribution in statistics [31].

The PDF of the normal distribution is given by

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)/2\sigma^2} \text{ for } x \text{ and } \mu \in R \text{ and } \sigma > 0 \tag{4.57}$$

where μ is the location and σ is the scale parameters respectively.

The likelihood function for n IID observations $(x_1, ..., x_n)$ is:

$$L(\mu,\sigma|x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\mu)/2\sigma^2}.$$

Therefore, the log-likelihood function will be in the following form:

$$\begin{aligned} \log L(\mu,\sigma|x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ -\frac{1}{2}\log(2\pi) - \log\sigma - \frac{(x_i-\mu)^2}{2\sigma^2} \right\} \\ \Lambda &= -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 \quad . \end{aligned} \tag{4.58}$$

Taking derivatives of Equation 4.58 with respect to μ and σ and solving the resulting system of first order conditions obtains the MLEs as below [12]:

$$\begin{aligned} \frac{d(\Lambda)}{d\mu} &= \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^2} = 0 \\ \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \hat{\mu} &= \bar{x} \end{aligned} \tag{4.59}$$

$$\begin{aligned}
\frac{d(\Lambda)}{d\sigma} &= -\frac{n}{\sigma} + \sigma^{-3}\sum_{i=1}^{n}(x_i-\mu)^2 = 0 \\
\hat{\sigma} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{\mu})^2} \\
\hat{\sigma} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}
\end{aligned} \tag{4.60}$$

As we have closed-form solutions that are presented in Equations 4.59 and 4.60, we can directly utilize them. R code for finding the MLEs of the normal distribution is given in Appendix A. Firstly, the log-likelihood function of the normal distribution, which is given in Equation 4.58 is coded.

Then the function, `MLEnormal(x)` is implemented for the parameter estimation of the normal distribution. As we have the analytical solutions, we directly use Equations 4.59 and 4.60 to find the MLEs of μ and σ respectively.

It is important to discuss the quality of the MLEs for the normal distribution. Firstly we examine the statistical bias of these estimates.

$$\begin{aligned}
E[\mu] &= \bar{x} = \hat{\mu} \\
E[\sigma^2] &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2 = \frac{n}{n-1}\hat{\sigma}^2
\end{aligned}$$

Hence, $\hat{\mu}$ is an unbiased and $\hat{\sigma}^2$ is a biased estimate. Although $\hat{\sigma}^2$ is biased for small samples, the difference between σ^2 and $\hat{\sigma}^2$ becomes negligibly small for large sample sizes. Moreover this estimator is consistent since as $n \to \infty$, $\hat{\sigma}^2 \to \sigma^2$ [45].

4.9. Student's t-Distribution

The Student's t-distribution is widely used when estimating the mean of a normally distributed population in situations where the sample size is small and the population standard deviation is unknown [46].

The Student's t-distribution is symmetric and bell-shaped thus resembling the normal distribution. However it is more likely producing values that fall far from its mean due to its heavier tails. This feature enables the t-distribution to describe extreme events [47].

It is parameterized by the shape parameter ν in the standard form. If we add the location parameter μ and the scale parameter σ to the standard form, the flexible form is obtained which includes three parameters.

4.9.1. One-Parameter Student's t-Distribution

The one-parameter PDF of the Student's t-distribution is given by

$$f(x|\nu) = \frac{1}{\sqrt{\nu}B\left(\frac{1}{2},\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad \text{for all } x \in R \text{ and } \nu > 0 \tag{4.61}$$

where parameter ν represents the shape parameter that is called degrees of freedom and B(.) denotes the beta function.

The shape of the t-distribution is affected by ν. As ν tends to infinity, the t distribution gets closer to the normal distribution [48].

Equation 4.61 can also be written as

$$f(x|\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} , \tag{4.62}$$

where $\Gamma(.)$ denotes the gamma function.

The likelihood function for n IID observations $(x_1, \ldots, x_n)$ is:

$$L(\nu|x_i) = \prod_{i=1}^{n} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x_i^2}{\nu}\right)^{-\frac{\nu+1}{2}} .$$

Therefore the log-likelihood function (Λ) will be in the following form:

$$\begin{aligned} \log L(\nu|x_i) &= \sum_{i=1}^{n} \left\{ \log\left(\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}\right) - \frac{\nu+1}{2}\log\left(1+\frac{x_i^2}{\nu}\right)\right\} \\ \Lambda &= \log\left(\frac{\Gamma^n\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}^n\Gamma^n\left(\frac{\nu}{2}\right)}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\left(1+\frac{x_i^2}{\nu}\right) . \end{aligned} \tag{4.63}$$

Taking the derivative with respect to ν and equating it to zero gives:

$$\begin{aligned} \frac{d\Lambda}{\nu} &= \frac{\frac{\frac{1}{2}\Gamma^n\left(\frac{\nu+1}{2}\right)n\Psi\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}^n\Gamma^n\left(\frac{\nu}{2}\right)} - \frac{\frac{1}{2}\Gamma^n\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}^n\Gamma^n\left(\frac{\nu}{2}\right)\frac{n}{\nu}} - \frac{\frac{1}{2}\Gamma^n\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}^n\Gamma^n\left(\frac{\nu}{2}\right)n\Psi\left(\frac{\nu}{2}\right)}}{\Gamma^n\left(\frac{\nu+1}{2}\right)\sqrt{\pi\nu}^n\Gamma^n\left(\frac{\nu}{2}\right)} \\ &\quad -\frac{1}{2}\left(\sum_{i=1}^{n}\log\left(1+\frac{x^2}{\nu}\right)\right) + \frac{\nu+1}{2}\sum_{i=1}^{n}\frac{x^2}{\nu+x^2} = 0 , \end{aligned} \tag{4.64}$$

where $\Psi(.)$ is digamma function representing the derivative of the log gamma function.

Equation 4.64 obviously shows that there is no analytical solution. So we have a one-dimensional non-linear optimization problem. In order to find the MLE, we can calculate the log-likelihood value (Λ) for various ν and find the maximum among those [47].

As an example, let's take a data set that includes 100 observations from the Student's t-distribution with $\nu = 6$ degrees of freedom. Figure 4.1 presents the log-likelihood values; $\Lambda = \log L(\nu|x_i)$ that are computed for $\nu = 1, 2, \ldots, 10$. The maxi-

mum is obtained for $\nu = 6$.

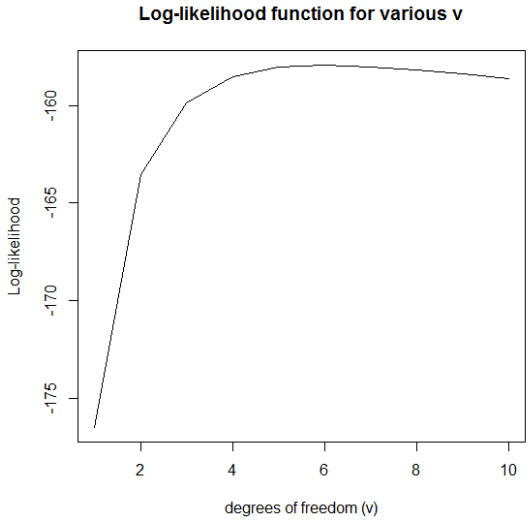

Figure 4.1. The log-likelihood function for Student's t-distribution ($\nu = 1, 2, \ldots, 10$).

If a data set comes from the Student's t-distribution, it is reasonable to expect finding values for ν between $(0, 120)$. This is due to the fact that after $\nu = 120$, the t-distribution is very similar to the normal distribution and even for $\nu \geq 30$ there is not much difference between those distributions [48]. So it is reasonable to search for the values of ν between $(0, 120)$. The one which gives the highest value of the log-likelihood function is found as the MLE of ν.

4.9.2. Three-Parameter Student's t-Distribution

The three-parameter PDF of the Student's t-distribution is given by

$$f(x|\mu, \sigma, \nu) = \frac{1}{\sigma\sqrt{\nu}B\left(\frac{1}{2}, \frac{\nu}{2}\right)}\left(1 + \frac{((x-\mu)/\sigma)^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad \text{for all } x \in R \tag{4.65}$$

where $\mu \in R$ is the location, $\sigma > 0$ is the scale, and $\nu > 0$ is the shape parameter. $B(.)$ denotes the beta function.

For the simplicity of the presentation, we can alternatively express Equation 4.65

by defining the corresponding standard variable as

$$a = \frac{x - \mu}{\sigma}.$$

Thus we get the PDF as

$$f(x|\mu, \sigma, \nu) = \frac{1}{\sigma\sqrt{\nu}B\left(\frac{1}{2}, \frac{\nu}{2}\right)} \left(1 + \frac{a^2}{\nu}\right)^{-\frac{\nu+1}{2}} . \tag{4.66}$$

Let x_i and a_i be the observed and the standard values in $i = 1, 2, \ldots, n$. Therefore the likelihood and the log-likelihood functions have the following form:

$$\begin{aligned} L(\mu, \sigma, \nu|x_i) &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{\nu}B\left(\frac{1}{2}, \frac{\nu}{2}\right)} \left(1 + \frac{a_i^2}{\nu}\right)^{-\frac{\nu+1}{2}} \\ \log L(\mu, \sigma, \nu|x_i) = \Lambda &= -n \log\left[\sigma\sqrt{\nu}B\left(\frac{1}{2}, \frac{\nu}{2}\right)\right] - \frac{\nu+1}{2} \sum_{i=1}^{n} \log\left(1 + \frac{{a_i}^2}{\nu}\right) \end{aligned} \tag{4.67}$$

The first derivatives of Λ with respect to the each parameter are:

$$\begin{aligned} \frac{d\Lambda}{\mu} &= \frac{\nu+1}{\sigma} \sum_{i=1}^{n} \left(\frac{{a_i}^2}{\nu + {a_i}^2}\right) \\ \frac{d\Lambda}{\sigma} &= -\frac{n}{\sigma} + \frac{\nu+1}{\sigma} \sum_{i=1}^{n} \left(\frac{{a_i}^2}{\nu + {a_i}^2}\right) \\ \frac{d\Lambda}{\nu} &= -\frac{n}{2\nu} - \frac{1}{2} \log\left(1 + \frac{{a_i}^2}{\nu}\right) - \frac{\nu+1}{2} . \end{aligned} \tag{4.68}$$

As there is no analytical expression, we need to implement a numerical solution [49, 50]. We need starting values for the optimization. For the location parameter μ, we can take the initial estimate as the sample mean since for the three-parameter Student's t-distribution $E[x] = \mu$. For the scale parameter σ and the shape parameter

ν, we may take the starting values slightly greater than 0 since both parameters are non-negative.

R code for finding the MLEs of the two forms of Student's t-distribution is given in Appendix A. Firstly, the log-likelihood functions of the three and one-parameter Student's t-distributions which are presented by Equations 4.67 and 4.64, are coded.

Then the optimization function, `MLEt(x, npar)` is implemented for the parameter estimation of the two forms of Student's t-distribution. For the three-parameter Student's t-distribution, the `optim()` function with its default Nelder-Mead method is utilized. For the one-parameter Student's t-distribution, the `optimize()` function is used due to its convenience for one-dimensional optimization.

4.10. Lognormal Distribution

The lognormal distribution is applicable to random variables that are constrained by zero but have a few very large values. It is widely used in reliability analysis, finance and wireless communication [31].

The lognormal distribution is defined in the interval $(0, \infty)$ and parameterized by two parameters: (μ, σ) in the standard form. If we add the location parameter γ to the standard form, the flexible form of the lognormal distribution is obtained which is defined in the interval (γ, ∞).

4.10.1. Two-Parameter Lognormal Distribution

A positive random variable x is said to be lognormally distributed with two parameters μ and σ, if $y = \log x$ is normally distributed with mean μ and variance σ^2. The two-parameter PDF of the lognormal distribution is given by

$$f(x|\mu, \sigma) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-(\ln x - \mu)/2\sigma^2} \text{ if } x > 0, -\infty < \mu < \infty \text{ and } \sigma > 0 \qquad (4.69)$$

where μ is the scale and σ is the shape parameters.

The likelihood and log-likelihood functions for n IID observations $(x_1, \ldots, x_n)$ are:

$$\begin{aligned} L(\mu, \sigma | x_i) &= \prod_{i=1}^{n} \frac{1}{x_i \sqrt{2\pi\sigma^2}} e^{-(\ln x_i - \mu)/2\sigma^2} \\ \log L(\mu, \sigma | x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ -\log x_i - \frac{1}{2}\log(2\pi) - \log \sigma - \frac{(\log x_i - \mu)^2}{2\sigma^2} \right\} . \end{aligned}$$

Thus, the log-likelihood function can be simplified to

$$\Lambda = -\sum_{i=1}^{n} \log x_i - \frac{n}{2}\log(2\pi) - n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (\log x_i - \mu)^2 . \tag{4.70}$$

If we take derivatives with respect to μ and σ and solve the resulting system of first order conditions, the MLEs are obtained as below [53]:

$$\begin{aligned} \frac{d(\Lambda)}{d\mu} &= \frac{\sum_{i=1}^{n} (\log x_i - \mu)}{\sigma^2} = 0 \\ \hat{\mu} = &= \frac{\sum_{i=1}^{n} \log x_i}{n} \end{aligned} \tag{4.71}$$

$$\begin{aligned} \frac{d(\Lambda)}{d\sigma} &= -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} (\log x_i - \mu)^2 = 0 \\ \hat{\sigma} &= \sqrt{\frac{\sum_{i=1}^{n} (\log x_i - \hat{\mu})^2}{n}} \end{aligned} \tag{4.72}$$

Since both parameters are in closed-form, we can directly use Equations 4.71 and 4.72 to find the MLEs.

4.10.2. Three-Parameter Lognormal Distribution

A random variable x, which can take any value exceeding a fixed value γ, is said to be lognormally distributed with three parameters μ, σ and γ if $y = \log(x-\gamma)$ is normally distributed with mean μ and variance σ^2 . For $x > \gamma$, $-\infty < \mu < \infty$ and $\sigma > 0$ the PDF of the three-parameter lognormal distribution is given by

$$f(x|\mu,\sigma,\gamma) = \frac{1}{(x-\gamma)\sqrt{2\pi\sigma^2}}\, e^{-(\ln(x-\gamma)-\mu)/2\sigma^2}. \tag{4.73}$$

The likelihood function for n IID observations $(x_1, \ldots, x_n)$ is:

$$L(\mu,\sigma,\gamma|x_i) = \prod_{i=1}^{n} \frac{1}{(x_i-\gamma)\sqrt{2\pi\sigma^2}} e^{-(\ln(x_i-\gamma)-\mu)/2\sigma^2}$$

For any ordered sample: $x_{(1)} < x_{(2)} < x_{(3)} < \ldots < x_{(n)}$; the likelihood function tends to ∞ as parameters (μ,σ,γ) approach $(-\infty,\infty,x_{(1)})$. This global maximum leads to inadmissible estimates $\hat{\mu} = -\infty$, $\sigma = \infty$ and $\gamma = x_{(1)}$. However, when we equate the three partial derivatives of the log-likelihood function to 0, we obtain the local MLEs [21].

The log-likelihood function is:

$$\log L(\mu,\sigma,\gamma|x_i) = \sum_{i=1}^{n} \left\{ -\log(x_i-\gamma) - \frac{1}{2}\log(2\pi) - \log\sigma - \frac{(\log(x_i-\gamma)-\mu)^2}{2\sigma^2} \right\}$$

$$\Lambda = -\sum_{i=1}^{n} \log(x_i-\gamma) - \frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\log(x_i-\gamma)-\mu)^2. \tag{4.74}$$

If we take partial derivatives, we can obtain the log-likelihood equations in the

following form:

$$\begin{aligned} \frac{d(\Lambda)}{d\mu} &= \frac{\sum_{i=1}^{n}(\log(x_i-\gamma)-\mu)}{\sigma^2}=0 \\ \hat{\mu} &= \frac{\sum_{i=1}^{n}\log(x_i-\hat{\gamma})}{n} \end{aligned} \tag{4.75}$$

$$\begin{aligned} \frac{d(\Lambda)}{d\sigma} &= -\frac{n}{\sigma}+\frac{1}{\sigma^3}\sum_{i=1}^{n}(\log(x_i-\gamma)-\mu)^2=0 \\ \hat{\sigma} &= \sqrt{\frac{\sum_{i=1}^{n}(\log(x_i-\hat{\gamma})-\hat{u})^2}{n}} \\ \hat{\sigma} &= \sqrt{\sum_{i=1}^{n}\left(\log(x_i-\hat{\gamma})-\frac{\sum_{i=1}^{n}\log(x_i-\hat{\gamma})}{n}\right)^2/n} \end{aligned} \tag{4.76}$$

$$\frac{d(\Lambda)}{d\gamma}=\frac{1}{\sigma^2}\sum_{i=1}^{n}\frac{(\log(x_i-\gamma)-\mu)}{(x_i-\gamma)}+\sum_{i=1}^{n}(x_i-\gamma)^{-1}=0 \tag{4.77}$$

It is clearly seen that we have closed-form solutions for $\hat{\mu}$ and $\hat{\sigma}$. Hence, if the location parameter γ is estimated, then $\hat{\mu}$ and $\hat{\sigma}$ can easily be computed using the above equations [52]. Therefore, we can construct a one-dimensional profile log-likelihood function by substituting Equations 4.75 and 4.76 into the log-likelihood function which is given by Equation 4.74.

$$\begin{aligned} \Lambda_P &= -\sum_{i=1}^{n}\log(x_i-\gamma)-\frac{n}{2}\log(2\pi) \\ &\quad -n\log\sqrt{\sum_{i=1}^{n}\left(\log(x_i-\hat{\gamma})-\frac{\sum_{i=1}^{n}\log(x_i-\hat{\gamma})}{n}\right)^2/n} \\ &\quad -\left(2\sum_{i=1}^{n}\left(\log(x_i-\hat{\gamma})-\frac{\sum_{i=1}^{n}\log(x_i-\hat{\gamma})}{n}\right)^2/n\right)^{-1} \\ &\quad X\sum_{i=1}^{n}\left(\log(x_i-\gamma)-\frac{\sum_{i=1}^{n}\log(x_i-\hat{\gamma})}{n}\right)^2 \end{aligned} \tag{4.78}$$

So, we can execute a one-dimensional optimization by considering only the location parameter, γ. The distribution support of the three-parameter lognormal distribution ($x > \gamma$) shows that γ should be less than each observation in the data set. Thus, when we maximize Equation 4.78, we begin searching for the values of :γ that are less than the minimum order statistic: $\gamma < x_{(1)}$.

R code for finding the MLEs of the two forms of lognormal distribution is given in Appendix A. First, the log-likelihood functions of the three and two-parameter lognormal distributions which are presented in Equations 4.78 and 4.70 respectively, are coded.

Then the optimization function, `MLElognormal(x, npar)` is implemented. We have analytical solutions for the two-parameter lognormal distribution. So, we can directly use Equations 4.71 and 4.72 to find the MLEs.

For the three-parameter lognormal distribution, we need a numerical solution. As the profile log-likelihood function; `llognormal3(x)` is one dimensional and we know the upper bound of the location parameter γ, we use the `optimize()` function that implements the Brent method. This method is useful when the optimization is one-dimensional and we have the knowledge of the bounds of the parameters. The upper bound for γ should be less than the minimum order statistic, so we begin with γ slightly smaller than that point by utilizing Equation 4.11. For the lower bound of γ, we choose a very small value. After $\hat{\gamma}$ is found, $\hat{\mu}$ and $\hat{\sigma}$ are computed using Equations 4.75 and 4.76 respectively.

It is important to discuss the quality of the MLEs for the lognormal distribution. Besides the MLE, there is extensive literature on estimation methods for the three-parameter lognormal distribution such as methods based on moments and quantiles. Although the MLE for this distribution has not been proven to have the usual asymptotic efficiency, the MLEs still have the best properties. Its variance is much smaller than that of the estimates based on moments or quantiles. Moreover the MLEs are

close to the minimal asymptotic variance for large sample size [52].

4.11. Logistic Distribution

The logistic distribution is used as a model for growth. Specifically, with a new product we often find that growth is initially slow, then gains momentum, and finally slows down when some form of equilibrium is reached. The logistic distribution is proper for these type of processes [31].

The PDF of the logistic distribution is given by:

$$f(x|\alpha,\mu) = \frac{e^{-(x-\mu)/\alpha}}{\alpha\left(1+e^{-(x-\mu)/\alpha}\right)^2} \text{ for } x,\mu \in (-\infty,+\infty) \text{ and } \alpha > 0 \tag{4.79}$$

where α is the scale and μ is the location parameter.

The logistic distribution has no shape parameter. This means that the logistic PDF has only one shape, the bell shape similar to the normal distribution. However, it is more peaked in the center and has heavier tails than the normal distribution [54].

The likelihood function for n IID observations $(x_1,\ldots,x_n)$ is:

$$L(\mu,\alpha|x_i) = \prod_{i=1}^{n} \frac{e^{-(x_i-\mu)/\alpha}}{\alpha\left(1+e^{-(x_i-\mu)/\alpha}\right)^2} \,.$$

Therefore, the log-likelihood function is in the following form.

$$\begin{aligned} \log L(\mu,\alpha|x_i) = \Lambda &= \sum_{i=1}^{n}\left\{-\frac{(x_i-\mu)}{\alpha} - \log(\alpha) - 2\log\left(1+e^{-(x_i-\mu)/\alpha}\right)\right\} \\ \Lambda &= -\sum_{i=1}^{n}\frac{(x_i-\mu)}{\alpha} - n\log(\alpha) - 2\sum_{i=1}^{n}\log\left(1+e^{-(x_i-\mu)/\alpha}\right) \end{aligned} \tag{4.80}$$

Taking the partial derivatives of Λ with respect to α and μ, we obtain:

$$\begin{aligned}
\frac{d(\Lambda)}{d\alpha} &= \sum_{i=1}^{n} \frac{(x_i-\mu)}{\alpha^2} - \frac{n}{\alpha} - \frac{2}{\alpha^2}\sum_{i=1}^{n}\left\{\frac{(x_i-\mu)e^{-(x_i-\mu)/\alpha}}{1+e^{-(x_i-\mu)/\alpha}}\right\} = 0 \\
\frac{d(\Lambda)}{d\mu} &= \frac{n}{\alpha} - \frac{2}{\alpha}\left(\frac{e^{-(x_i-\mu)/\alpha}}{1+e^{-(x_i-\mu)/\alpha}}\right) = 0 \, .
\end{aligned}$$

As there is no closed-form solution, the MLEs can only be obtained numerically [55]. So, we need starting values for the optimization. We can use the method of moments estimators since they are in closed-form. They are given as

$$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4.81}$$

$$\hat{\alpha} = \frac{\sqrt{3}}{\pi}\left(\frac{\sum_{i=1}^{n} x_i^2}{n} - (\hat{\mu})^2\right)^{1/2} . \tag{4.82}$$

R code for finding the MLEs of the logistic distribution is given in Appendix A. Firstly, the log-likelihood function which is presented by Equation 4.80 is coded.

Then the optimization function, `MLElogistic(x)` is coded to execute the parameter estimation of the logistic distribution. The starting values are calculated inside this function as `starting.logistic`. Equations 4.81 and 4.82 are utilized for the initial values.

To carry out the optimization, the `optim()` function with the Nelder-Mead method is utilized. The optimization function, minimizes the negative log-likelihood function; `llogistic(x)`, using the starting values, `starting.logistic`.

4.12. Rayleigh Distribution

The Rayleigh distribution is a special case of the Weibull distribution. It has many applications, such as, in life testing, in clinical studies dealing with cancer patients, in

communication engineering and in applied statistics. It has a linearly increasing failure rate that enables the study of components which age rapidly with time [56].

The Rayleigh distribution is defined in the interval $[0, \infty)$ and parameterized by the scale parameter in the standard form. If we add the location parameter γ to the standard form, the flexible form of the Rayleigh distribution is obtained which is defined in the interval $[\gamma, \infty)$.

4.12.1. One-Parameter Rayleigh Distribution

The one-parameter PDF of the Rayleigh distribution is given by

$$f(x|\sigma) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2} \text{ if } x \geq 0 \text{ and } \sigma > 0 \tag{4.83}$$

where σ is the scale parameter.

The likelihood function for n IID observations $(x_1, \ldots, x_n)$ is:

$$L(\sigma|x_i) = \prod_{i=1}^{n} \frac{x_i}{\sigma^2} e^{-x_i^2/2\sigma^2} .$$

Therefore, the log-likelihood function is in the following form:

$$\begin{aligned} \log L(\sigma|x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ \log x_i - 2\log\sigma - \frac{x_i^2}{2\sigma^2} \right\} \\ \Lambda &= \sum_{i=1}^{n} \log x_i - 2n\log\sigma - \sum_{i=1}^{n} \frac{x_i^2}{2\sigma^2} . \end{aligned} \tag{4.84}$$

Taking partial derivative of Λ with respect to σ obtains the explicit formula

below [57].

$$\begin{aligned} \frac{d(\Lambda)}{d\sigma} &= -\frac{2n}{\sigma} + \sum_{i=1}^{n} \frac{x_i^2}{\sigma^3} = 0 \\ \hat{\sigma} &= \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{2n}} \end{aligned} \tag{4.85}$$

As we have a closed-form solution, we directly utilize it to find the MLE of the scale parameter, σ.

4.12.2. Two-Parameter Rayleigh Distribution

The two-parameter PDF of the Rayleigh distribution is given by

$$f(x|\gamma,\sigma) = \frac{(x-\gamma)}{\sigma^2} e^{-(x-\gamma)^2/2\sigma^2} \text{ if } x \geq \gamma \text{ and } \sigma > 0 \tag{4.86}$$

where γ is the location and σ is the scale parameters respectively.

The likelihood function for n IID observations $(x_1, \ldots, x_n)$ is:

$$L(\gamma,\sigma|x_i) = \prod_{i=1}^{n} \frac{(x_i-\gamma)}{\sigma^2} e^{-(x_i-\gamma)^2/2\sigma^2}.$$

Therefore, the log-likelihood function is in the following form:

$$\begin{aligned} \log L(\gamma,\sigma|x_i) = \Lambda &= \sum_{i=1}^{n} \left\{ \log(x_i-\gamma) - 2\log\sigma - \frac{(x_i-\gamma)^2}{2\sigma^2} \right\} \\ \Lambda &= \sum_{i=1}^{n} \log(x_i-\gamma) - 2n\log\sigma - \sum_{i=1}^{n} \frac{(x_i-\gamma)^2}{2\sigma^2} \,. \end{aligned} \tag{4.87}$$

Taking partial derivative of Λ with respect to the scale parameter, we obtain the

explicit formula:

$$\begin{aligned} \frac{d(\Lambda)}{d\sigma} &= -\frac{2n}{\sigma} + \sum_{i=1}^{n} \frac{(x_i - \gamma)^2}{\sigma^3} = 0 \\ \hat{\sigma} &= \sqrt{\frac{\sum_{i=1}^{n} (x_i - \gamma)^2}{2n}}. \end{aligned} \tag{4.88}$$

However, we do not find a closed-form solution for the location parameter γ. So this means, we have to use numerical optimization.

We can construct the profile log-likelihood by substituting the closed-form solution of the scale parameter which is given in Equation 4.88 into 4.87, thus we obtain:

$$\Lambda_P = \sum_{i=1}^{n} \log(x_i - \gamma) - 2n \log\left(\sqrt{\frac{\sum_{i=1}^{n} (x_i - \gamma)^2}{2n}}\right) - \sum_{i=1}^{n} \frac{(x_i - \gamma)^2}{2\left(\sqrt{\frac{\sum_{i=1}^{n} (x_i - \gamma)^2}{2n}}\right)^2}. \tag{4.89}$$

This means, we execute a one-dimensional optimization by considering only the location parameter, γ. The distribution support of the two-parameter Rayleigh distribution ($x \geq \gamma$) shows that γ should be less than or equal to any observation in the data set. So the highest value for γ is equal to the minimum order statistic $x_{(1)}$. Thus, when we maximize Equation 4.89, we begin searching for the values of γ that are less than or equal to the minimum order statistic: $\gamma \leq x_{(1)}$.

R code for finding the MLEs of the two forms of Rayleigh distribution is given in Appendix A. Firstly, the log-likelihood functions of the two and one-parameter Rayleigh distributions which are presented by Equations 4.89 and 4.84 respectively, are coded.

Then the optimization function, `MLErayleigh(x, npar)` is coded to implement the parameter estimation for both forms of the Rayleigh distribution. We have an

analytical solution for the one-parameter Rayleigh distribution. So, we can directly use Equation 4.85 to find the MLE of the scale parameter, σ.

For the two-parameter Rayleigh distribution, we need a numerical solution. As the profile log-likelihood function: `lrayleigh2(x)` is one dimensional and we know the upper limit of the location parameter γ, we use the `optimize()` function that implements the Brent method. The upper limit for γ is the minimum order statistic, so we begin slightly left of this point by utilizing Equation 4.11. For the lower limit of γ, we choose a very small value. After $\hat{\gamma}$ is found, $\hat{\sigma}$ is computed using Equation 4.88.

It is important to note that the MLEs of the two-parameter Rayleigh distribution are biased. Alternative estimation approaches are possible (e.g. modified likelihood estimators) to reduce this bias.

However, the MLEs are asymptotically unbiased and efficient [58]. Mahdi and Cenac [55] indicated that the mean square errors obtained by the MLE are smaller than the other estimation methods such as the method of moments and the probability weighted moment methods for all sample sizes. In addition, for sample size ($n > 100$), the bias is considerably decreased. Hence for the two-parameter Rayleigh distribution, the MLE is the most preferable method.

5. GOODNESS OF FIT TESTS

Goodness of fit tests provide helpful guidance for evaluating the suitability of a candidate model. In this section we present the three widely used tests: The Chi-Square, the Kolmogorov-Smirnov and the Anderson-Darling tests. In our Fit All function, we prefer to use the Chi-Square test as it performs better.

All these three tests consider the null hypothesis H_0 that the data follow the specified distribution and the alternative hypothesis H_1 that the data do not follow the specified distribution. Then the P-value for the corresponding test statistic is computed and compared to the level of significance: α. If the P-value exceeds this level of significance, it is concluded that the specified distribution has an accaptable fit. Conversely, it is concluded that the specified distribution has a poor fit and the data are not represented by this distribution.

It is important to note that choosing a reasonable level of significance is a vital part of the goodness of fit tests as it directly affects the test decision. If very little data are available, then a goodness of fit test is unlikely to reject any candidate distribution. Conversely, if a lot of data are available, then a goodness of fit test will likely to reject all candidate distributions [2]. Therefore, it is especially important to reflect the effect of the sample size to the value of the level of significance.

5.1. Chi-Square Test

The Chi-Square is a statistical test that formalizes the intuitive idea of comparing the histogram of the data to the shape of the candidate mass function [2].

This test procedure begins by arranging the n observations into a set of k class

intervals. The Chi-Square test statistic χ_0^2 is given by

$$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{5.1}$$

where O_i is the observed frequency in the i th class interval, E_i is the expected frequency in that class interval and k is the number of class intervals (cells).

χ_0^2 approximately follows the chi-square distribution with $(k - s - 1)$ degrees of freedom where s represents the number of parameters of the hypothesized distribution estimated by sample statistic. Obviously, if the fit is good, we expect the value of χ_0^2 to be small.

Although there is no optimal rule for defining the number of class intervals, there are common recommendations with respect to sample size n. For instance, if $n > 100$, the number of class intervals can be set to $k = \sqrt{n}$.

For a continuous distribution, the equiprobable approach can be used. This means that each interval has the same probability and different width of class intervals. To ensure the validity of this test, the minimum size of the expected frequecy is recommended to be $E_i \geq 5$ for each class interval [5].

After the Chi-Square test statistic is computed, we find the corresponding P-value for $(k - s - 1)$ degrees of freedom from the table of the Chi-Square distribution. If this P-value is greater than α, we conclude that the specified distribution has a proper fit.

In our Fit All function, we perform the Chi-Square test to assess the goodness of fit. Since all the distributions in our Fit All function are continuous, the equiprobable approach is followed to calculate the expected frequencies. The simple formula that our Fit All function uses for finding the level of significance is:

$$\alpha = \frac{1000}{n} \, \% \tag{5.2}$$

where n denotes the sample size. If the sample size is larger than 20000, we take the constant value; $\alpha = 0.05\%$ for the level of significance.

Now we explain briefly how our R-code performs the Chi-Square test: We coded a general function that performs the Chi-Square test for all 25 distributions in our Fit All function. In order to avoid repetition, we present only the R-code for the normal distribution (see Appendix A).

First we arrange the data into a set of intervals. Instead of finding the endpoints of the class intervals, we follow a simpler operation: Firstly we use the `unif.normal` function that computes the values of the CDFs by using the MLEs of the normal distribution. Then, we find the number of class intervals by using $k = \sqrt{n}$. Since this value should be an integer, we use the $\texttt{k} = \texttt{floor}(\sqrt{\texttt{n}})$ function in R. Afterwards the observed frequencies are computed by $\texttt{obs.normal} = \texttt{table}(\texttt{floor}(\texttt{unif.normal} * \texttt{k}))$ and the expected frequencies are simply equal to n/k.

Then, we use the `chisq.test()` function in R which performs Equation 5.1. We compute the Chi-Square test statistic; `ct.normal` and the corresponding P-value by using $1 - \texttt{pchisq}()$. In the end, we compare the Chi-Square P-value with the level of significance. If it is greater than this level, we conclude that the normal distribution has an acceptable fit for the given data. Otherwise, we reject it. The results display the Chi-Square P-value and the Chi-Square test decision which is either accept or reject.

An attractive feature of the Chi-Square test is that it can be applied to any univariate discrete or continuous distribution for which we can calculate the cumulative distribution function. However, it has some disadvantages. Firstly, this test is applied to binned data, so the value of the Chi-Square test statistic depends on how observations are arranged into a set of class intervals. Secondly, it is significant for large sample sizes [5].

5.2. Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test formalizes the idea behind q-q plot. It compares the cumulative empirical distribution and the cumulative hypothesized distribution. If the maximum difference between these CDFs exceeds a critical Kolmogorov-Smirnov value, then the observed distribution is assumed to come from the hypothesized distribution. In Figure 5.1, this test is used to determine whether or not a normal distribution provides a good fit for the observed data.

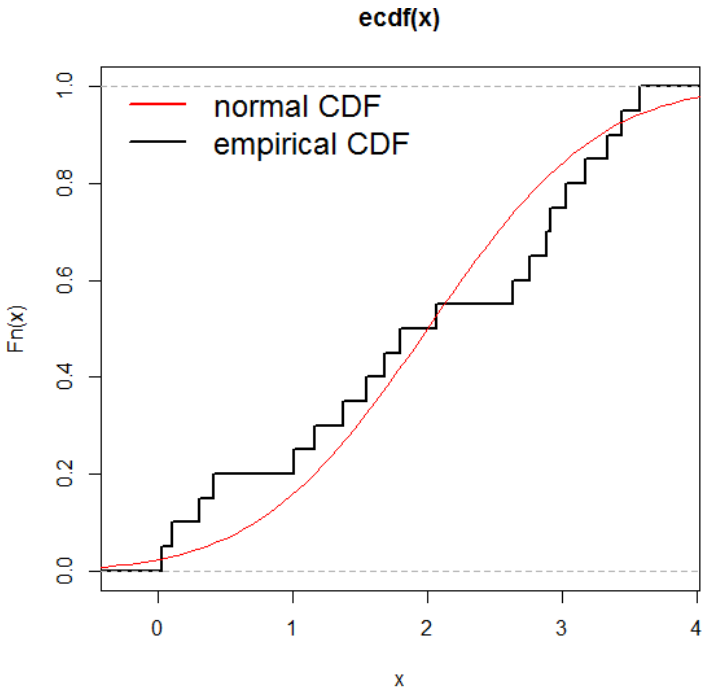

Figure 5.1. Kolmogorov-Smirnov test.

The Kolmogorov-Smirnov test statistic D_n is defined by

$$D_n = \sup_{x \in R} \left| \hat{F}(x) - F_0(x) \right| \tag{5.3}$$

where $\hat{F}$ is the empirical CDF and F_0 is the hypothesized CDF. The empirical CDF,

$\hat{F}$ is defined by

$$\hat{F}(x) = \frac{\#(\text{i: } x_i \leq x)}{n} \tag{5.4}$$

Note that the supremum must occur at one of the observed values x_i or to the left of x_i. Therefore, Equation 5.3 can be rearranged as

$$D_n = \max_{1 \leq i \leq n} \left(F_0(x_i) - \frac{i-1}{n}, \frac{i}{n} - F_0(x_i) \right). \tag{5.5}$$

The hypothesized distribution is rejected at the chosen significance level α if the test statistic D_n is greater than the critical value obtained from d table. The d table has two parameters, the sample size and the level of significance. Common significance levels are 0.01 and 0.05.

An attractive feature of this test is the distribution of the Kolmogorov-Smirnov test statistic itself does not depend on the underlying cumulative distribution function being tested. Moreover, the Kolmogorov-Smirnov test is valid even for small samples [2].

Despite these advantages, the Kolmogorov-Smirnov test has several important limitations. First and the most important, the standard tables of critical values used for this test are only valid if a data set is from a completely specified distribution. When parameter estimates have been used, the critical values for this test are biased since they do not depend on the specific distribution being tested. Secondly, the Kolmogorov-Smirnov test is only applicable to continuous distributions. Thirdly, it tends to be more sensitive near the center of the distribution than in the tails [59].

It is generally accepted that the Kolmogorov-Smirnov test has less ability to properly fit data than other techniques (e.g: the Chi-Square test). For this reason it

should be utilized only when the number of data points is extremely limited and the Chi-square test cannot be properly applied [5].

5.3. Anderson-Darling Test

The Anderson-Darling test is a modification of the Kolmogorov-Smirnov test. Similar to the Kolmogorov-Smirnov test, it is based on the difference between the empirical CDF and and hypothesized CDF, but it also gives weight to the tails of the distributions. The Anderson-Darling statistic A^2 is defined by

$$A^2 = n \int_{-\infty}^{\infty} (\hat{F}(x) - F_0(x))^2 [F_0(x)(1 - F_0(x))]^{-1} dF_0(x) \tag{5.6}$$

where $\hat{F}$ is the empirical CDF and F_0 is the hypothesized CDF.

It is computed as

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^{n} (2i - 1) \left[\log F_0(x_i) + \log(1 - F_0(x_{n-i+1})) \right] \tag{5.7}$$

where the x_i are the ordered data.

If A^2 is smaller than the critical value, it is concluded that data may come from the hypothesized distribution. Conversely, the hypothesized distribution is rejected as usual [59].

The Anderson-Darling test removes the limitation of the Kolmogorov Smirnov test that all the parameters of the hypothesized distribution are known [2]. However, the critical values depend on the candidate distribution. Currently tables of critical values are available for the normal, lognormal, exponential, Weibull, extreme value type I, and logistic distributions. For many distributions, they are not easy to find [59]. Therefore, some softwares that perform the Anderson-Darling test utilize the same critical values for all distributions (e.g EasyFit) [60].

6. THE FIT ALL FUNCTION

The Fit All function is implemented in R which is a free software environment for statistical computing. This function implements input modeling and thus helps the user to find proper models for his input variables.

The Fit All function automatically specifies and ranks probability distributions and identifies the best fitting distribution. It also tells the user whether the fitted distribution is good enough to actually use it in the simulation model. If it does not provide an adequate fit, then the Fit All function can suggest to use a non-parametric model. As it was stated before, there is not always an input model for any random input. Therefore the model that the Fit All function selects should be interpreted with care.

6.1. Main Structure of The Fit All Function

It is useful to explain the main structure of the Fit All function from the statistical point of view. It includes totally 25 continuous distributions. The MLE is executed to estimate parameters for all these distributions. In order to rank them from best to worst, AIC is utilized and to learn whether the best fitting distribution is actually a good representation of the data, the Chi-Square test is implemented. If the best fitting distribution passes the Chi-Square test, it is selected as the input model and the input modeling process ends. Otherwise, the Fit All function recommends the non-parametric model which is called "resampling with noise". The structure of the Fit All function can be clearly summarized by the flow chart given in Figure 6.1.

6.2. Data Required by the Fit All Function

The Fit All function is designed to deal with independent and identically distributed data coming from continuous distributions. It is important to consider the

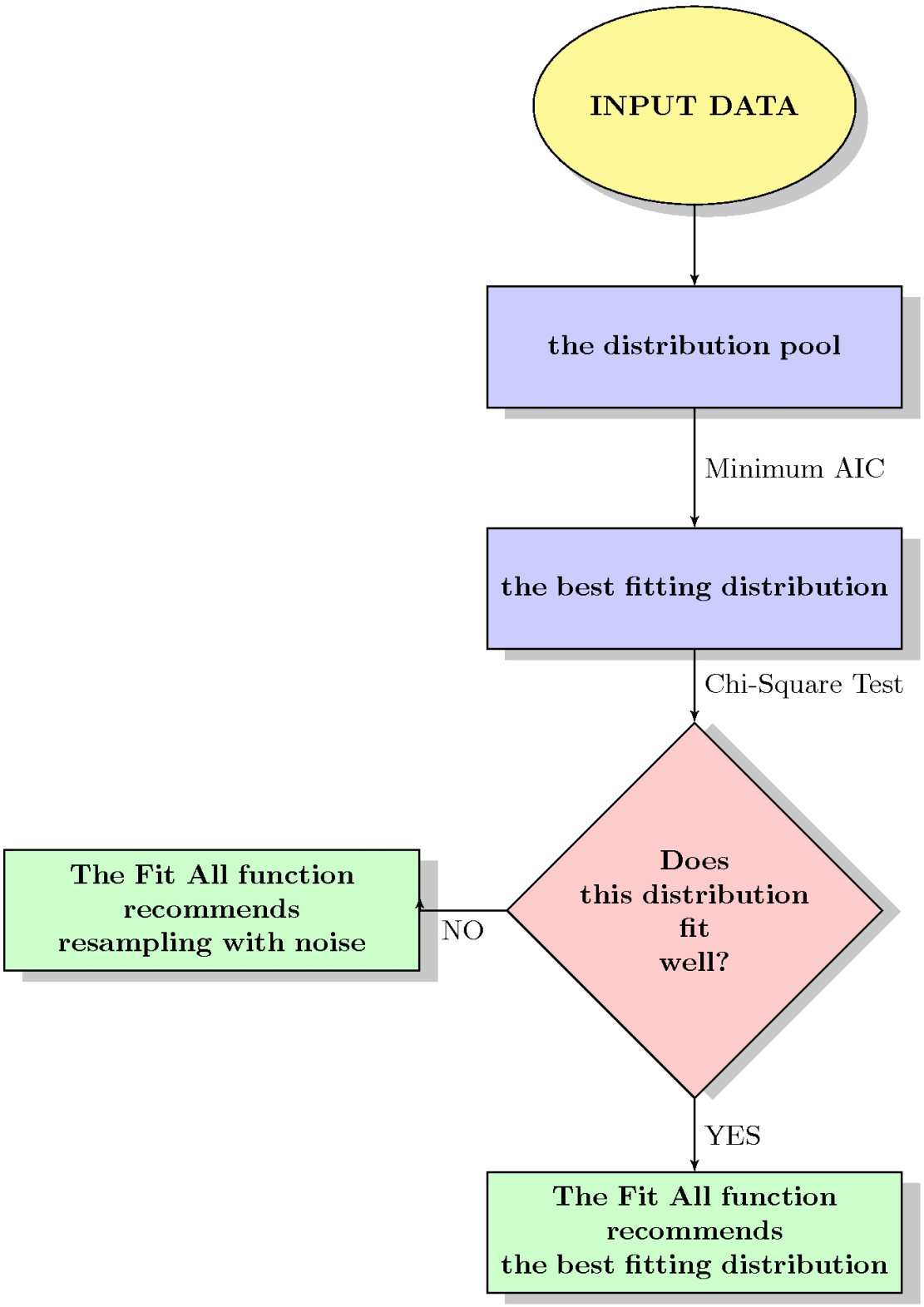

Figure 6.1. Main structure of the Fit All function.

number of observations. If the number of observations in the data set is too small, it is not sensible to use the model directly that the Fit All function recommends. Instead, it is recommended to obtain extra knowledge on the process (i.e: expert knowledge, engineering data etc.). However, if the number of observations in the data set is large enough, the Fit All function finds a proper model without the necessity of extra knowledge. This is obviously related to the fact that the larger the data sets, the more closely they match the distribution of the parent population.

6.3. Distribution Pool of the Fit All Function

As our Fit All function designed to model continuous distributions, it contains the most common continuous distributions. There are 12 standard distributions in the Fit All function given in the first column of Table 6.1.

Table 6.1. Distribution pool of the Fit All function.

Standard Distribution	**Flexible Form**	**Negative Form**
Beta (2P)	Beta (4P)	-
Uniform (1P)	Uniform (2P)	-
Triangular(3P)	-	-
Gamma (2P)	Gamma (3P)	N-Gamma (3P)
Erlang (2P)	-	-
Weibull (2P)	Weibull (3P)	N-Weibull (3P)
Exponential (1P)	Exponential (2P)	N-Exponential (2P)
Normal(2P)	-	-
Lognormal (2P)	Lognormal (3P)	N-Lognormal (3P)
Student's t (1P)	Student's t (3P)	-
Logistic (2P)	-	-
Rayleigh (1P)	Rayleigh (2P)	N-Rayleigh (2P)

Moreover, for the cases where these standard distributions are not flexible enough to represent key characteristics of the data, their flexible forms are also included to the Fit All function. (see the second column of Table 6.1). The flexible forms are obtained by adding location and/or scale parameters to some standard distributions (if possible). As a result the distribution pool of the Fit All function is enriched. With

these distributions, there are totally 20 distributions included in the Fit All function.

For instance, the gamma distribution includes two parameters in the standard form which is expressed by Gamma (2P) in Table 6.1. By adding a location parameter to it, the flexible form of the gamma distribution: Gamma (3P) that includes three parameters is obtained.

6.3.1. Fitting Negative Versions of Flexible Forms of Distributions to the Data

It is sensible to mention the relation between the types of distributions and the skewness of the data. If the data are positively skewed, there are many distributions to represent these data (e.g: the lognormal, the gamma and the Weibull etc). However, if the data exhibit negative skewness, the choice of distributions is limited. In our distribution pool, the beta distribution is the only possibility that allows for negative skewness [61]. In order to overcome this limitation, the Fit All function fits the negative version of flexible forms of the distributions to the data. In the third column of Table 6.1, we present the names of these distributions. With these distributions, there are totally 25 distributions included in the Fit All function.

6.4. Results Obtained by the Fit All Function

The Fit All function mainly gives two results: Firstly it recommends the most proper model and secondly it gives a summary of all fitted distributions.

6.4.1. Recommendation of the Most Proper Model

After the user calls the Fit All function, there are two cases for model selection. The Fit All function recommends the user either a parametric or a non-parametric model with respect to the following scenarios:

- The Fit All function recommends the best fitting distribution: The best fitting distribution is the one which has the minimum AIC value among all distributions. If the P-value that is computed for the best fitting distribution exceeds the level of significance (α), then it is confirmed that the data are presented quite well by this distribution. For this case, the Fit All function displays the name of the best fitting distribution, its estimated parameters and AIC value.
- The Fit All function recommends resampling with noise: If the P-value that is computed for the best fitting distribution is smaller than the level of significance, the Fit All function concludes that the data are not presented by this distribution. In this case the non-parametric model is recommended which is called resampling with noise. We resample the data and add to each of the resampled observations some noise coming from a standard normal distribution. By introducing the bandwidth value b, we select the variance of the random noise (see Section 2.1.1). In this case, the Fit All function displays the value of the optimal bandwidth calculated for the sample.

6.4.2. The Summary of All Fitted Distributions

Besides recommending the most proper model, the Fit All function also presents the summary of all distributions that are fitted to the data in a list. In that list, all fitted distributions are in ascending order with respect to their AIC values. The best fitting distribution is in the first row as it has the minimum AIC value and the worst fitting distribution is in the last row as it has the maximum AIC value. For each fitted distribution, the MLEs, the P-value that is obtained by the Chi-Square Test, the Chi-Square test decision and the difference to the smallest AIC value are displayed respectively.

6.4.2.1. The Maximum Likelihood Estimates. Firstly, we coded the log-likelihood function for each distribution. If the distribution has an analytical solution for the MLEs (e.g: exponential, normal), then we use it directly. However, if the distribution involves many parameters and/or it is highly nonlinear (e.g: Student's-t, beta), a numerical so-

lution is required to find the MLEs. The numerical solution is found iteratively, starting with a tentative θ_0 and updating it through various iterations until an optimal value, $\hat{\theta}$, is found. To carry out this operation, we utilize the `optim`() or the `optimize`() function of the statistical software R. By giving reasonable starting values for the parameters, we initiate the optimization. For multi-dimensional optimization problems, the Nelder Mead method is utilized for the `optim`(). For one-dimensional optimization problems, the BFGS inside the `optim`() or the Brent inside the `optimize`() are utilized. After optimization finishes, the Fit All function presents the MLEs and AIC values for each fitted distribution.

6.4.2.2. The P-value Obtained by Chi-Square Test. To implement the Chi-Square test, we need to arrange the data into a set of intervals. If the number of observations: n, are greater than 100; the number of class intervals are computed as $k = \sqrt{n}$. Since this value must be an integer, we find it as $k = \texttt{floor}(\sqrt{\texttt{n}})$ in the statistical software R. As all distributions in the Fit All function are continuous, we use the same probability for each class interval to find the expected frequency. Let the number of parameters for the corresponding distribution be denoted by s. After we obtain observed and expected frequencies, we simply calculate the Chi-Square test statistic and find the corresponding P-value for $(k - s - 1)$ degrees of freedom.

6.4.2.3. Chi-Square Test Decision. A critical issue that directly affects the acceptance or rejection of the best fitting distribution is that of the level of significance: α. The level of significance is selected by considering the effect of the sample size. If very little data are available, then a goodness of fit test is unlikely to reject any candidate distribution. Conversely, if a lot of data are available, then a goodness of fit test will likely reject all candidate distributions (see Banks [2]). Therefore, it is reasonable to prefer small α values for large sample sizes and large α values for small sample sizes. In order to achieve this balance, the Fit All function uses the simple formula given by Equation 5.1 for finding the level of significance (see Section 5.1).

If the P-value that is obtained by the Chi-Square Test exceeds the level of sig-

nificance, the Chi-Square test decision is displayed as accept which indicates that the corresponding distribution passes the Chi-Square test. Conversely, reject is displayed which shows that the corresponding distribution fails.

6.4.2.4. Delta AIC Value. This value helps the user to see how far away the other distributions are from the the best fitting distribution. By simply subtracting the minimum AIC value from each AIC value, the delta AIC value is computed for each fitted distribution. Of course, this value is zero for the best fitting distribution. If the delta AIC value is small for a distribution, this shows that it is also a possible input model.

6.5. R Code for the Fit All Function

R code for the Fit All function mainly includes two parts. Firstly, we obtain the MLEs which are presented in detail in Chapter 4. We gather all the MLEs and AIC values in the same list which is called `alldistributions(x)`. Secondly, we perform the Chi-Square test for all the fitted distributions which is explained in Chapter 5. We gather all Chi-Square test results in a list which is called `cs.test(x)`.

After we have obtained the MLEs and Chi-Square test results, we combine them in a single list and order all fitted distributions with respect to their AIC values. The best fitting distribution is the one with the minimum AIC value. If this distribution passes the Chi-Square test, then we recommend it to the user. Conversely we recommend resampling with noise with the calculated bandwidth value. In either case, we present the summary of all fitted distributions in the same list by presenting the MLEs, the P-values, the Chi-Square test decisions and the delta AIC values.

Once the user imports the data into a vector `x`, he can use the command `fitall(x)`, and gets the results at once. In Appendix A, we present R code for the Fit All function.

6.6. Examples for Using the Fit All Function

In this section, we will present some examples to show how input modeling is performed with the Fit All function. Firstly we present the standard use of the Fit All function, then we show how the Fit All function can be used when prior knowledge is available.

6.6.1. Standard Use of the Fit All Function

We work on the stock data which were obtained from http://finance.yahoo.com/.

Example 1: The stock data include $n = 503$ observations. When we import these data into a vector x and call `fitall(x)`, the results are:

```
 fitall(x)
$model
[1] "It is recommended to use the Student's-t(3) distribution with the parameters:
 location=0,scale=0.007,degrees of freedom=5.095"

$AIC
[1] -3381.014

$parameters
[1] 0.000 0.007 5.095
```

```
$summary
                       AIC           p1           p2           p3       p4   CS.P.value CS.Decision   Delta.AIC
Student's-t(3)   -3381.0136 -4.580765e-04  6.803600e-03  5.094871e+00       NA 7.666919e-01      accept    0.000000
Logistic         -3375.3526 -4.288440e-04  4.666496e-03           NA       NA 7.124366e-01      accept    5.661011
Lognormal(3)     -3333.4498 -2.174229e+00  7.697746e-02 -1.143029e-01       NA 4.862258e-02      accept   47.563830
Gamma(3)         -3331.6098  9.833813e+01 -8.751623e-02  8.872227e-04       NA 5.918054e-02      accept   49.403810
Erlang(3)        -3331.6067  9.800000e+01 -8.751623e-02  8.902839e-04       NA 6.312201e-02      accept   49.406923
Beta(4)          -3328.8958 -8.509504e-02  1.207007e+00  8.662948e+01 1232.555 2.874794e-02      accept   52.117789
Normal           -3321.1767 -2.684064e-04  8.877075e-03           NA       NA 1.374951e-01      accept   59.836913
N-Lognormal(3)   -3319.1767  1.339414e+01  1.352904e-08 -6.561496e+05       NA 1.057681e-01      accept   61.836914
N-Gamma(3)       -3319.0276  3.294862e+06 -1.615026e+01  4.901730e-06       NA 1.167126e-01      accept   61.985973
N-Weibull(3)     -3293.0086  8.749469e+00 -7.276975e-02  7.677820e-02       NA 3.645421e-05      reject   88.004963
Weibull(3)       -3260.9925  3.630985e+00 -3.346161e-02  3.640010e-02       NA 3.204820e-08      reject  120.021131
Rayieigh(2)      -3027.8902 -3.205495e-02  2.333652e-02           NA       NA 0.000000e+00      reject  353.123363
Triangular       -2897.4572 -3.225575e-02  6.004430e-02 -4.517818e-03       NA 0.000000e+00      reject  483.556387
N-Rayleigh(2)    -2470.7659 -6.000764e-02  4.308134e-02           NA       NA 0.000000e+00      reject  910.247653
Exponential(2)   -2463.4151  3.158552e+01 -3.192848e-02           NA       NA 0.000000e+00      reject  917.598513
Uniform(2)       -2398.9740 -3.192848e-02  5.982571e-02           NA       NA 0.000000e+00      reject  982.039595
N-Exponential(2) -1818.7144  1.664056e+01 -5.982571e-02           NA       NA 0.000000e+00      reject 1562.299156
```

As it is shown, the Fit All function recommends the three-parameter Student's t-distribution with the location=0, the scale=0.007 and the degrees of freedom=5.095 for these stock data. From the summary list, we see all fitted distributions. The Logistic distribution is in rank two. It has the P-value of 0.713 and the delta AIC value of 5.6, hence the user may select this distribution as a possible model for these data.

If a parametric model is recommended by the Fit All function as in this example, the user can resample new data by simply using R-code below:

```
Y <- [best_fitting_distr(n=1000,shape)]*scale + location
```

where `best_fitting_distr`() denotes the random variate generation function for that distribution. As the best fitting model is the three-parameter Student's t-distribution for this example, the user can easily resample these data in the vector Y as

```
Y <- rt(1000,5.095)*0.007 + 0
```

Here `rt`() is the function that generates random samples from the Student's t-distribution in R. For example, if the best fitting distribution is the gamma distribution, then the user should use `rgamma`() that generates random samples from the gamma distribution.

Example 2: The stock data include $n = 503$ observations. When we import these data into a vector x and call `fitall(x)`, the results are:

```
 fitall(x)
$model
[1] "It is recommended to use resampling with noise with bandwidth value of 0.006"

$bandwidth
[1] 0.006
```

```
$summary
                        AIC            p1            p2            p3       p4  CS.P.value CS.Decision Delta.AIC
Student's-t(3)   -1844.2992 -1.155835e-03  1.076859e-02  8.768394e-01       NA 8.09035e-09      reject     0.000
Logistic           254.9777 -3.000611e-05  1.277750e-01            NA       NA 0.00000e+00      reject  2099.277
Normal             836.9881 -1.226788e-05  5.538184e-01            NA       NA 0.00000e+00      reject  2681.287
Lognormal(3)       838.9881  9.901795e+00  2.773781e-05 -1.996619e+04       NA 0.00000e+00      reject  2683.287
N-Lognormal(3)     838.9881  1.400196e+01  4.596126e-07 -1.204968e+06       NA 0.00000e+00      reject  2683.287
N-Gamma(3)         839.1503  2.766522e+04 -9.216040e+01  3.331274e-03       NA 0.00000e+00      reject  2683.449
Gamma(3)           839.3081  1.379559e+04 -6.516423e+01  4.723555e-03       NA 0.00000e+00      reject  2683.607
Erlang(3)          839.3082  1.379500e+04 -6.516423e+01  4.723756e-03       NA 0.00000e+00      reject  2683.607
Beta(4)            841.1842 -7.875540e+01  8.397853e+01  1.038425e+04 11073.01 0.00000e+00      reject  2685.483
N-Weibull(3)       980.6887  4.811528e+00 -3.068542e+00  3.280046e+00       NA 0.00000e+00      reject  2824.988
Weibull(3)         980.7231  4.810372e+00 -3.067789e+00  3.279288e+00       NA 0.00000e+00      reject  2825.022
Student's-t(1)    1080.2029  4.600108e+01            NA            NA       NA 0.00000e+00      reject  2924.502
Triangular        1116.9490 -2.574094e+00  2.575496e+00 -1.339428e-06       NA 0.00000e+00      reject  2961.248
Rayieigh(2)       1363.9943 -2.526871e+00  1.829171e+00            NA       NA 0.00000e+00      reject  3208.294
N-Rayleigh(2)     1364.5499 -2.530960e+00  1.832012e+00            NA       NA 0.00000e+00      reject  3208.849
Uniform(2)        1626.4801 -2.504067e+00  2.512788e+00            NA       NA 0.00000e+00      reject  3470.779
Exponential(2)    1933.4188  3.993523e-01 -2.504067e+00            NA       NA 0.00000e+00      reject  3777.718
N-Exponential(2)  1936.9262  3.979624e-01 -2.512788e+00            NA       NA 0.00000e+00      reject  3781.225
```

For these stock data, the Fit All function could not find a proper parametric model, hence it recommends to use resampling with noise with the bandwidth value of 0.006. From the summary list, we see that three-parameter Student's t-distribution is the best fitting parametric model as it is presented in the first row. However the corresponding P-value for this distribution is approximately zero that is why it is rejected by the Chi-Square test.

If the non-parametric model is recommended by the Fit All function as in this example, the user can resample the data simply using the R-code:

```
Y <- x[ceiling(runif(1000)*n)]+b*rnorm(1000)
```

where x is the vector that contains the data, n is the sample size and b is the value of the bandwidth.

As the recommended model is resampling with noise with the bandwidth value of 0.006 and the sample size, n equals to 503 for this example, the user can easily resample these data in the vector Y as

```
Y <- x[ceiling(runif(1000)*503)]+0.006*rnorm(1000) .
```

6.6.2. Using the Fit All Function with Prior Knowledge

It is important to emphasize that the Fit All function does not have prior knowledge on the distributions. It automatically tries all the distributions available considering the data that are given. So, it suggests an input model that conforms as closely as possible to the given data.

However, there may be situations where the user has the prior knowledge of the process. For example, suppose the user wants to find a proper input model for the process of customer interarrival times. It is well-known that this type of process can be often described by the exponential distribution. So it is rational for the user to expect that the exponential distribution may be a proper input model. On the other hand, if the Fit All function does not find the exponential distribution as the best fitting distribution in the results of the model selection procedure (i.e finds it in rank three or four), the user may still select this distribution as a possible choice for the input model.

Briefly, combining the Fit All function results and the prior knowledge will lead to more desirable input models rather than strictly dedicating the FitAll function results. In order to show how the Fit All function results can be evaluated with prior knowledge, we will present a small example.

Example : Consider a data set that are collected at a bank that show customer interarrival times in minutes. There are 300 observations in this data set. For this process, it is natural to expect the exponential distribution as a proper model. Once we import these data into a vector `x` and use command `fitall(x)`, we get the results:

```
 fitall(x)
$model
[1] "It is recommended to use the Gamma(3) distribution with the parameters:
```

```
 shape=0.944,location=0.008 and scale=4.399"

$AIC
[1] 1456.268

$parameters
[1] 0.944 0.008 4.399


$summary
```

	AIC	p1	p2	p3	p4	CS.P.value	CS.Decision	Delta.AIC
Gamma(3)	1456.268	9.443896e-01	8.000000e-03	4.399208e+00	NA	3.785557e-01	accept	0.0000000
Weibull(3)	1456.448	9.451796e-01	8.000000e-03	4.052490e+00	NA	2.384985e-01	accept	0.1802461
Exponential(1)	1457.679	2.402364e-01	NA	NA	NA	6.177426e-01	accept	1.4113694
Exponential(2)	1458.525	2.406990e-01	8.000000e-03	NA	NA	5.244249e-01	accept	2.2571252
Weibull(2)	1459.451	9.791264e-01	4.123785e+00	NA	NA	3.970908e-01	accept	3.1836958
Gamma(2)	1459.608	9.810606e-01	4.242925e+00	NA	NA	4.295524e-01	accept	3.3400174
Erlang(2)	1459.679	1.000000e+00	4.162567e+00	NA	NA	5.423289e-01	accept	3.4113694
Erlang(3)	1460.525	1.000000e+00	8.000000e-03	4.154567e+00	NA	4.457540e-01	accept	4.2571252
Beta(4)	1467.416	8.000000e-03	3.201889e+01	6.238628e-01	3.739003	7.841033e-05	reject	11.1482220
Lognormal(3)	1475.387	1.045229e+00	9.849254e-01	-2.591538e-01	NA	5.551482e-01	accept	19.1192261
Lognormal(2)	1508.035	8.364447e-01	1.285749e+00	NA	NA	3.252950e-03	reject	51.7669312
N-Weibull(3)	1579.120	1.220996e+02	-3.218458e+02	3.193890e+02	NA	7.860528e-08	reject	122.8523520
Rayieigh(2)	1637.247	-2.269075e+00	5.504738e+00	NA	NA	8.770762e-15	reject	180.9793631
Student's-t(3)	1642.877	2.763615e+00	2.259487e+00	2.183021e+00	NA	1.909584e-14	reject	186.6089092
Logistic	1678.137	3.453932e+00	2.120638e+00	NA	NA	0.000000e+00	reject	221.8687904
Triangular	1726.511	6.151915e-03	2.911723e+01	6.152076e-03	NA	0.000000e+00	reject	270.2428470
Normal	1742.433	4.162567e+00	4.386144e+00	NA	NA	0.000000e+00	reject	286.1656716
N-Lognormal(3)	1744.435	1.450864e+01	2.193107e-06	-1.999977e+06	NA	0.000000e+00	reject	288.1671383
N-Gamma(3)	1744.557	1.312982e+07	-1.589633e+04	1.210388e-03	NA	0.000000e+00	reject	288.2888415
Rayleigh(1)	1843.703	4.275817e+00	NA	NA	NA	0.000000e+00	reject	387.4356701
Uniform(1)	2022.398	2.900100e+01	NA	NA	NA	0.000000e+00	reject	566.1304340
Uniform(2)	2024.233	8.000000e-03	2.900100e+01	NA	NA	0.000000e+00	reject	567.9648996
Student's-t(1)	2036.535	7.030046e-01	NA	NA	NA	0.000000e+00	reject	580.2674077
N-Rayleigh(2)	2157.850	-2.909773e+01	1.790252e+01	NA	NA	0.000000e+00	reject	701.5821709
N-Exponential(2)	2531.435	4.026019e-02	-2.900100e+01	NA	NA	0.000000e+00	reject	1075.1675575

The results show that the three-parameter gamma distribution is the recommended model for these data. The three-parameter Weibull distribution is in rank two and the one-parameter exponential distribution is in rank three. These results are not surprising since the exponential distribution is a special case of both the gamma and the Weibull distributions for shape parameter $\alpha = 1$. Moreover, when we consider the delta AIC value, we see that it is about 1.4 for the one-parameter exponential distribution. This result also confirms that we can use the exponential distribution as a proper model for describing this process.

When the user has an idea about the underlying distribution as in this example,

Table 6.2. Manual model selection commands.

	Standard Distribution	**Flexible Form**	**Negative Form**
Beta	`MLEbeta(x, npar = 2)`	`MLEbeta(x, npar = 4)`	-
Uniform	`MLEuniform(x, npar = 1)`	`MLEuniform(x, npar = 2)`	-
Triangular	`MLEtriangular(x)`	-	-
Gamma	`MLEgamma(x, npar = 2)`	`MLEgamma(x, npar = 3)`	`MLEgamma(-x,npar=3)`
Erlang	`MLEerlang(x, npar = 2)`	–	-
Weibull	`MLEweibull(x, npar = 2)`	`MLEweibull(x, npar = 3)`	`MLEweibull(-x,npar=3)`
Exponential	`MLEexponential(x, npar = 1)`	`MLEexponential(x, npar = 2)`	`MLEexponential(-x,npar=2)`
Normal	`MLEnormal(x)`	-	-
Lognormal	`MLElognormal(x, npar = 2)`	`MLElognormal(x, npar = 3)`	`MLElognormal(-x,npar=3)`
Student's t	`MLEt(x, npar = 1)`	`MLEt(x, npar = 3)`	-
Logistic	`MLElogistic(x)`	-	-
Rayleigh	`MLErayleigh(x, npar = 1)`	`MLErayleigh(x, npar = 1)`	`MLErayleigh(-x,npar=2)`

there is no need to fit all distributions to the data. However, the user may want to obtain parameter values. For this case, the Fit All function enables fitting individual models to the data. Once the user imports the data into a vector x, he can use the commands in Table 6.2 for each distribution.

If we again consider the example of the customer interarrival times, it is logical to narrow the choice to the exponential distribution. So, the aim of fitting the exponential distribution to these data is to estimate the parameter value. By simply calling `MLEexponential(x, npar = 1)`, this result is easily obtained.

```
 MLEexponential(x,npar=1)
         AIC         rate
1457.6791227    0.2402364
```

7. EXPERIMENTS AND RESULTS

We compare the recommended models of the Fit All function, the Arena Input Analyzer and EasyFit for different samples. Firstly we did some experiments with samples which were generated in R from standard distributions (e.g: normal, beta, Weibull etc.). This means we know from which distribution each sample was generated, hence we can easily compare the results of these tools by considering their best fitting models. Secondly, we did some experiments by using real data and again compare the recommended models of these three tools.

We have presented how the Fit All function, the Arena Input Analyzer and EasyFit implements input modeling previously. To remind how these three tools differ form each other, we summarize their mechanisms for model selection:

- The Fit All function implements the maximum likelihood method to estimate values the parameters for all distributions. In order to rank them from best to worst, AIC is utilized. The best fitting model is that with the minimal AIC value. Then the Chi-Square test is performed for this distribution. If the best-fitting distribution passes the Chi-Square test, then it is recommended to the user. If not, the non-parametric model which is called resampling with noise is recommended.
- Arena Input Analyzer implements the method of moments, the maximum likelihood method or the least square methods for different distributions to estimate the parameters. The recommended model has the minimal square error. It is important to note that model selection is not considering the number of parameters of the distribution.
- EasyFit implements the method of moments, the maximum likelihood method, the least square methods or the method of L-moments to estimate values of parameters for different distributions. The Kolmogorov-Smirnov, the Anderson-Darling and the Chi-Square tests are applied to rank distributions from best

to worst. The user has the flexibility to choose one of these rankings. In our experiments, we preffered to use the Chi-Square test results as critical values for the Kolmogorov-Smirnov and the Anderson-Darling tests are not well-defined. Note that in the P-value calculations for all three tests, EasyFit is not considering the number of parameters of the distribution.

7.1. Experiments with Samples Generated in the Statistical Software R

We generated random samples from the normal, beta, lognormal and Weibull distributions for the sample sizes $n = 100$, $n = 250$ and $n = 1000$ in R. For each sample size, we generated 5 samples using the same parameter values; so a total of 15 samples were generated for each distribution. We report the recommended models of the Fit All function, the Arena Input Analyzer and EasyFit for each sample.

In addition, the delta AIC values; difference to the minimal AIC value obtained from the Fit All function. For example, if the sample is generated from the normal distribution and the recommended distribution is the normal distribution, delta AIC value is 0. But if the recommended distribution is the logistic distribution, then we give the delta AIC value corresponding to the normal distribution. If it is small, it is possible for this distribution to be a candidate model.

Firstly, we generate random samples from the normal distribution with parameter values: $\mu = 2$ and $\sigma = 0.8$. Table 7.1 presents the best fitting distributions with their parameter values. The results show that, the Fit All function found the best fitting distribution to a normal distribution for 12 cases out of 15. When we examine the delta AIC values for the remaining 3 cases, they are all small; so the normal distribution is still a candidate model for these samples. On the other hand, the Arena Input Analyzer and EasyFit found the normal distribution only 10 and 3 times respectively.

Secondly, we generate random samples from the lognormal distribution with parameter values: $\mu = 2$ and $\sigma = 0.4$. Table 7.2 presents the best fitting distributions

Table 7.1. Best fitting distributions for the samples generated from the normal distribution.

BEST FITTING DISTRIBUTION				
Sample	**The Fit All Function**	**Delta AIC**	**The Arena Input Analyzer**	**EasyFit**
Normal(2, 0.8) n=100	Logistic(2.09,0.45)	0.5	2 + Gamma(12.2,0.25)	-7.83 + Gamma(151,0.07)
	Normal(2.11,0.91)	0	Normal(2.11,0.91)	Gamma(5.26,0.4)
	Normal(1.99,0.74)	0	Normal(1.99,0.74)	Logistic(1.99,0.41)
	Normal(1.99,0.86)	0	Weibull(2.3,2.15)	-3.1 + Erlang(34,0.15)
	Normal(1.96,0.78)	0	0.74 + 5.32 * Beta(7.55,7.29)	-0.94 + Weibull(4.09,3.19)
Normal(2, 0.8) n=250	-7.22 + Lognormal(2.21,0.09)	1.3	Normal(1.93,0.86)	Logistic(1.93,0.48)
	Normal(1.93,0.8)	0	Normal(1.93,0.8)	Normal(1.93,0.8)
	-0.54 + Weibull(3.56,2.81)	0.1	-0.54 + Weibull(3.56,2.81)	-6.74 + Gamma(118.18,0.07)
	Normal(2.08,0.77)	0	Normal(2.08,0.77)	-7.46 + Erlang(151,0.06)
	Normal(2.04,0.81)	0	Normal(2.04,0.81)	-7.18 + Erlang(129,0.07)
Normal(2, 0.8) n=1000	Normal(1.95,0.79)	0	Normal(1.95,0.79)	-20.6 + Lognormal(3.12,0.04)
	Normal(1.97,0.81)	0	Normal(1.97,0.81)	-17.45 + Lognormal(2.97,0.04)
	Normal(1.98,0.82)	0	Normal(1.98,0.82)	Normal(1.98,0.82)
	Normal(1.98,0.80)	0	Normal(1.98,0.80)	Normal(1.98,0.80)
	Normal(2,0.77)	0	-1 + 5.78 * Beta(6.65,6.19)	-2.48 + 8.39*Beta(14.98,13.13)

with their parameter values. The results show that, the Fit All function found the best fitting distribution to be the lognormal distribution in 14 out of 15 cases. For the remaining case, the delta AIC value is small; so the normal distribution is still a candidate model for this sample. On the other hand, the Arena Input Analyzer couldn't find the lognormal distribution as the best for any run. EasyFit found the three-parameter and the two-parameter lognormal distributions as the best fitting distributions for 5 and 4 times respectively.

Table 7.2. Best fitting distributions for the samples generated from the lognormal distribution.

BEST FITTING DISTRIBUTION				
Sample	**The Fit All Function**	**Delta AIC**	**The Arena Input Analyzer**	**EasyFit**
Lognormal(2, 0.4) n=100	Lognormal(1.95,0.42)	0	2 + Erlang(3,1.89)	0.74 + Lognormal(1.83,0.47)
	Lognormal(2.01, 0.41)	0	2 + Erlang(3,2.04)	Lognormal(2.01, 0.41)
	Lognormal(2.01, 0.41)	0	2 + Erlang(3,2.05)	1.39 + Erlang(4,1.68)
	Lognormal(2.03, 0.39)	0	2 + Gamma(3.65,1.68)	Lognormal(2.02, 0.39)
	Lognormal(2.08, 0.38)	0	3 + Weibull(1.71,6.28)	2 + Erlang(4,1.69)
Lognormal(2, 0.4) n=250	Lognormal(1.99,0.38)	0	2 + Gamma(3.52,1.68)	-1.08 + Lognormal(2.14,0.33)
	Lognormal(2.01, 0.38)	0	2 + Erlang(4,1.51)	-0.49 + Lognormal(2.08,0.35)
	Lognormal(1.98, 0.4)	0	2 + Gamma(3.34,1.75)	Lognormal(1.98, 0.4)
	Lognormal(1.98, 0.39)	0	2 + 24 * Beta(2.93, 8.82)	Weibull(3.19,8.6)
	Lognormal(2, 0.4)	0	2 + Gamma(3.36,1.78)	Gamma(5.78,1.38)
Lognormal(2, 0.4) n=1000	Lognormal(1.99,0.4)	0	1 + Erlang(5,1.39)	-0.63 + Lognormal(2.08,0.37)
	2 .16+ Gamma(3.18,1.83)	3.7	2 + Gamma(3.42,1.75)	2.17 + 3.6e6*Beta(3.17,2e6)
	Lognormal(2, 0.39)	0	2 + Gamma(3.4,1.75)	1.43 + Gamma(4.7,1.5)
	Lognormal(1.98, 0.39)	0	2 + Gamma(3.39,1.72)	Lognormal(1.98, 0.39)
	Lognormal(1.99, 0.4)	0	2 + Gamma(3.26,1.84)	-0.04 + Lognormal(2, 0.4)

Thirdly, we generate random samples from the beta distribution with shape parameters 2 and 8. Table 7.3 presents the best fitting distributions with their parameter values. The Fit All function found the beta distribution to be the best distribution in 12 out of 15 cases. For two cases, the delta AIC values are small; so the beta distribution is still a candidate model for this sample. And for one case, the delta AIC value is not small which means the beta distribution may not be a proper model for this sample. On the other hand, the Arena Input Analyzer could find the beta distribution as the best fitting distribution only for 3 times. EasyFit found the four-parameter beta distribution as the best fitting distribution for 6 times.

Table 7.3. Best fitting distributions for the samples generated from the beta distribution.

BEST FITTING DISTRIBUTION				
Sample	**The Fit All Function**	**Delta AIC**	**The Arena Input Analyzer**	**EasyFit**
Beta(2,8) n=100	Beta(1.78,7.40)	0	Gamma(2.14,0.09)	-0.005 + Gamma(2.32,0.08)
	0.01 + 0.53* Beta(0.8,1.91)	8.6	Gamma(1.95,0.09)	Weibull(1.50,0.19)
	Beta(1.78,7.25)	0	Erlang(2,0.01)	0.005 + Weibull(1.54,0.21)
	Beta(2.12,8.89)	0	Weibull(1.78,0.22)	0.01 + 0.58* Beta(1.52,3.34)
	Beta(1.8,7.06)	0	Beta(1.46,3.04)	0.01 + 0.61* Beta(1.34,2.81)
Beta(2,8) n=250	Beta(2.3,8.83)	0	Beta(1.96,4.86)	0.004 + 0.91* Beta(2.15,7.46)
	Beta(2.06,8.31)	0	Gamma(2.53,0.08)	-0.009 + Gamma(2.87,0.07)
	Weibull(1.62,0.22)	0.2	Weibull(1.63,0.22)	Weibull(1.6,0.22)
	Beta(2.04,8.38)	0	2 + 24 * Beta(2.93, 8.82)	0.01 + 0.65 * Beta(1.54, 3.81)
	Beta(2.4,9.71)	0	Lognormal(0.203, 0.144)	Gamma(3.12,0.06)
Beta(2,8) n=1000	Beta(1.93,7.57)	0	Gamma(2.05,6.89)	Weibull(1.67,0.23)
	Beta(1.94,7.88)	0	Weibull(1.63,0.22)	0.98 * Beta(1.94, 7.64)
	0.7*Beta(1.64,4.45)	2.7	Weibull(1.69,0.22)	0.005 + 0.74 * Beta(1.64, 4.46)
	Beta(2.13,8.36)	0	Weibull(1.76,0.22)	-0.01+Gamma(3.1,0.07)
	Beta(2.02,7.84)	0	Beta(1.77,4.69)	Weibull(1.76,0.23)

Lastly, we generate random samples from the Weibull distribution with shape parameter 3 and scale parameter 7. Table 7.4 presents the best fitting distributions and the corresponding parameter values. The Fit All function found the Weibull distribution to be the best distribution in 14 out of 15 cases. For the remaining one case, the delta AIC value is small; so the Weibull distribution is still a candidate model for presenting this sample. On the other hand, the Arena Input Analyzer could find the Weibull distribution as the best fitting distribution for only 3 times. EasyFit found the three-parameter and two-parameter Weibull distributions as the best distributions for 4 and 2 times respectively.

Table 7.4. Best fitting distributions for the samples generated from the Weibull distribution.

BEST FITTING DISTRIBUTION				
Sample	**The Fit All Function**	**Delta AIC**	**The Arena Input Analyzer**	**EasyFit**
Weibull(3, 7) n=100	Weibull(3.19, 6.73)	0	Normal(6.03, 2.11)	-1.35 + Weibull(3.97, 8.15)
	Weibull(3, 7)	0	Normal(6.26, 2.25)	Normal(6.26, 2.25)
	Weibull(2.88, 7.18)	0	1 + 12 * Beta(2.25, 2.75)	-0.1 + Weibull(2.93, 7.29)
	Weibull(2.91, 6.98)	0	Triangular(1, 4.64, 13)	-13.91+ Lognormal(2.99,0.12)
	Normal(6.27,2.32)	3.46	12 * Beta(2.94, 2.69)	-3.29 + 19.3*Beta(7.62, 6.99)
Weibull(3, 7) n=250	Weibull(2.83, 7)	0	Normal(6.24, 2.39)	-29.76 + Lognormal(3.58,0.07)
	Weibull(2.79, 7.08)	0	Normal(6.31, 2.45)	-38.61+ Lognormal(3.80,0.05)
	Weibull(2.98,6.89)	0	Normal(6.16, 2.25)	Normal(6.16, 2.25)
	Weibull(3.09,6.89)	0	1 + 11 * Beta(2.45, 2.77)	Weibull(2.95, 6.89)
	Weibull(2.81, 6.72)	0	Gamma(6.05,0.99)	0.39 + Weibull(2.61, 6.29)
Weibull(3, 7) n=1000	Weibull(2.85, 6.90)	0	Normal(6.15, 2.35)	Normal(6.15, 2.35)
	Weibull(3.01, 7.1)	0	Weibull(2.93,7.04)	-0.38 + 15.5 * Beta(4.37, 5.76)
	Weibull(3.07, 6.88)	0	Weibull(3.01,6.97)	Weibull(3.05, 6.89)
	Weibull(3.11, 6.98)	0	Normal(6.24, 2.2)	-15.09 + Gamma(93.48, 0.23)
	Weibull(3.05, 7.11)	0	Weibull(3.09, 7.01)	0.38 + Weibull(2.85,6.71)

To sum up, our results clearly show that the Fit All function is the most powerful tool to determine the correct best fitting model. Moreover, as the sample size increases, the MLEs calculated by the Fit All function generally approach to the true values. On the other hand, EasyFit is prone to selecting the flexible forms of distributions rather than the standard distributions (e.g: it selects the three-parameter Weibull instead of the two-parameter Weibull). The Arena Input Analyzer has a completely different mechanism for model selection and could not select the correct distribution in most cases. For example for the samples which were generated from the Weibull distribution, the Arena Input Analyzer often found the best fitting model to be the normal distribution and the Weibull distribution in rank two with respect to the minimum square error distance measure.

7.2. Experiments with Real Data

Here we make some experiments with real stock data. These data were obtained from http://finance.yahoo.com/. As we have to work with IID data, we first calculate the log returns. We present the results of the recommended distributions by the Fit All function, the Arena Input Analyzer and EasyFit for 5 stocks.

For the stock INTC, the Fit All function and EasyFit both recommend the logistic distribution with the same parameter values. However the P-values that are obtained from the Chi-Square test are quite different. This is due to the fact that EasyFit uses 9 and the Fit All function uses 22 class intervals for n=503 observations. On the other hand, the Arena Input Analyzer recommends the normal distribution, which is only in rank three for the Fit All function. Table 7.5 summarizes the most proper models and the results of the goodness of fit tests.

Table 7.5. Recommended input models for the stock INTC.

Stock Name: INTC	**The Fit All Function**	**The Arena Input Analyzer**	**EasyFit**
Recommended Model:	Logistic Distribution	Normal Distribution	Logistic Distribution
Parameters:	L($\mu = 0$, $\sigma = 0.01$)	N($\mu = 0$, $\sigma = 0.02$)	L($\mu = 0$, $\sigma = 0.01$)
Goodness of Fit Measures:	CS P-value=0.159	Square Error= 0.005 CS P-value=0.007 KS P-value>0.15	CS P-value=0.965 KS P-value=0.867

For the stock DELL, the Fit All function and EasyFit could not find a best fitting distribution. The results of the the Arena Input Analyzer indicate that although the normal distribution is selected as the best-fitting model, the square error is too high and the P-value is below 0.005, so this model may not be sensible. Therefore, we can conclude that all these three tools arrive at the same conclusion for this stock. Additionally, the Fit All function recommends the empirical model: resampling with noise which may be helpful for the user in this case. Table 7.6 summarizes the most proper models and the results of the goodness of fit tests.

Table 7.6. Recommended input models for the stock DELL.

Stock Name: DELL	**The Fit All Function**	**The Arena Input Analyzer**	**EasyFit**
Recommended Model:	Resampling with noise	Normal Distribution	There is No fit
Parameters:	bandwidth= 0.006	N($\mu = 0$, $\sigma = 0.55$)	-
Goodness of Fit Measures:	-	Square Error= 0.230 CS P-value<0.005 KS P-value>0.01	-

For the stock KMB all three tools make different recommendations. The normal distribution that is recommended by the Arena Input Analyzer is in rank seven and the logistic distribution that is recommended by Easyfit is in rank two of the Fit All function. Table 7.7 summarizes the most proper models and the results of the goodness of fit tests.

Table 7.7. Recommended input models for the stock KMB.

Stock Name: KMB	**The Fit All Function**	**The Arena Input Analyzer**	**EasyFit**
Recommended Model:	3P-Student's T-Distribution	Normal distribution	Logistic Distribution
Parameters:	T($\mu = 0$, $\sigma = 0.01$, $\nu = 5.09$)	N($\mu = 0$, $\sigma = 0.01$)	L($\mu = 0$, $\sigma = 0.01$)
Goodness of Fit Measures:	CS P-value=0.059	Square Error= 0.004 CS P-value=0.012 KS P-value>0.15	CS P-value=0.477 KS P-value=0,724

For the stock MO the Fit All function recommends the three-parameter Student's t-distribution. The normal and the logistic distributions are rejected at $\alpha = 0.02$ level. The reason why the Arena Input Analyzer and EasyFit find high P-values for these distribution is again related to the small number of class intervals that are used in the Chi-Square test. Table 7.8 summarizes the most proper models and the results of the goodness of fit tests.

Table 7.8. Recommended input models for the stock MO.

Stock Name: MO	**The Fit All Function**	**The Arena Input Analyzer**	**EasyFit**
Recommended Model:	3P-Student's T-Distribution	Normal Distribution	Logistic Distribution
Parameters:	T($\mu = 0$, $\sigma = 0.01$, $\nu = 4.45$)	N($\mu = 0$, $\sigma = 0.01$)	L($\mu = 0$, $\sigma = 0.01$)
Goodness of Fit Measure:	CS P-value=0.028	Square Error= 0.007 CS P-value<0.005 KS P-value =0.019	CS P-value=0.193 KS P-value=0.09

For the stock VOD, the Fit All function and EasyFit both recommend the logistic distribution with the same parameter values. Only the Arena Input Analyzer

recommends the four-parameter beta distribution with respect to the minimum square error measure. However it is clearly seen that the P-value that is obtained from the Chi-Square test is below 0.005 which indicates that it is not a proper model. The Fit All function and EasyFit strongly reject this model. Table 7.9 summarizes the most proper models and the results of the goodness of fit tests.

Table 7.9. Recommended input models for the stock VOD.

Stock Name: VOD	**The Fit All Function**	**The Arena Input Analyzer**	**EasyFit**
Recommended Model:	Logistic Distribution	Beta Distribution	Logistic Distribution
Parameters:	L($\mu = 0$, $\sigma = 0.01$)	B(($\alpha_1 = 22.2$, $\alpha_2 = 18.5$) and ($\gamma_1 = -0.11$, $\gamma_2 = 0.32$))	L($\mu = 0$, $\sigma = 0.01$)
Goodness of Fit Measures:	CS P-value=0.272	Square Error= 0.002 CS P-value<0.005 KS P-value >0.15	CS P-value=0.44 KS P-value=0,279

The above results show that for these stock data, if there is an acceptable fit for the Fit All function, the recommended distribution is either the logistic or the three-parameter Student's t-distribution. EasyFit does not include the three-parameter version of Student's t-distribution and selects the logistic distribution as the most proper model. The Arena Input Analyzer includes neither the logistic nor the three-parameter Student's t-distribution and it generally selects the normal distribution.

To sum up, although the distribution pool of EasyFit and the Fit All function are different and the recommended models change for this reason, we can infer that they have similar results. For example if Easyfit finds the logistic distribution as the best fitting model, then the Fit All function either selects this model as the best fit or ranks it second. However the model selection of the Arena Input Analyzer is quite different. As it is described in the above examples, the Arena Input Analyzer is likely to find a a best fitting model which is strongly rejected by the Fit All function and/or EasyFit.

8. CONCLUSION

First and foremost our new input modeling tool which is called the Fit All function accomplishes the main aim of input modeling: It recommends the most proper model for the given data. As the results of the simulation study show, the Fit All function identifies the correct distribution more successfully than the Arena Input Analyzer or EasyFit.

The Fit All function includes many distributions that are both in standard (beta, Student's t) and flexible form (e.g: four-parameter beta, three-parameter Student's t). It also includes the negative versions of the flexible forms of some standard distributions which is an attractive feature when work with negatively skewed data.

For an input modeling software, it is important to obtain high-quality parameter estimates. The Arena Input Analyzer and EasyFit use different parameter estimation methods for different distributions (e.g: the MLE, the method of moments, the least square method etc.) to obtain faster results. However, the Fit All function uses the MLE for all distributions as it attaches more importance to obtaining good estimates. The results of the simulation study demonstrate that the parameter estimates which are presented by the Fit All function are closer to the original parameters.

The model selection criterion is also a vital part of input modeling. The Fit All function utilizes AIC which is a popular and theoretically sound criterion for model selection. As AIC is based on calculating the log likelihood of the fitted distribution, it is consistent with the parameter estimation method used. AIC penalizes distributions with more parameters, and thus helps to avoid over-fitting. The results of the simulation study show that AIC performs much better than the other model selection procedures.

Finally, when no proper distribution is found, then the Fit All function recom-

mends a simple empirical distribution model which is called resampling with noise. Resampling with noise is much better in approximating the density of the unknown distribution than empirical CDF approximations. Thus the Fit All function includes a new and practically important feature that the Arena Input Analyzer and EasyFit do not have.

APPENDIX A: R CODES

R code for generating random variates from kernel density of the data is given below.

```
n <- length(x)                    #sample size
s <- sd(x)                        #standard deviation
R <- IQR(x)                       #interquartile range
b <- 1.06*min(s,R/1.34)*n^(-1/5) #bandwidth
Y <- x[ceiling(runif(10000)*n)]+b*rnorm(10000)
```

R code for finding the MLEs of the two forms of beta distribution is given below.

```
# BETA DISTRIBUTION
# 4P-Beta Distribution/LOGLIKELIHOOD FUNCTION
lbeta4 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=gamma_1, theta[2]=gamma_2
# theta[3]=alpha_1 and theta[4]=alpha_2
n <- length(x)
res <- (-n*lbeta((theta[3]*(theta[3]>0)),(theta[4] *(theta[4]>0))))
+(theta[3]-1)*sum(log((x-theta[1])*(x>=theta[1])))
+(theta[4]-1)*sum(log((theta[2]-x)*(theta[2]>=x)))
-n*(theta[3]+theta[4]-1)*log((theta[2]-theta[1])*(theta[2]>=theta[1]))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}
# 2P-Beta Distribution/LOGLIKELIHOOD FUNCTION
lbeta2 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=alpha_1 and theta[2]=alpha_2
n <-length(x)
```

```
res <- (-n*lbeta(theta[1]*(theta[1]>0),theta[2]*(theta[2]>0)))
+(theta[1]-1)*sum(log((x)*(x>=0)*(x<=1)))
+(theta[2]-1)*sum(log((1-x)*(x>=0)*(x<=1)))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}

# Beta Distribution/OPTIMIZATION FUNCTION
MLEbeta <- function(x,npar){
  # MLEs for 4P-Beta distribution:
  if (npar==4)  {
  # STARTING VALUES for 4P-Beta distribution:
  # Sample size:
  n <- length(x)

  # Boundary parameters; gamma_1 and gamma_2:
  if (min(x)<0)  gamma_1 <- min(x)*(n+1)/(n)
  else           gamma_1 <- min(x)*(n)/(n+1)
  if (max(x)<0)  gamma_2 <- max(x)*(n)/(n+1)
  else           gamma_2 <- max(x)*(n+1)/(n)

 # Mean and variance are computed for 4P-beta distribution:
  a <- (mean(x)-gamma_1)/(gamma_2-gamma_1) # Mean
  v <- var(x)/(gamma_2-gamma_1)^2          # Variance

  # Shape parameters; alpha_1 and alpha_2:
  alpha_1 <- a*((a*(1-a)/v)-1)
  alpha_2 <- (1-a)*((a*(1-a)/v)-1)

  # We gather all starting values in the same vector:
  starting.beta <- c(gamma_1,gamma_2,alpha_1,alpha_2)
```

```
  # NUMERICAL SOLUTION for 4P-Beta distribution:
  res <- optim(starting.beta,lbeta4,x=x,addsign=T)
  r <- c(-2*(-res$value)+2*4,res$par)
  names(r) <- c("AIC","lower bound","upper bound","shape1","shape2")
  return (r)
  }
  # MLEs for 2P-Beta distribution:
  if (npar==2) {
  # distribution support for 2P-Beta distribution:
  if (min(x)<0|max(x)>1) return (NA)

  # STARTING VALUES for 2P-Beta distribution:
  # Mean and variance for 2P-beta distribution:
  a <- mean(x)      # Mean
  v <- var(x)       # Variance

  # Shape parameters; alpha_1 and alpha_2:
  alpha_1 <- a*((a*(1-a)/v)-1)
  alpha_2 <- (1-a)*((a*(1-a)/v)-1)

  # We gather all starting values in the same vector:
  starting.beta <- c(alpha_1,alpha_2)

  # NUMERICAL SOLUTION for 2P-Beta distribution
  res <- optim(starting.beta,lbeta2,x=x,addsign=T)
  r <- c(-2*(-res$value)+2*2,res$par)
  names(r) <- c("AIC","shape1","shape2")
  return (r)
  }
}
```

R code for finding the MLEs of the two forms of uniform distribution is given below.

```
# UNIFORM DISTRIBUTION
# 2P-Uniform Distribution/LOGLIKELIHOOD FUNCTION
luniform2 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=a, theta[2]=b
n <- length(x)
res <- -n*(log((theta[2]-theta[1])*(theta[2]>=theta[1])))
+sum(log(x>=theta[1]))+sum(log(x<=theta[2]))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}
# 1P-Uniform Distribution/LOGLIKELIHOOD FUNCTION
luniform1 <- function(theta,x,logyn=T,addsign=F){
# theta=b
n <- length(x)
res <- -n*(log(theta))+sum(log(x>=0))+sum(log(x<=theta))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}

# Uniform Distribution/MLE FUNCTION
MLEuniform <- function(x,npar){
  # MLEs for 2P-Uniform distribution:
  if (npar==2){
  # ANALYTICAL SOLUTION for 2P-Uniform distribution:
  # MLEs of boundary parameters; a and b:
  a_hat <- min(x)
  b_hat <- max(x)
```

```
  # LOG-LIKELHOOD VALUE FOR THE MLEs:
  res <- luniform2(c(a_hat,b_hat),x)
  r <- c(-2*(res)+2*2,a_hat,b_hat)
  names(r) <- c("AIC","lower bound","upper bound")
  return (r)
  }
  # MLE for 1P-Uniform distribution:
  if (npar==1){
  # distribution support for 2P-Uniform distribution:
  if (min(x)<0)  return (NA)

  # ANALYTICAL SOLUTION for 2P-Uniform distribution:
  # MLE of boundary parameter, b:
  b_hat <- max(x)

  # LOG-LIKELHOOD VALUE FOR THE MLE:
  res <- luniform1(b_hat,x)
  r <- c(-2*(res)+2*1,b_hat)
  names(r)<-c("AIC","upper bound")
  return (r)
  }
}
```

R code for finding the MLEs of the triangular distribution is given below.

```
# TRIANGULAR DISTRIBUTION
# Triangular Distribution/LOGLIKELIHOOD FUNCTION
ltriangular <- function(theta,x,logyn=T,addsign=F){
n <- length(x)
res <- sum((((log(2*(x-theta[1])*(x>=theta[1])))
-log((theta[2]-theta[1])*(theta[2]>theta[1]))
-log((theta[3]-theta[1])*(theta[3]>=theta[1])))
*((x>=theta[1])*(x<=theta[3])==1))
```

```
+(((log(2*(theta[2]-x)*(x<=theta[2])))-log((theta[2]-theta[1])
*(theta[2]>theta[1]))-log((theta[2]-theta[3])
*(theta[3]<=theta[2])))*((x>theta[3])*(x<=theta[2])==1)))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}

# Triangular Distribution/OPTIMIZATION FUNCTION
MLEtriangular <- function(x){
  # STARTING VALUES for Triangular distribution:
  # Sample size:
  n <- length(x)

  # Boundary parameters; a and b:
  if  (min(x)<0)  a <- min(x)*(n+1)/(n)
  else            a <- min(x)*(n)/(n+1)
  if  (max(x)<0)  b <- max(x)*(n)/(n+1)
  else            b <- max(x)*(n+1)/n

  # Shape parameter (mode value); c:
  Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
  }
  c <- Mode(x)

  # We gather all starting values in the same vector:
  starting.triangular <- c(a,b,c)

  # NUMERICAL SOLUTION for Triangular distribution:
  res <- optim(starting.triangular,ltriangular,x=x,addsign=T)
  r <- c(-2*(-res$value)+2*3,res$par)
```

```
  names(r) <- c("AIC","lower bound","upper bound","mode")
  return (r)
}
```

R code for finding the MLEs of the two forms of gamma distribution is given below.

```
# GAMMA DISTRIBUTION
# 3P-Gamma Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lgamma3 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=alpha, theta[2]=gamma
n <- length(x)
res <- -n*(theta[1]*(theta[1]>0))*log(sum((x-theta[2])
*(x>=theta[2])))+n*(theta[1]*(theta[1]>0))
*log((n*theta[1]*(theta[1]>0)))+(theta[1]*(theta[1]>0)-1)
*sum(log((x-theta[2])*(x>=theta[2])))-(n*theta[1]*(theta[1]>0))
-n*lgamma(theta[1]*(theta[1]>0))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}
# 2P-Gamma Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lgamma2 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=alpha
n <- length(x)
res <- -n*(theta*(theta>0))*log(sum((x)*(x>=0)))
+n*(theta*(theta>0))*log((n*theta*(theta>0)))
+(theta*(theta>0)-1)*sum(log((x)*(x>=0)))
-(n*theta*(theta>0))-n*lgamma(theta*(theta>0))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}
```

```
# Gamma Distribution/OPTIMIZATION FUNCTION
MLEgamma<-function(x,npar){
  # MLEs for 3P-Gamma distribution:
  if (npar==3) {
  # STARTING VALUES for 3P-Gamma distribution:
  # Sample size:
  n <- length(x)

  # Location parameter, gamma:
  if (min(x)<0)  gamma <- min(x)*(n+1)/(n)
  else           gamma <- min(x)*(n)/(n+1)

  # Shape parameter, alpha:
  xs <- x-gamma
  alpha <- (mean(xs)/sd(xs))^2

  # We gather all starting values in the same vector:
  starting.gamma <- c(alpha,gamma)

  # NUMERICAL SOLUTION for 3P-Gamma distribution:
  res <- optim(starting.gamma,lgamma3,x=x,addsign=T)

  # Scale parameter, beta:
  beta_hat <- sum(x-res$par[2])/(n*res$par[1])
  r <- c(-2*(-res$value)+2*3,c(res$par,beta_hat))
  names(r)<-c("AIC","shape","location","scale")
  return (r)
  }
  # MLEs for 2P-Gamma distribution:
  if (npar==2){
  # distribution support for 2P-Gamma distribution:
  if(min(x)<0)     return(NA)
```

```
  # STARTING VALUE for 2P-Gamma distribution:
  # Shape parameter, alpha:
  alpha <- (mean(x)/sd(x))^2
  starting.gamma <- alpha

  # NUMERICAL SOLUTION for 2P-Gamma distribution:
  res <- optim(starting.gamma,lgamma2,method="BFGS",x=x,addsign=T)

  # Scale parameter, beta:
  n <- length(x)
  beta <- sum(x)/(n*res$par)
  r <- c(-2*(-res$value)+2*2,c(res$par,beta))
  names(r) <- c("AIC","shape","scale")
  return (r)
  }
}
```

R code for finding the MLEs of the Erlang distribution is given below.

```
#Erlang Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lerlang2 <- function(theta,x,logyn=T,addsign=F){
#theta=alpha
n <- length(x)
res<- -n*as.integer(theta*(theta>0))*log(sum((x)*(x>=0)))
+n*as.integer(theta*(theta>0))*log((n*as.integer(theta*(theta>0))))
+as.integer(theta*(theta>0)-1)*sum(log((x)*(x>=0)))
-(n*as.integer(theta*(theta>0)))-n*lgamma(as.integer(theta*(theta>0)))
if (!logyn)  res <- exp(res)
if (addsign) res <- -res
res
}
```

```
# Erlang Distribution/OPTIMIZATION FUNCTION
MLEerlang<-function(x){
  # MLEs for Erlang distribution:
  # distribution support for 2P-Erlang distribution:
  if(min(x)<0)     return(NA)

  # STARTING VALUE for 2P-Erlang distribution:
  # Shape parameter, alpha:
  k <- MLEgamma(x,npar=3)[2]
  if (min(x)<0) {alpha <- NA}
  else if (k<1) {alpha <- 1}
  else          {alpha <- round(k)}
  starting.erlang <- alpha

  # NUMERICAL SOLUTION for 2P-Erlang distribution:
  res <- optim(starting.erlang,lerlang2,method="BFGS",x=x,addsign=T)

  # Scale parameter, beta:
  n <- length(x)
  beta <- sum(x)/(n*res$par)

  r <- c(-2*(-res$value)+2*2,c(res$par,beta))
  names(r) <- c("AIC","shape","scale")
  return (r)
 }
```

R code for finding the MLEs of the two forms of Weibull distribution is given below.

```
# WEIBULL DISTRIBUTION
# 3P-Weibull Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lweibull3 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=alpha, theta[2]=gamma
```

```
n <- length(x)
res <- n*log(theta[1]*(theta[1]>0))-n*theta[1]
*log(((sum((x-theta[2])^(theta[1])))/n)^(1/theta[1]))
+(theta[1]-1)*sum(log((x-theta[2])*(x>=theta[2])))
-sum(((x-theta[2])*(x>=theta[2])
/((sum((x-theta[2])^(theta[1])))/n)
^(1/theta[1]))^theta[1])
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}
# 2P-Weibull Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lweibull2 <- function(theta,x,logyn=T,addsign=F){
# theta=alpha
n <- length(x)
res <- n*log(theta*(theta>0))-n*theta*log(((sum(x^(theta)))/n)
^(1/theta))+(theta-1)*sum(log((x)*(x>=0)))-sum(((x)*(x>=0)
/((sum(x^(theta)))/n)^(1/theta))^theta)
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Weibull Distribution/OPTIMIZATION FUNCTION
MLEweibull <- function(x,npar){
  # MLEs for 3P-Weibull distribution:
  if (npar==3){
  # STARTING VALUES for 3P-Weibull distribution:
  # Location parameter, gamma:
  if (min(x)<0)  gamma <- min(x)+0.1*min(x)
  else           gamma <- min(x)-0.1*min(x)
```

```
# Shape parameter, alpha:
xs <- x-gamma
alpha <- mean(xs)/sd(xs)

# We gather all starting values in the same vector:
starting.weibull <- c(alpha,gamma)

# NUMERICAL SOLUTION for 3P-Weibull distribution:
res <- optim(starting.weibull,lweibull3,x=x,addsign=T)
n <- length(x)

# Scale parameter, beta:
beta_hat <- ((sum((x-res$par[2])^(res$par[1])))/n)^(1/res$par[1])
r <- c(-2*(-res$value)+2*3,c(res$par,beta_hat))
names(r) <- c("AIC","shape","location","scale")
return (r)
}
# MLEs for 2P-Weibull distribution:
if (npar==2){
# distribution support for 2P-Weibull distribution:
if(min(x)<0)     return(NA)

# STARTING VALUE for 2P-Weibull distribution:
# Shape parameter=alpha:
alpha <- mean(x)/sd(x)
starting.weibull <- alpha

# NUMERICAL SOLUTION for 2P-Weibull distribution:
res <- optim(starting.weibull,lweibull2,method="BFGS",x=x,addsign=T)

# Scale parameter, beta:
n <- length(x)
beta_hat <- ((sum(x^(res$par)))/n)^(1/res$par)
```

```
  r <- c(-2*(-res$value)+2*2,c(res$par,beta_hat))
  names(r) <- c("AIC","shape","scale")
  return (r)
  }
}
```

R code for finding the MLEs of the two forms of exponential distribution is given below.

```
# EXPONENTIAL DISTRIBUTION
# 2P-Exponential Distribution/LOGLIKELIHOOD FUNCTION
lexponential2 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=beta, theta[2]=gamma
n <- length(x)
res <- n*log(theta[1]*(theta[1]>0))-((theta[1])*(theta[1]>0)
*sum((x-theta[2])*(x>=theta[2])))+sum(log(x>=theta[2]))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}
# 1P-Exponential Distribution/LOGLIKELIHOOD FUNCTION
lexponential1 <- function(theta,x,logyn=T,addsign=F){
# theta=beta
n <- length(x)
res <- n*log(theta*(theta>0))-((theta)
*(theta>0)*sum((x)*(x>=0)))+sum(log(x>=0))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Exponential Distribution/MLE FUNCTION
MLEexponential <- function(x,npar){
```

```
  # MLEs for 2P-Exponential distribution:
  if(npar==2){

  # ANALYTICAL SOLUTION for 2P-Exponential distribution:
  # MLEs of location and rate parameters:
    # location parameter, gamma:
    gamma_hat <- min(x)

    # rate parameter, beta:
    beta_hat <- 1/mean(x-gamma_hat)

  # LOG-LIKELHOOD VALUE FOR THE MLEs:
   res <- lexponential2(c(beta_hat,gamma_hat),x)
   r <- c(-2*(res)+2*2,beta_hat,gamma_hat)
   names(r) <- c("AIC","rate","location")
   return (r)
   }
  # MLEs for 1P-Exponential distribution:
   if(npar==1){
  # distribution support for 1P-Exponential distribution:
   if (min(x)<0)  return (NA)

  # MLEs of rate parameter, beta:
   beta_hat <- 1/mean(x)

  # LOG-LIKELHOOD VALUE FOR THE MLE:
   res <- lexponential1(beta_hat,x)
   r <- c(-2*(res)+2*1,beta_hat)
   names(r)<-c("AIC","rate")
   return (r)
   }
}
```

R code for finding the MLEs of the two normal distribution is given below.

```
# Normal Distribution/LOGLIKELIHOOD FUNCTION
lnormal <- function(theta,x,logyn=T,addsign=F){
# theta[1]=mu,theta[2]=sigma
n <- length(x)
res <- -n*log(sqrt(2*pi))-n*log(theta[2]*(theta[2]>0))
-sum((((x)-theta[1])^2)/(2*(theta[2]*(theta[2]>0))^2))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Normal Distribution/MLE FUNCTION
MLEnormal <- function(x){
  # MLEs for Normal distribution:
  # ANALYTICAL SOLUTION for Normal distribution:
  # MLEs of location and scale parameters:

    # Location parameter, mu:
    n <- length(x)
    mu_hat <- mean(x)

    # Scale parameter, sigma:
    sigma_hat <- sqrt((1/n)*sum((x-mu_hat)^2))

  # LOG-LIKELHOOD VALUE FOR THE MLEs:
    res <- lnormal(c(mu_hat,sigma_hat),x)
    r <- c(-2*(res)+2*2,mu_hat,sigma_hat)
    names(r) <- c("AIC","location","scale")
    return (r)
}
```

R code for finding the MLEs of the two forms of Student's t-distribution is given below.

```
# STUDENT'S T-DISTRIBUTION
# 3P-Student's T Distribution/LOGLIKELIHOOD FUNCTION
lt3 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=mu,theta[2]=sigma,theta[3]=nu(degrees of freedom)
n <- length(x)
res <- n*lgamma((theta[3]*(theta[3]>0)+1)/2)
-n*lgamma(theta[3]*(theta[3]>0)/2)-n*log(theta[2]*(theta[2]>0))
-n*(1/2)*log(pi*theta[3]*(theta[3]>0))-((theta[3]*(theta[3]>0)+1)/2)
*sum(log(1+(((x-theta[1])/theta[2])^2)/(theta[3]*(theta[3]>0))))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}
# 1P-Student's T Distribution/LOGLIKELIHOOD FUNCTION
lt1 <- function(theta,x,logyn=T,addsign=F){
# theta=nu(degrees of freedom)
# theta=nu
n <- length(x)
res <- (-n/2)*log(theta*(theta>0))
-n*lbeta((1/2),((theta*(theta>0))/2))
-((theta*(theta>0)+1)/2)*sum(log(1+((x)^2)/theta*(theta>0)))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Student's t-distribution/OPTIMIZATION FUNCTION
MLEt <- function(x,npar){
  # MLEs for 3P-Student's T distribution:
  if (npar==3){
```

```
  # STARTING VALUES for 3P-Student's T Distribution:
  # Location parameter, gamma:
  gamma <- mean(x)
  # Scale parameter, sigma:
  sigma <- 0.01
  # Degrees of freedom, nu:
  nu <- 0.01

  # We gather all starting values in the same vector:
  starting.st <- c(gamma,sigma,nu)

  # NUMERICAL SOLUTION for 3P-Student's T Distribution:
  res <- optim(starting.st,lt3,x=x,addsign=T)
  r <- c(-2*(-res$value)+2*3, res$par)
  names(r) <- c("AIC","location","scale","degrees of freedom")
  return (r)
  }
  # MLEs for 1P-Student's T distribution:
  if (npar==1){
  # STARTING VALUE for 1P-Student's T Distribution:
  # Degrees of freedom, nu:
  nu_lower <- 0.01
  nu_upper <- 120

  starting.st <- c(nu_lower,nu_upper)

  # NUMERICAL SOLUTION for 1P-Student's T Distribution:
  res <- optimize(lt1, x=x, starting.st, maximum=TRUE)
  r <- c(-2*(res$objective)+2*1, res$maximum)
  names(r) <- c("AIC","degrees of freedom")
  return (r)
  }
}
```

R code for finding the MLEs of the two forms of lognormal distribution is given below.

```
# 3P-Lognormal Distribution/PROFILE LOGLIKELIHOOD FUNCTION
llognormal3 <- function(theta,x,logyn=T,addsign=F){
# theta=gamma
n <- length(x)
res <- -(n*(log(sqrt(2*pi))))-(n*log(sqrt(sum((log((x-theta)
*(x>theta))-sum(log((x-theta)*(x>theta))/n))^2)/n)))
-sum(log((x-theta)*(x>theta)))-sum((((log(((x-theta))
*(x>theta)))-sum(log((x-theta)*(x>theta))/n))^2)
/(2*(sqrt(sum((log((x-theta)*(x>theta))
-sum(log((x-theta)*(x>theta))/n))^2)/n))^2))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}
# 2P-Lognormal Distribution/LOGLIKELIHOOD FUNCTION
llognormal2 <- function(theta,x,logyn=T,addsign=F){
# theta[1]=mu and theta[2]=sigma
n <- length(x)
res<- -(n*(log(sqrt(2*pi))))-(n*log(theta[2]*(theta[2]>0)))
-sum(log(x*(x>0)))-sum((((log((x)*(x>0)))-theta[1])^2)
/(2*(theta[2]*(theta[2]>0))^2))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Lognormal Distribution/OPTIMIZATION FUNCTION
MLElognormal <- function(x,npar){
  # MLEs for 3P-Lognormal distribution:
  if(npar==3){
```

```
# STARTING VALUES for 3P-Lognormal distribution:
n <- length(x)
# Location parameter, gamma:
if (min(x)<0)   gamma <- min(x)*(n+1)/(n)
else            gamma <- min(x)*(n)/(n+1)
gamma_upper <- gamma
gamma_lower <- -100000 #lower limit for gamma which is a dummy number
starting.lognormal <- c(gamma_lower,gamma_upper)

# NUMERICAL SOLUTION for 3P-Lognormal distribution:
res <- optimize(llognormal3,x=x,starting.lognormal,addsign=T)
gamma_hat <- res$minimum

# Scale parameter, mu:
mu_hat <- sum(log((x-gamma_hat))/n)

# Shape parameter, sigma:
sigma_hat <- sqrt(sum((log(x-gamma_hat)-mu_hat)^2)/n)

r <- c(-2*(-res$objective)+2*3,mu_hat,sigma_hat,gamma_hat)
names(r) <- c("AIC","scale","shape","location")
return (r)
}
# MLEs for 2P-Lognormal distribution:
if(npar==2){
# distribution support for 2P-Lognormal distribution:
if (min(x)<=0)    return (NA)

# ANALYTICAL SOLUTION for 2P-Lognormal distribution
# MLEs of scale and shape parameters:
  # Scale parameter, mu:
  n <- length(x)
  mu_hat <- sum(log(x)/n)
```

```
    # Shape parameter, sigma:
    sigma_hat <- sqrt(sum((log(x)-mu_hat)^2)/n)

  # LOG-LIKELHOOD VALUE FOR THE MLEs:
  res <- llognormal2(c(mu_hat,sigma_hat),x)
  r <- c(-2*(res)+2*2,mu_hat,sigma_hat)
  names(r) <- c("AIC","scale","shape")
  return (r)
  }
}
```

R code for finding the MLEs of the logistic distribution is given below.

```
# LOGISTIC DISTRIBUTION
# Logistic Distribution/LOGLIKELIHOOD FUNCTION
llogistic <- function(theta,x,logyn=T,addsign=F){
# theta[1]=mu and theta[2]=alpha
n <- length(x)
res <- (-sum(x-theta[1])/(theta[2]*(theta[2]>0)))
-n*log((theta[2]*(theta[2]>0)))
-2*sum(log(1+exp(-(x-theta[1])/(theta[2]*(theta[2]>0)))))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Logistic Distribution/OPTIMIZATION FUNCTION
MLElogistic <- function(x){
  # MLEs for Logistic distribution:
  # STARTING VALUES for Logistic distribution:
  n <- length(x)
```

```
  # Location parameter, mu:
  mu <- mean(x)
  # Scale parameter, alpha:
  alpha <- (sqrt(3)/pi)*((sum(x^2)/n)-mu^2)^(1/2)

  # We gather all starting values in the same vector:
  starting.logistic <- c(mu,alpha)

  # NUMERICAL SOLUTION for Logistic distribution:
  # if(!is.finite(llogistic(starting.logistic,x))) return(NA)

  res <- optim(starting.logistic,llogistic,x=x,addsign=T)
  r <- c(-2*(-res$value)+2*2, res$par)
  names(r) <- c("AIC","location","scale")
  return (r)
}
```

R code for finding the MLEs of the two forms of Rayleigh distribution is given below.

```
# RAYLEIGH DISTRIBUTION
# 2P-Rayleigh Distribution/PROFILE LOGLIKELIHOOD FUNCTION
lrayleigh2 <- function(theta,x,logyn=T,addsign=F){
# theta=mu
n <- length(x)
res <- sum(log((x-theta)*(x>=theta)))
-2*n*log(sqrt((sum(((x-theta)*(x>=theta))^2))/(2*n)))
-(sum(((x-theta)*(x>=theta))^2)
/(2*(sqrt((sum(((x-theta)*(x>=theta))^2))/(2*n)))^2))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}
```

```
# 1P-Rayleigh Distribution/LOGLIKELIHOOD FUNCTION
lrayleigh1 <- function(theta,x,logyn=T,addsign=F){
# theta=sigma
n <- length(x)
res <- sum(log(x*(x>=0)))
-2*n*log((theta)*(theta>0))
-(sum((x*(x>=0))^2)/(2*(theta*(theta>0))^2))
if (!logyn) res <- exp(res)
if (addsign) res <- -res
res
}

# Rayleigh Distribution/OPTIMIZATION FUNCTION
MLErayleigh <- function(x,npar){
  # MLEs for 2P-Rayleigh distribution:
  if (npar==2){
  # STARTING VALUES for 2P-Rayleigh distribution:
  n <- length(x)
  # Location parameter, gamma:
  if (min(x)<0) gamma <- min(x)*(n+1)/(n)
  else          gamma <- min(x)*(n)/(n+1)
  gamma_upper <- gamma
  gamma_lower <- -100000 #lower limit for gamma which is a dummy number
  starting.rayleigh <- c(gamma_lower,gamma_upper)

  # NUMERICAL SOLUTION for 2P-Rayleigh distribution:
  res <- optimize(lrayleigh2,x=x,starting.rayleigh,addsign=T)
  gamma_hat <- res$minimum
  # Scale parameter, sigma:
  sigma_hat <- sqrt((sum((x-gamma_hat)^2))/(2*n))

  r <- c(-2*(-res$objective)+2*2,gamma_hat,sigma_hat)
  names(r) <- c("AIC","location","scale")
```

```
  return (r)
  }
  # MLE for 1P-Rayleigh distribution:
  if (npar==1){

  # distribution support for 1P-Rayleigh distribution:
  if (min(x)<0) return (NA)

  # ANALYTICAL SOLUTION for 1P-Rayleigh distribution:
  n <- length(x)
  # Scale parameter, sigma:
  sigma_hat <- sqrt((sum(x^2))/(2*n))



  # LOG-LIKELHOOD VALUE FOR THE MLE:
  res <- (lrayleigh1(sigma_hat,x))
  r <- c(-2*(res)+2*1,sigma_hat)
  names(r) <- c("AIC","scale")
  return (r)
  }
}
```

R code for executing the Chi-Square test for the normal distribution is below.

```
CHI-SQUARE TEST for NORMAL DISTRIBUTION
cs.normal <- function(x){
# WE BIN THE DATA:
# Number of class intervals:
  k <- floor(sqrt(length(x)))

# Find observed frequencies:
  unif.normal <- pnorm(x,mean=MLEnormal(x)[2],sd=MLEnormal(x)[3])
  obs.normal <- table(floor(unif.normal*k))
```

```
# Compute Chi-Square Test statistic
  b <- length(obs.normal)
  ct.normal <-(chisq.test(obs.normal, p = rep(1/b,b)))$statistic

# Compute Chi-Square P-value:
  # s=2, degrees of freedom=b-2-1
  P.normal <- 1-pchisq(ct.normal,(b-2-1))

# Sample size:
  n <- length(x)
# The level of significance for the Chi-Square test (a):
   if   (n<=20000)          a <- 1000/n       #%
   else                     a <- 0.05         #%

# Chi-Square Test decision:
  if (P.normal>(a*0.01))  CS.Decision <-"accept"
  else                    CS.Decision <-"reject"

  result <- data.frame(P.normal,CS.Decision)
  result
  }
}
```

R code for generation of the Fit All function is below.

```
fitall<-function(x){
### PARAMETRIC MODEL
#  Find values of AIC and MLEs
   m <- alldistributions(x)
#  Compute the Chi-Square  P-values and show whether each model
passes or fails the Chi-Square Test:
   cs <- cs.test(x)
```

```
   CS.P.value  <- cs[,3]
   CS.Decision <- cs[,5]

#  Gather values of AIC, MLEs, the Chi-Square Test and names of parameters
 in the same list:
   k <- data.frame(m,CS.P.value,CS.Decision,parnames)

#  Order results in ascending order with respect to AIC values:
   ms <- k[order(k[,1]),]

#  Do not show "NA"s.
   e <- ms[!is.na(ms[,1]),]

#  Gather values of AIC, MLEs, the Chi-Square Test and
Delta AIC in the same list:
   f<-e[,1:7]

   #  Calculate delta AIC to see how different the other models are from
the the best-fitted model:
   Delta.AIC <- f[,1]-f[1,1]
   ALL <- cbind(f,Delta.AIC)

### NON-PARAMETRIC MODEL: RESAMPLING with NOISE
    n <- length(x)                   #sample size
    s <- sd(x)                       #standard deviation
    R <- IQR(x)                      #interquartile range
    b <- 1.06*min(s,R/1.34)*n^(-1/5) #bandwidth
    Y <- x[ceiling(runif(10000)*n)]+b*rnorm(10000)

# If the best fitting distribution passes the Chi-Square Test, then
recommend it by showing AIC values and parameters:
     if (ALL[1,7]=="accept") {
     distribution.name <- rownames(ALL[])[1]
```

```
     # Display AIC value in 3 digits:
     AIC<-round(ALL[1,1],3)

    # parameters:
     par<-ALL[1,2:5]
     # Display parameters in 3 digits by omitting "NA"s:
     parameters<-round(par[!is.na(par[1,1:4])],3)
     names<-e[1,8:11]
     par.names<-names[!is.na(names[1,1:4])]
     full.definition.parameters<-paste(par.names,parameters,
sep="=",collapse=",")
     recommend<-paste("It is recommended to use the", distribution.name,
"distribution with the parameters:",full.definition.parameters)
     res<-list("model"=recommend,"AIC"=AIC,"parameters"=parameters,
"summary"=ALL)
     res
     }
# Else, recommend resampling with noise by displaying the value of
bandwidth:
     else {
     # Display the value of bandwidth in 3 digits
     bandwidth<-round(b,3)
     recommend<-paste("It is recommended to use resampling with noise",
 "with bandwidth value of", bandwidth)
     res<-list("model"=recommend,"bandwidth"=bandwidth,"summary"=ALL)
     res
     }
return (res)
 }
fitall(x)
```

REFERENCES

1. Biller, B. and B. L. Nelson, "Answers to Top Ten Input Modeling Questions, In Winter Simulation Conference", *WSC '02: Proceedings of the 34th Conference on Winter Simulation*, IEEE Computer Society, pp. 35-40, San Diego, 2002.

2. Banks, J., J. S. Carson, B. L. Nelson and D. M. Nicol, *Discrete Event System Simulation*, 3rd edition, Prentice Hall, New Jersey, 2000.

3. Biller, B. and C. Gunes, "Introduction to Simulation Input Modeling", In Winter Simulation Conference, *WSC '10: Proceedings of the 42nd Conference on Winter Simulation*, IEEE Computer Society, pp. 49-58, Baltimore, 2010.

4. Chung, C. A., *Simulation Modeling Handbook: A Practical Approach*, CRC Press, Texas, 2003.

5. Law, A. M. and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, New York, 1991.

6. Devroye, L., *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.

7. Hörmann, W. and O. Bayar, "Modeling Probability Distributions from Data and Its Influence on Simulation", *IMACS Journal*, No.15, pp. 429-435, 2000.

8. Zucchini, W., *Applied Smoothing Techniques: Kernel Density Estimation*, 2003, `http://isc.temple.edu/economics/Econ616/Kernel/ast_part1.pdf`, accessed at June 2012.

9. Altıok, T. and B. Melamed, *Simulation Modeling and Analysis with Arena*, Academic Press, Burlington, 2007.

10. Mathwave, *What Parameter Estimation Methods Are Used in EasyFit?*, `http://mathwave.com/blog/2011/10/28/what-parameter-estimation-methods-are-used-in-easyfit`, accessed at June 2012.

11. Wikipedia, *Overfitting*, `http://en.wikipedia.org/wiki/Over_fitting`, accessed at June 2012.

12. Hörmann, W., "Statistical Inference", 2010, *Lecture Notes for Statistical Inference Course*, Boğaziçi University, İstanbul.

13. King, G., *Unifying Political Methodology: The Likelihood Theory of Statistical Inference*, Cambridge University Press, New York, 1998.

14. Myung, I. J., "Tutorial on Maximum Likelihood Estimation", *Journal of Mathematical Psychology*, Vol. 47, pp. 90-100, 2003.

15. Wolfram, *Numerical Nonlinear Global Optimization*, `http://reference.wolfram.com/mathematica/tutorial/ConstrainedOptimizationGlobalNumerical.html`, accessed at June 2012.

16. Cousineau, D., S. Brown and A. Heathcote, "Fitting Distributions Using Maximum Likelihood: Methods and Packages", *Behavior Research Methods, Instruments and Computers*, Vol. 36, pp. 742-756, 2004.

17. Bolker, B., "Optimization and All That", *Princeton University Press*, Vol. 1, pp. 1-49, 2007.

18. Maitra, R., "Concentrated or Profile Log-Likelihood Function", 2006, *Lecture Notes for Statistical Computing Course*, Iowa State University, Ames.

19. Balakrishnan, N. and J. Wang, "Simple Efficient Estimation for the Three-Parameter Gamma Distribution", *Journal of Statistical Planning and Inference*, Vol. 85, pp. 115-126, 2010.

20. Cohen, C. A. and B. Whitten, "Modified maximum likelihood and modified moment estimators for the three-parameter weibull distribution", *Communications in Statistics-Theory and Methods*, Vol. 11, pp. 2631-2656, 1982.

21. Cohen, C. A. and B. Whitten, "Estimation in the Three-Parameter Lognormal Distribution", *Journal of the American Statistical Association*, Vol. 75, No. 370, pp. 399-404, 1980.

22. Moitra, S. D., "Skewness and the Beta Distribution", *Journal of the Operational Research Society*, Vol. 41, No. 10, pp. 953-961, 1990.

23. Wikipedia, *Beta Distribution*, `http://en.wikipedia.org/wiki/Beta_distribution`, accessed at June 2012.

24. Viele, K., *MLEs and Their Asymptotics*, 2008, `http://web.as.uky.edu/statistics/users/viele/sta601s08/mles.pdf`, accessed at June 2012.

25. Owen, C. B., *Parameter Estimation for Beta Distribution*, Diploma thesis, Brigham Young University, 2008.

26. Wang, J. Z., "A Note on Estimation in the Four-Parameter Beta Distribution", *Communications in Statistics-Simulation and Computation*, Vol. 34, pp. 495-501, 2005.

27. Carnahana, J. V., "Maximum Likelihood Estimation for the Four-Parameter Beta Distribution", *Communications in Statistics-Simulation and Computation*, Vol. 18, pp. 513-536, 1989.

28. Engineering Handbook, *Beta Distribution*, `http://www.itl.nist.gov/div898/handbook/eda/section3/eda366h.htm`, accessed at June 2012.

29. Smithson, M. and J. Verkuilen, "A Better Lemon Squeezer? Maximum-Likelihood

Regression with Beta-Distributed Dependent Variables", *Psychological Methods*, Vol. 11, No. 1, pp. 54-71, 2006.

30. Cheng, R. C. H. and T. C. Iles, "Corrected Maximum Likelihood in Nonregular Estimation", *Journal of the Royal Statistical Society*, Vol. 49, pp. 95-101, 1987.

31. Evans, M., N. Hastings and B. Peacock, *Statistical Distributions*, 3rd edition, John Willey and Sons, New York, 2000.

32. Panchenko, D., "Maximum Likelihood Estimators", 2006, *Lecture Notes on Statistics for Applications*, Massachusetts Institute of Technology, Massachusetts.

33. Dorp, J. R. V., "Maximum Likelihood Estimation for Uniform Distribution", 2011, *Lecture Notes on Discrete System Simulation*, The George Washington University, Washington DC.

34. Mittelhammer, R. C., *Mathematical Statistics for Economics and Business*, Springer-Verlag, New York, 1996.

35. Dorp, J. R. V., and S. Kotz, "A Novel Extension of the Triangular Distribution and Its Parameter Estimation", *The Statistician*, Vol. 51, pp. 63-79, Washington DC, 2002.

36. Engineering Handbook, *Gamma Distribution*, `http://www.itl.nist.gov/div898/handbook/apr/section1/apr165.htm`, accessed at June 2012.

37. Giles, D. E. and H. Feng, "Bias of the Maximum Likelihood Estimators of the Two-Parameter Gamma Distribution Revisited", *Econometrics Working Papers*, No. 908, University of Victoria, Canada, 2009.

38. Tzavelas, G., "Estimation in the Three-Parameter Gamma Distribution Based on the Profile Log-Likelihood Function", *Communications in Statistics-Simulation*

and Computation, Vol. 38, pp. 573-583, 2009.

39. Wiens, D. P., J. Cheng, and N. C. Beaulieu, "A Class of Method of Moments Estimators for the Two-Parameter Gamma Family", *Pakistan Journal of Statistics*, Vol.19, pp. 129-141, 2003.

40. Relawiki, *Weibull Distribution*, `http://www.weibull.com/LifeDataWeb/characteristics_of_the_weibull_distribution.htm`, accessed at June 2012.

41. Balakrishnan, N., M. Kateri, "On the Maximum Likelihood Estimation of Parameters of Weibull Distribution Based on Complete and Censored Data, *Statistics and Probability Letters*, Vol. 78, pp. 2971-2975, 2008.

42. Lockhart, R. A., and M. A. Stephens, "Estimation and Tests of Fit for the Three-Parameter Weibull Distribution", *Journal of the Royal Statistical Society*, Vol. 56, No. 3, pp. 491-500, 1994.

43. Hanan, M. A., "The Exponential Distribution", 2010, *Lecture Notes on Reliability Course*, King Saud University, Riyadh.

44. Rashid, M. Z. and A. S. Akhter, "Estimation Accuracy of Exponential Distribution Parameters", *Pakistan Journal of Statistics and Operation Research*, Vol. 7, No. 2, pp. 217-232, 2011.

45. Wikipedia, *Normal Distribution*, `http://en.wikipedia.org/wiki/Normal_distribution`, accessed at June 2012.

46. Wikipedia, *Student's t-Distribution*, `http://en.wikipedia.org/wiki/Student's_t-distribution`, accessed at June 2012.

47. Gatz, J., *Properties and Applications of the Student T Copula*, Master thesis, Delft University of Technology, 2007.

48. Walpole, M., *Probability and Statistics for Engineers and Scientists*, 8th edition, Prentice Hall, New Jersey, 2007.

49. Divgi, D. R., "Robust Estimation Using Student's t Distribution", *Center for Naval Analyses*, pp. 90-217, 1990.

50. Goorbergh, R., and P. Vlaar, "Value-at-Risk Analysis of Stock Returns Historical Simulation, Variance Techniques or Tail Index Estimation", *Netherlands Central Bank*, 1999.

51. Jackman, S., *Bayesian Analysis for the Social Sciences*, Wiley-Blackwells, New Jersey, 2009.

52. Chen, C., "Tests of Fit for the Three-Parameter Lognormal Distribution", *Computational Statistics and Data Analysis*, Vol. 50, pp. 1418-1440, 2006.

53. Wikipedia, *Lognormal Distribution*, `http://en.wikipedia.org/wiki/Lognormal_distribution`, accessed at June 2012.

54. Gupta, R. D. and D. Kundu, "Generalized Logistic Distributions", *Journal of Applied Statistical Sciences*, Vol. 18, no. 1, pp. 51-66, 2010.

55. Mahdi, S. and M. Cenac, "Estimating and Assessing the Parameters of the Logistic and Rayleigh Distributions from Three Methods of Estimation", *Caribbean Journal of Mathematical and Computing Sciences*, Vol. 13, pp. 25-34, 2006.

56. Adatiaa, A., "Approximate Blue of the Scale Parameter of the Rayleigh Distribution Based on Doubly Censored Samples, *Journal of Statistical Computation and Simulation*, Vol. 56, pp. 251-271, 1997.

57. Best, D. J., J. C. W. Rayner, and O. Thas, "Easily Applied Tests of Fit for the Rayleigh Distribution", *Indian Statistical Institute*, Vol. 72, pp. 254-263, 2010.

58. Mahdi, S., "Improved Parameter Estimation in Rayleigh Model", *Advances in Methodology and Statistics*, Vol. 3, No. 1, pp. 63-74, 2006.

59. Engineering Handbook, *Kolmogorov-Smirnov Goodness-of-Fit Test*, http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm, accessed at June 2012.

60. Mathwave, *Anderson-Darling Test*, http://www.mathwave.com/help/easyfit/html/analyses/goodness_of_ fit/anderson-darling.html, accessed at June 2012.

61. Damodaran, A., *Statistical Distributions*, http://pages.stern.nyu.edu/adamodar/New_ Home_Page/StatFile/statdistns.htm, accessed at June 2012.