Comparison of methods to calculate measures of inequality based on interval data

by

Willem Francois Neethling

Thesis presented in partial fulfilment

of the requirements for the degree of Master of Statistics

in the Faculty of Economic and Management Sciences

at Stellenbosch University.

Supervisor: Prof Tertius de Wet

Co-Supervisor: Dr Ariane Neethling

December 2015

Plagiarism declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date:...................................

Abstract

In recent decades, economists and sociologists have taken an increasing interest in the study of income attainment and income inequality. Many of these studies have used census data, but social surveys have also increasingly been utilised as sources for these analyses. In these surveys, respondents' incomes are most often not measured in true amounts, but in categories of which the last category is open-ended. The reason is that income is seen as sensitive data and/or is sometimes difficult to reveal.

Continuous data divided into categories is often more difficult to work with than ungrouped data. In this study, we compare different methods to convert grouped data to data where each observation has a specific value or point. For some methods, all the observations in an interval receive the same value; an example is the midpoint method, where all the observations in an interval are assigned the midpoint. Other methods include random methods, where each observation receives a random point between the lower and upper bound of the interval. For some methods, random and non-random, a distribution is fitted to the data and a value is calculated according to the distribution.

The non-random methods that we use are the midpoint-, Pareto means- and lognormal means methods; the random methods are the random midpoint-, random Pareto- and random lognormal methods. Since our focus falls on income data, which usually follows a heavy-tailed distribution, we use the Pareto and lognormal distributions in our methods.

The above-mentioned methods are applied to simulated and real datasets. The raw values of these datasets are known, and are categorised into intervals. These methods are then applied to the interval data to reconvert the interval data to point data. To test the effectiveness of these methods, we calculate some measures of inequality. The measures considered are the Gini coefficient, quintile share ratio (QSR), the Theil measure and the Atkinson measure. The estimated measures of inequality, calculated from each dataset obtained through these methods, are then compared to the true measures of inequality.

Opsomming

Oor die afgelope dekades het ekonome en sosioloë ŉ toenemende belangstelling getoon in studies aangaande inkomsteverkryging en inkomste-ongelykheid. Baie van die studies maak gebruik van sensus data, maar die gebruik van sosiale opnames as bronne vir die ontledings het ook merkbaar toegeneem. In die opnames word die inkomste van ŉ persoon meestal in kategorieë aangedui waar die laaste interval oop is, in plaas van numeriese waardes. Die rede vir die kategorieë is dat inkomste data as sensitief beskou word en soms is dit ook moeilik om aan te dui.

Kontinue data wat in kategorieë opgedeel is, is meeste van die tyd moeiliker om mee te werk as ongegroepeerde data. In dié studie word verskeie metodes vergelyk om gegroepeerde data om te skakel na data waar elke waarneming ŉ numeriese waarde het. Vir van die metodes word dieselfde waarde aan al die waarnemings in ŉ interval gegee, byvoorbeeld die ‘midpoint’ metode waar elke waarde die middelpunt van die interval verkry. Ander metodes is ewekansige metodes waar elke waarneming ŉ ewekansige waarde kry tussen die onder- en bogrens van die interval. Vir sommige van die metodes, ewekansig en nie-ewekansig, word ŉ verdeling oor die data gepas en ŉ waarde bereken volgens die verdeling.

Die nie-ewekansige metodes wat gebruik word, is die ‘midpoint’, ‘Pareto means’ en ‘Lognormal means’ en die ewekansige metodes is die ‘random midpoint’, ‘random Pareto’ en ‘random lognormal’. Ons fokus is op inkomste data, wat gewoonlik ŉ swaar stertverdeling volg, en om hierdie rede maak ons gebruik van die Pareto en lognormaal verdelings in ons metodes.

Al die metodes word toegepas op gesimuleerde en werklike datastelle. Die rou waardes van die datastelle is bekend en word in intervalle gekategoriseer. Die metodes word dan op die interval data toegepas om dit terug te skakel na data waar elke waarneming ŉ numeriese waardes het. Om die doeltreffendheid van die metodes te toets word ŉ paar maatstawwe van ongelykheid bereken. Die maatstawwe sluit in die Gini koeffisiënt, ‘quintile share ratio’ (QSR), die Theil en Atkinson maatstawwe. Die beraamde maatstawwe van ongelykheid, wat bereken is vanaf die datastelle verkry deur die metodes, word dan vergelyk met die ware maatstawwe van ongelykheid.

Acknowledgements

I would like to express my sincere appreciation to the following people:

- My study leader, Prof T de Wet, for his guidance and assistance throughout the study and also for his encouragement.
- My father and mother, for the opportunity to study and also for their encouragement and interest in my studies.

Table of contents

List of figures

List of tables

List of abbreviations and/or acronyms

A_{ε}	Atkinson measure
EVI	Extreme Value Index
exp	Exponent
I_G	Gini coefficient
f	Density function of the data
F	Cumulative distribution function of the data
GE	Generalized Entropy
MAD	Median Absolute Deviation
n	The sample size
N	The population size in case of a finite population
QSR	Quintile Share Ratio
R	Number of replications
θ	The population parameter (true value)
$\hat{\theta}$	The estimator of the parameter θ
ϕ	Standard normal density function
Φ	Standard normal cumulative distribution function

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In recent decades, economists and sociologists have taken an increasing interest in the study of income attainment and income inequality. In this regard, several articles have been published that focus on individual income as a phenomenon to be explained. Many of these studies have used census data, but increasingly, social surveys have also been used as sources for these analyses (West, 1986; Yu, 2013; Malherbe, 2007).

In these surveys, respondents' income are most often not measured in exact amounts, but in categories, of which the last category is open-ended. The reason is that income is seen as sensitive data and/or is sometimes difficult to reveal. In some cases, individuals are not willing to disclose their exact income when undertaking a survey, or may not be in a position to provide an exact amount, as their income varies from month to month. This can result in non-responses in a survey. One way in which non-responses may be reduced is to make use of intervals. Respondents may feel more comfortable indicating an interval into which their income falls, rather than providing an exact amount; the use of intervals also makes it easier for individuals whose income varies on a monthly basis to provide useable data.

However, the use of survey data grouped in categories (with the last category being open-ended), may present an important measurement problem (West, 1986). The problem with this type of categorical measurement is especially acute when the researcher intends to estimate income through the application of statistical techniques such as regression; the problem also occurs when estimating income-based quantities such as inequality measures, which assume specific measurements.

Before continuing, let us conceptualise the following: data can be defined as any set of information where each observation describes a given entry. Data may be represented as either grouped or ungrouped data. Ungrouped data is raw data, where each observation has a specific value. Grouped data is data that has been divided into groups, also known as classes. Each class has a certain width (called the class interval) and consists of a lower and upper bound. The widths of the intervals may either be the same, or they may differ.

Grouped data is most often represented in frequency tables. This means that the lower and upper bound of each interval are given, with only the frequency of the observations in that interval known. The exact value of an observation is thus not known for grouped data. This data is difficult to work with; not all calculations can be carried out on such data, and those which can be carried out are more complicated than when working with data where each observation has a specific value.

In this study, we compare different methods to convert grouped data to data where each observation has a specific value or point, called point data. For some methods, all the observations in each interval are given the same value (the midpoint or mean); for the random methods, each value in an interval is assigned a random value between the lower and upper bound of the interval. For certain methods we also fit a distribution to the data and determine a value for each observation according to this distribution. This value is either a conditional mean or a random point according to the distribution, where the random point is between the lower and upper bound.

In this study, we make use of six different methods. For the midpoint-, Pareto means- and lognormal means methods, the same calculated value is assigned to all the observations in a specific interval. For the random midpoint-, random Pareto- and random lognormal methods, a random value is assigned to each observation between the lower and upper bound of the interval. In these cases all the observations in an interval will not have the same value (Yu, 2013; Von Fintel, 2006).

For the Pareto means method, a Pareto distribution is fitted over the data, and a conditional mean, according to the Pareto distribution, is calculated between the lower and upper bound of the interval and assigned to each observation in the interval. Likewise, for the lognormal means method, a lognormal distribution is fitted over the data, and a conditional mean is calculated according to the lognormal distribution between the lower and upper bound of the interval. For the random Pareto- and random lognormal methods, a Pareto and lognormal distribution is also fitted to the data, but a random value is assigned to each observation according to the distribution, between the lower and upper bound.

All the methods are tested on simulated and real datasets. The simulated datasets are obtained by simulating from different Pareto, lognormal and Burr distributions. For the real data we make use of the Income and Expenditure Survey (IES) 2005/2006 data. For the simulated data, the distribution simulated from (as well as its parameters) is known, and we can therefore determine the true value of the measure and compare the measures of each method to the true value. For the real data we only have the raw data. Therefore, the measures obtained with each method are compared to the measure obtained with the raw data.

After the observations are categorised into intervals, and each method is used to convert the grouped data to point or continuous data, some measures of inequality are estimated for each method. These estimated measures of inequality for each method are then compared to the true values, in order to determine the effectiveness of each method. The measures of inequality that are used in this study are the Gini coefficient, quintile share ratio (QSR), Theil measure and Atkinson measure (Haughton and Khandker, 2009; Atkinson, 1970). The QSR, the least known of these measures, is defined as the ratio of the total income received by 20% of a country's population with the highest income to the total income received by 20% of a country's population with the lowest income.

1.2 PURPOSE OF THE STUDY

The simplest method to convert grouped data to point data is to make use of the midpoint method. For this method the midpoint of each interval is assigned to each of the observations in the interval. The midpoint method is a method that is easy to use and understand, and no statistical or mathematical background is necessary. For this reason it is a method commonly used by researchers in several disciplines.

The purpose of this study is to compare the midpoint method to other methods such as the Pareto means-, lognormal means-, random midpoint-, random Pareto- and random lognormal methods. Specifically, we want to compare the effectiveness of the different methods in converting grouped data to point data.

To test the effectiveness of each method we make use of four measures of inequality. Three of these measures are well-documented in the literature and frequently applied in practice; they are the Gini coefficient, Theil- and Atkinson measures. The fourth measure is the so-called quintile share ratio (QSR). This is a lesser-known measure, but it is one of the two measures used in the European Union to measure inequality; the other is the Gini coefficient.

Each of the four inequality measures will be calculated for each dataset obtained from each method, as well as for the raw data. The measures obtained from each method will be compared to the true measure.

1.3 CHAPTER OUTLINE

This document consists of seven chapters.

Following this introductory chapter, Chapter Two begins with an overview of the methods used in previous South African studies. The methods used in this study are then presented in depth, and where necessary, formulas are derived to calculate a point or random value.

In Chapter Three, the four measures of inequality are studied. Some background information of each measure is presented, and formulas to calculate each measure for a finite and infinite population are given. The formulas to calculate each measure for each distribution that is simulated from are also derived.

In Chapter Four, the simulation process is presented. The parameters used for each distribution simulated from are also chosen. The 90th percentile, median, expected value and mode are calculated for each of these distributions.

Chapter Five focuses on the analysis of the results obtained from the simulated data. The formulas for the measures of performance used, as well as the root mean square error (RMSE), the median absolute deviation (MAD) and the standard errors are given.

In Chapter Six the IES 2005/2006 data is studied. Some background information about the dataset is presented, and the results that are obtained from this dataset are analysed and studied.

Chapter Seven provides summaries of the entire process and the main results obtained for the simulated and real data. Some recommendations and thoughts on further studies are also discussed.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

Research on income is increasingly based on data from social surveys. In these surveys, the respondents' income is often not accessible as an amount, but only available in grouped data format. Since the formulas of inequality measures generally rely on continuous data, there is a need to 'convert' grouped data to continuous or point data.

A variety of methods have been used in previous studies to convert grouped data to point data, as discussed below. In this study, the conversion will be based on the calculation of inequality measures when income data is only available in intervals. It is important to decide which methods are available, which of these methods are the best to use for such data, and which estimation methods have to be used to estimate parameters where necessary. This will be the focus of this chapter, as will the derivation of a general formula to obtain a mean for methods where a distribution is used to fit the data.

2.2 OVERVIEW OF METHODS AND THEIR IMPLEMENTATION IN PREVIOUS SOUTH AFRICAN STUDIES

In this section an overview is given of the methods that have been used in previous studies. Thereafter, the methods that are further used and applied in this study are explained in more detail, with mathematical derivations given in section 2.3.

In previous South African studies, Von Fintel (2006) considers the September 2003 Labour Force Survey data. He examines different methods to convert 'bad data' (data that consists of categorical and nominal data), to 'good data' (data that researchers are readily able to use for the purpose of the analysis of earnings data). Von Fintel considers the midpoint-, midpoint-Pareto- (called 'Pareto means' in this study), lognormal means- and interval regression methods.

For the midpoint method, the midpoint of each interval is calculated and each observation in the interval receives the midpoint as its point value. This is the simplest method, which may account for its frequent usage. For the Pareto means method, the parameters of the Pareto distribution are estimated by fitting a Pareto distribution to the interval data. The estimates are then used in a formula derived to obtain the conditional mean of the Pareto distribution between the lower and upper bounds of an interval. The lognormal means method is the same as the Pareto means method, except that the lognormal distribution is used instead of the Pareto distribution. Thus, the parameters of the lognormal distribution are estimated by fitting a lognormal distribution over the interval data; these estimates are used in a formula derived to obtain the conditional mean of the lognormal distribution between the lower and upper bound. Von Fintel (2006) uses the ordinary least square estimation method to estimate the parameters of the Pareto distribution, and the maximum likelihood estimation method to estimate the parameters for the lognormal distribution.

The interval regression method attempts to predict a specific point through a model fitted to a dataset that consists of interval data. The model is fitted to the data using some variables that explain the dependent variable. This model is then used to predict a specific figure (or amount) for the dependent variable, based on these well-chosen variables. The interval regression method is not considered in this study.

Malherbe (2007) focuses on the analysis of income data in South Africa by focusing on poverty and income distribution, and poverty and inequality measures. She uses the 2000 Income and Expenditure Survey (IES) data (where income is continuous), and creates a grouped income variable using the income intervals of Census 2001. In the study, the midpoint-, interval regression- and random midpoint methods are used to derive a point value for each observation in each interval.

The random midpoint method is a method used to create a continuous dataset from grouped data. It is a variation of the midpoint method. This method makes use of the midpoint of an income interval, and then distributes the observations within the interval across the interval in a random manner. The random midpoint is calculated by taking the midpoint, and randomly adding or subtracting a random uniform number of the difference between the midpoint and lower bound of an interval.

Malherbe finds that the poverty estimates for the continuous dataset and the midpoint method are very close to one another, while the results obtained from interval regression method and the random midpoint method are different. The interval regression method does not produce good results, and it seems to underestimate poverty. The model of interval regression inaccurately predicts a figure of 71.22%. This means that 71.22% of the predicted intervals differ from the original intervals. The results obtained with the random midpoint method were not useable and eventually omitted.

Yu (2013) uses data collected in household surveys conducted between 1993 and 2009. He examines various factors that affect the comparability and reliability of poverty estimates. Yu also studies the trends across household surveys. Some of the data that he uses is in interval form, while other data is in exact amounts. If the data is in interval form, Yu explores methods of converting the interval data to continuous data for the purpose of poverty analysis. He uses the abovementioned midpoint-, Pareto means-, interval regression-, random midpoint- and equal distribution methods to convert interval data into point data.

The equal distribution method distributes the observations equally within each interval. For example, if there are 500 observations within the interval R500 − R999, R500 will be assigned to the first observation, R501 to the second observation, R502 to the third observation and so on, until R999 is assigned to the 500th observation. Since income data is not uniformly distributed, this method will not be applied in this study.

Yu examines the effect of each method on poverty estimates. In his study, the Pareto means method was found to be the most appropriate to convert interval data to point data.

In some earlier studies, Hofmeyr (2001) examines data of the 1995 and 1999 October Household Surveys, and applies the midpoint of the interval as a specific point value for the interval. The study of Rospabé (2002) is based on data from the 1993 Project for Statistics on Living Standards and Development (PSLSD) and the 1999 October Household Survey. Rospabé uses interval regression (a generalisation of the Tobit model), as an estimation method.

2.3 DISCUSSION OF EXISTING METHODS USED IN THIS STUDY AND THEIR APPLICATION.

There are several different ways to assign a point value to data that consists of intervals. In this study the following methods are used to convert interval data to point data.

2.3.1 Midpoint method

The midpoint method is the simplest method, and is widely used among researchers because of the limited knowledge of statistics needed to implement this method. For each variable that consists of an interval, the midpoint of that interval is assumed as the value for each observation in that interval. For example, if the interval is [0, 100), each observation in that interval will take the value 50 as the point value. For the last open interval, Statistics South Africa (StatsSA) has used the lower bound times two as the specific value for that interval. For example, for the interval of 2 457 601 or more, the point value of 4 915 202 is assigned. Other studies, such as those of Yu (2013) and Von Fintel (2006), use the method of Fields (1989), and take the lower bound of the open interval and multiply it with 110%, thus assuming that the mean exceeds the lower bound by 10%. For example, if the lower bound of the open interval is 20 000, the midpoint is assumed to be 20 000×1.1=22 000. In this study, we will consider both the lower bound times two, and the lower bound plus 10%, for the last interval.

Seiver (1979) states that the true mean of an interval of any given length for income data, will most often be lower than the midpoint, given that the interval starts with a zero, as reported income will tend to heap at levels ending on zero. For example, if the income categories were [6 000, 7 999]; [8 000, 9 999], then people earning R8 000 would fall in the latter interval, while the former interval would be dominated by those earning R6 000. On the other hand if the interval were, for example, [6 001, 8 000]; [8 001, 10 000], it may be expected that the former interval would probably be dominated by people earning R8 000, and that the true mean in this case would exceed the midpoint of R7 000.

2.3.2 Distribution means methods

Usually the lower intervals for income data are narrow, and the width of the intervals increases for the high intervals. The distribution of income for the intervals at the bottom is not influenced by the midpoint method in a noticeable way, and because of the greater skewness within the intervals at the end, a parametric approach with a heavy-tailed distribution is necessary (Von Fintel, 2006).

Heavy-tailed distributions are distributions that have a larger probability of observing very large values. An example of a heavy-tailed distribution is if 80% of a country's wealth is owned by 20% of the people. A distribution that has a heavier tail than an exponential distribution is defined as a heavy-tailed distribution; i.e.

$$\lim_{x\to\infty} \frac{\exp(-\lambda x)}{\bar{F}(x)} = 0, \text{ for any } \lambda > 0,$$

where $\bar{F}(x) = 1 - F(x)$ (Kpanzou, 2011).

Some commonly used heavy-tailed distributions include the Pareto-, lognormal-, Weibull- and Burr distributions.

The distribution means method makes use of a distribution, and calculates the conditional mean of an interval from the distribution. Let a and b be the lower and upper bounds of an interval, $F_X(x)$ the cumulative distribution function for variable X, and $f_X(x)$ the corresponding density function. A general formula to calculate the conditional mean of a distribution between a and b is derived as follows:

Let T be the random variable defined as $X \mid a < X < b$, then $E(X|a < X < b) = E(T)$, and the cumulative distribution function can be written as

$$\begin{aligned} F_T(t) &= P(T \le t) \quad \text{where } a < t < b \\ &= P(X \le t \mid a < X < b) \\ &= P(a < X \le t \mid a < X < b) \\ &= \frac{P(a < X < t)}{P(a < X < b)} \\ &= \frac{F_X(t) - F_X(a)}{F_X(b) - F_X(a)}, \quad \text{where } a < t < b. \end{aligned}$$

The corresponding density function for the variable T can be written in terms of the density and distribution of X, as follows:

$$f_T(t) = \begin{cases} \dfrac{f_X(t)}{F_X(b) - F_X(a)} & \text{if } \; a < t < b \\ 0 & \text{otherwise} \end{cases}$$

thus,

$$E(X|a < X < b) = E(T) = \int_0^\infty t f_T(t) dt = \int_a^b \frac{t f_X(t)}{F_X(b) - F_X(a)} dt = \frac{1}{F_X(b) - F_X(a)} \int_a^b t f_X(t)\, dt.$$

The general formula to calculate the conditional mean of a distribution is thus:

$$E(X|a < X < b) = \frac{1}{F_X(b) - F_X(a)} \int_a^b t f_X(t)\, dt. \qquad (2.3.1)$$

Formula (2.3.1) can be used to calculate the conditional mean of any distribution. In section 2.3.2.1, it is assumed that the data or the tail of the data follows a Pareto distribution. Subsequently, in section 2.3.2.2, it is assumed that the data or tail of the data follows a lognormal distribution (Von Fintel, 2006; Whiteford & McGrath, 1994; Gustavsson, 2004).

2.3.2.1 Pareto means method

The first distribution considered for the distribution means method is the Pareto distribution. Vilfredo Pareto, who developed the Pareto distribution, was the first to consider the theoretical properties of the income distribution. Pareto intended to provide a justification for the properties of the right tail of the distribution that relates to the empirical income distribution (Dagsvik & Vatne, 1999). The Pareto mean can be used for the last open interval but can also be used for a selected number of intervals (Von Fintel, 2006).

The Pareto distribution has the following density and distribution function:

$$f(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}} \text{ for } x \geq k \text{ and } \alpha > 0,$$

and

$$F(x) = 1 - \left(\frac{k}{x}\right)^\alpha \text{ for } x \geq k \text{ and } \alpha > 0.$$

The following formula is used to calculate the mean of the Pareto distribution for closed intervals (intervals that have a lower and upper bound):

$$\bar{x} = \left[\frac{\hat{\alpha}}{1-\hat{\alpha}}\right]\left[\frac{x_1^{-\hat{\alpha}+1} - x_2^{-\hat{\alpha}+1}}{x_2^{-\hat{\alpha}} - x_1^{-\hat{\alpha}}}\right] \text{ for } \hat{\alpha} > 1, \tag{2.3.2}$$

where x_1 and x_2 are the upper and lower bounds of the interval and $\hat{\alpha}$ is the estimate of the Pareto coefficient. For the last open interval the following formula is used:

$$\bar{x} = \left[\frac{\hat{\alpha}}{\hat{\alpha}-1}\right] x_\infty \text{ for } \hat{\alpha} > 1, \tag{2.3.3}$$

where x_∞ represents the lower bound of the open interval.

We now prove formulas (2.3.2) and (2.3.3) by using the general formula of (2.3.1):

Since $X \sim Pareto(\alpha, k)$, the density and distribution functions are

$$f_X(x) = \begin{cases} \frac{\alpha k^\alpha}{x^{\alpha+1}} & \text{for } x \geq k \\ 0 & \text{for } x < k \end{cases}$$

and

$$F_X(x) = \begin{cases} 1 - \left(\frac{k}{x}\right)^\alpha & \text{for } x \geq k \\ 0 & \text{for } x < k \end{cases},$$

respectively.

It follows that

$$\begin{aligned}
E(X|a < X < b) &= \frac{1}{F_X(b) - F_X(a)} \int_a^b t f_X(t)\, dt \\
&= \frac{1}{1 - \left(\frac{k}{b}\right)^\alpha - \left(1 - \left(\frac{k}{a}\right)^\alpha\right)} \int_a^b t \frac{\alpha k^\alpha}{t^{\alpha+1}}\, dt \\
&= \frac{1}{\left(\frac{k}{a}\right)^\alpha - \left(\frac{k}{b}\right)^\alpha} \int_a^b t \frac{\alpha k^\alpha}{t^{\alpha+1}}\, dt \\
&= \frac{\alpha k^\alpha}{\left(\frac{k}{a}\right)^\alpha - \left(\frac{k}{b}\right)^\alpha} \int_a^b t^{-\alpha} dt \\
&= \frac{\alpha k^\alpha}{k^\alpha a^{-\alpha} - k^\alpha b^{-\alpha}} \left[\frac{1}{-\alpha+1} t^{-\alpha+1}\right]_a^b \\
&= \frac{\alpha k^\alpha}{k^\alpha (a^{-\alpha} - b^{-\alpha})} \left[\frac{1}{-\alpha+1} b^{-\alpha+1} - \frac{1}{-\alpha+1} a^{-\alpha+1}\right]
\end{aligned}$$

$$= \frac{\alpha}{a^{-\alpha} - b^{-\alpha}} \left[\frac{b^{-\alpha+1} - a^{-\alpha+1}}{-\alpha + 1} \right]$$

$$= \frac{\alpha}{1 - \alpha} \times \frac{b^{-\alpha+1} - a^{-\alpha+1}}{a^{-\alpha} - b^{-\alpha}} .$$

When α is replaced with its estimator, namely $\hat{\alpha}$, the following formula is obtained for the Pareto means method for a closed interval:

$$\frac{\hat{\alpha}}{1 - \hat{\alpha}} \times \frac{b^{-\hat{\alpha}+1} - a^{-\hat{\alpha}+1}}{a^{-\hat{\alpha}} - b^{-\hat{\alpha}}}. \tag{2.3.4}$$

When the last open interval is used, a will still be the lower bound of the open interval, but b, the upper bound, will tend to infinity. Formula (2.3.4) will change as follows:

$$b^{-\alpha+1} \to 0 \text{ and } b^{-\alpha} \to 0 \quad \text{if } \alpha > 1.$$

The following formula is then obtained:

$$\frac{\alpha}{1 - \alpha} \times \frac{0 - a^{-\alpha+1}}{a^{-\alpha} - 0}$$

$$= \frac{\alpha}{\alpha - 1} a.$$

When α is replaced with its estimator, namely $\hat{\alpha}$, the following formula is obtained for the Pareto means method for the open interval:

$$\frac{\hat{\alpha}}{\hat{\alpha} - 1} a . \tag{2.3.5}$$

The linear form derived from the distribution function, $F(x)$, will be used to estimate the parameters for the Pareto means method. The linear form can be written as

$$\ln(M) = d - \alpha \ln(x), \tag{2.3.6}$$

where

$d = \alpha \ln(k)$,

x = the lower bound of the interval,

M represents the number of entries above x, and

ln is the natural logarithm.

The ordinary least squares estimation method will be used to estimate a value for α in formula (2.3.6), and substituted into formulas (2.3.2) and (2.3.3) to calculate the Pareto mean.

In this study, the Pareto means method is applied in different ways. First, the parameter, α, of the Pareto distribution is estimated by making use of all the intervals of the grouped data. The estimate is then substituted into formulas (2.3.2) and (2.3.3) in order to obtain the Pareto means. Hereafter, the parameter is estimated on all the intervals but the first interval. This new estimate is then used to calculate the Pareto means again for all the intervals, excluding the first interval. The midpoint method is then applied on the first interval and the Pareto means on the remaining intervals. Then the parameter is estimated on all the intervals, now excluding the first two intervals. Then the midpoint method is applied to the first two intervals, while the Pareto means method is applied to the remaining intervals. This continues until there are only two intervals left. When there are only two intervals left, the parameter cannot be estimated because more than two values are needed for ordinary least squares estimation.

In studies such as those of Whiteford & McGrath (1994), Von Fintel (2006) and Yu (2013), the Pareto means method is applied in three different ways.

The first way is to determine the coefficient of determination for all intervals. Then the first interval is removed, and the coefficient of determination is calculated again. Intervals continue to be removed until the coefficient of determination is calculated on the last three intervals. The Pareto means method is then used on the number of intervals with the largest coefficient of determination, while the midpoint method is applied to the rest of the intervals. For example, if it is assumed that there are ten intervals, and intervals five to ten resulted in the largest coefficient of determination, the midpoint will then be assigned to the observations in intervals one to four, and Pareto mean to the observations in intervals five to ten.

The second way is to make use of the midpoint method up to and including the interval that contains the population median, while formulas (2.3.2) and (2.3.3) are used for the remaining intervals. Thus, if the median is contained in interval five, the observations in intervals one to five will be assigned the midpoint, while the remaining intervals will be assigned the Pareto mean.

The third way is to make use of the midpoint method to assign a specific value (i.e. the midpoint), to all the intervals with the exception of the last interval. The Pareto means method, formula (2.3.3), is only used on the last open interval.

The disadvantage of the third method is that one cannot estimate the parameters by only having one observation, i.e. the last interval. To estimate the parameters through least square estimation, one needs at least three observations, i.e. three intervals. Yu (2013) used the estimated parameter obtained in the second way (described in the above paragraph) as the estimated parameter for the third way, in which the Pareto means method is applied only to the last open interval. Although the Pareto mean is used only for the last interval, the Pareto parameter used to calculate the Pareto mean is estimated by using more intervals. For this reason, it was decided not to use this third way in this study.

2.3.2.2 Lognormal means method

The lognormal distribution is another (semi-) heavy-tailed distribution that is used in this study to determine a mean by fitting a model. A lognormal distribution is defined as a normal distribution fitted to the log of the data. Gustavsson (2004) obtains more accurate results overall with mean-approximation with the lognormal distribution, than with the Pareto means method.

The lognormal distribution has the following density and distribution function:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{\frac{-(\ln(x) - \mu)^2}{2\sigma^2}\right\}, \quad x > 0$$

$$F(x) = \Phi\left(\frac{\ln(x) - \mu}{\sigma}\right),$$

where Φ is the cumulative distribution function of the standard normal distribution.

The mean for the lognormal means method is calculated as follows (Von Fintel, 2006):

$$\bar{y}_{lognormal|a\le y\le b} = \hat{\mu} - \hat{\sigma}\frac{\phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) - \phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}{\Phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) - \Phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}, \qquad (2.3.7)$$

where

a is the lower bound of the category,

b is the upper bound of the category,

$\hat{\mu}$ is the estimator of the normal mean of the logged data,

$\hat{\sigma}$ is the estimator of the normal standard deviation of the logged data,

$\phi(x)$ is the standard normal density function.

Formula (2.3.7) can be used for both bounded intervals and the open interval at the end. For the last open interval $b \to \infty$ and (2.3.7) will simplify through $\phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) \to 0$ and $\Phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) \to 1$. Formula (2.3.7) is derived with the following calculations:

The general formula of the conditional mean of a distribution was derived at formula (2.3.1) above and is given as

$$\frac{1}{F_X(b) - F_X(a)} \int_a^b t f_X(t)\, dt.$$

With the general formula (2.3.1), the conditional mean of the lognormal distribution can be derived as follows:

$Y \equiv Earnings$

$X = \ln(Y) \sim N(\mu; \sigma^2)$

Thus, the conditional mean is calculated as

$$E(X|a < X < b) = \frac{1}{F_X(b) - F_X(a)} \int_a^b t f_X(t)\, dt$$

$$= \frac{1}{F_X(b) - F_X(a)} \int_a^b t \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] dt$$

Let $z = \dfrac{t-\mu}{\sigma} \Rightarrow t = z\sigma + \mu \Rightarrow dt = -\sigma dz$

$$= \frac{1}{\Phi(b^*) - \Phi(a^*)} \int_{a^*}^{b^*} (z\sigma + \mu) \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}z^2\right] dz$$

where $a^* = \dfrac{a-\mu}{\sigma}$ and $b^* = \dfrac{b-\mu}{\sigma}$

$$= \frac{1}{\Phi(b^*) - \Phi(a^*)} \left\{\mu \int_{a^*}^{b^*} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}z^2\right] dz + \sigma \int_{a^*}^{b^*} \frac{z}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}z^2\right] dz\right\}$$

Let $w = -\dfrac{1}{2}z^2 \Rightarrow dw = -zdz$

$$= \mu - \frac{1}{\Phi(b^*) - \Phi(a^*)} \{\sigma[\phi(b^*) - \phi(a^*)]\}$$

$$= \mu - \sigma \frac{\phi(b^*) - \phi(a^*)}{\Phi(b^*) - \Phi(a^*)}$$

$$= \mu - \sigma \frac{\phi\left(\frac{b-\mu}{\sigma}\right) - \phi\left(\frac{a-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}.$$

When μ and σ are replaced with their estimates, namely $\hat{\mu}$ and $\hat{\sigma}$, the following formula is obtained for the lognormal means method for a closed interval:

$$\hat{\mu} - \hat{\sigma} \frac{\phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) - \phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}{\Phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) - \Phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}. \tag{2.3.8}$$

When the last open interval is used, a will still be the lower bound of the open interval, but b, the upper bound, will tend to infinity. Formula (2.3.8) will change as follows:

$$\phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) \to 0 \text{ and } \Phi\left(\frac{b-\hat{\mu}}{\hat{\sigma}}\right) \to 1.$$

Then

$$\mu - \sigma \frac{0 - \phi\left(\frac{a-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{a-\mu}{\sigma}\right)}$$

$$= \mu - \sigma \frac{\phi\left(\frac{a-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{a-\mu}{\sigma}\right)}.$$

When μ and σ are replaced with their estimates, namely $\hat{\mu}$ and $\hat{\sigma}$, the following formula is obtained for the lognormal means method for the open interval:

$$\hat{\mu} - \hat{\sigma} \frac{\phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}{1 - \Phi\left(\frac{a-\hat{\mu}}{\hat{\sigma}}\right)}. \tag{2.3.9}$$

The estimates $\hat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} \ln(Y_i)$ and $\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\ln(Y_i) - \hat{\mu})^2}$, the parameters of the lognormal distribution, will be estimated by making use of maximum likelihood estimation. These estimates will then be substituted into formulas (2.3.8) and (2.3.9) to obtain a value in the interval, in order to apply the lognormal means method.

In this study, when making use of the lognormal means method, the midpoint method is used for the first interval, and the lognormal means method for the remaining intervals. If the midpoint method is not used for the first interval and the estimates for the lognormal means method are estimated over all the intervals, some of the lognormal means fall outside the interval bounds. The necessary parameters are therefore estimated on all the intervals except the first interval.

2.3.3 Random midpoint method

This method makes use of the midpoint of an income interval, and then distributes the observations within the interval across the interval in a random manner. The random midpoint is calculated in the following way:

Let f_i be the frequency of observations within the income interval i, and let x_i be the midpoint of interval i; then the following formula is applied to obtain the random midpoint dataset (Malherbe; 2007):

$$Y_{ij} = x_i + sign_{ij} U_{ij}$$

where

Y_{ij} is the new random midpoint income value for income level i and observation j, j = 1,2, ... , f_i,

x_i is the midpoint for interval i,

$sign_{ij}$ is the sign for interval i and observation j, where $sign_{ij}$ has a 50% chance of being +1 or -1,

$U_{ij} \sim Uniform\ (0; x_i - lowerbound)$, with $lowerbound$ the lower bound of interval i.

For example, if the interval is 500 − 999, the midpoint of the interval is 750. If $sign_{ij}$ is equal to +1 and if $Uniform(0; 750 - 500)$ = $Uniform(0; 250)$ equals 150, then Y_{ij} will be equal to 750 + $(+1)(150)$ = 900; if $sign_{ij}$ is equal to -1 with $Uniform(0{,}250)$ equal to 150, then Y_{ij} will be equal to 750 + $(-1)(150)$ = 600.

2.4 PROPOSED RANDOM DISTRIBUTION METHODS

In section 2.3.2, the distribution means methods were considered. For those methods, each observation in an interval was assigned the same value, which was the conditional mean of a distribution. The random midpoint method was considered in section 2.3.3, and is an adjustment of the general midpoint method. Instead of assigning the midpoint of the interval to each observation in the interval, each observation is assigned the midpoint plus or minus a random value. Each observation in an interval is not assigned the same value as for the midpoint method, but rather a random value. The random distribution methods are an adjustment of the distribution means methods. Instead of assigning a single conditional value to each observation in an interval according to a distribution, a random conditional value is assigned to each observation in an interval according to a distribution.

The random methods are used to examine whether there is a difference between the assignment of a single value to each observation in an interval and the assignment of a different/random value to each observation in an interval. The results of the random methods will be compared to the non-random methods, to observe whether the random methods are more accurate. The effect on the standard error of a random method and non-random method will also be examined.

The following general formula is used to calculate the random value between the lower and upper bound of an interval. Let a and b be the lower and the upper bounds of an interval and F_X the cumulative distribution function. Then using the probability integral transformation

$$U = F_{a,b}(Y) = \frac{F_X(Y) - F_X(a)}{F_X(b) - F_X(a)} \sim U(0,1)$$

$$F_X(Y) = F_X(a) + U\big(F_X(b) - F_X(a)\big).$$

When solving for Y, the following formula is obtained:

$$Y = F_X^{-1}\Big(F_X(a) + U\big(F_X(b) - F_X(a)\big)\Big). \qquad (2.4.1)$$

This formula can be used to calculate a random point between a and b for any distribution F_X. We apply it to the Pareto and lognormal distributions.

2.4.1 Random Pareto method

For the Pareto means method, a mean was calculated according to a Pareto distribution. Each observation in each interval was assigned the same value. For the random Pareto method, instead of assigning the same value to all the observations in an interval, a random value will be assigned to each observation between the lower and upper bound according to the Pareto distribution. The same estimated parameter obtained when estimation is carried out over all the intervals for the Pareto means method will be used for the random Pareto method.

The random point for the random Pareto method will be calculated as follows:

If $X \sim Pareto(\alpha, k)$ then $F_X(z) = 1 - \left(\frac{k}{z}\right)^{\alpha}$. Thus,

$$F_X^{-1}(u) = k(1-u)^{-\frac{1}{\alpha}},$$

and using formula (2.4.1)

$$Y = k\left(\left(\frac{k}{a}\right)^{\alpha} + U\left(\left(\frac{k}{a}\right)^{\alpha} - \left(\frac{k}{b}\right)^{\alpha}\right)\right)^{-\frac{1}{\alpha}}. \tag{2.4.2}$$

Formula (2.4.2) will be used to calculate the random point for each observation for the random Pareto method.

2.4.2 Random lognormal method

For the lognormal means method, a mean was calculated according to the lognormal distribution. The same value was assigned to each observation in an interval. For the random lognormal method, instead of assigning the same value to all the observations in an interval, a random value will be assigned to each observation between the lower and upper bound according to the lognormal distribution.

The random point for the random lognormal method will be calculated as follows:

If $X \sim lognormal(\mu, \sigma)$, then $F_X(z) = \Phi\left(\frac{\ln(z)-\mu}{\sigma}\right)$. Thus,

$$F_X^{-1}(u) = \exp\left(\mu + \sigma\Phi^{-1}(u)\right).$$

Substituting into formula (2.4.1) then gives

$$Y = \exp\left\{\mu + \sigma\Phi^{-1}\left(\Phi\left(\frac{\ln(a)-\mu}{\sigma}\right) + U\left(\Phi\left(\frac{\ln(b)-\mu}{\sigma}\right) - \Phi\left(\frac{\ln(a)-\mu}{\sigma}\right)\right)\right)\right\}. \tag{2.4.3}$$

Formula (2.4.3) will be used to calculate the random point for each observation for the random lognormal method.

2.5 SUMMARY

In this chapter, different existing and proposed methods to convert grouped income data to point data were considered. Firstly, in section 2.3, an overview of existing methods and their implementation in previous South African studies were given. Thereafter, the existing methods used in this study and their application were discussed. These methods are the midpoint method (the simplest method), the distribution means methods (the Pareto means method and

the lognormal means method, both of which make use of a distribution and calculate the conditional mean of an interval from the distribution), and the random midpoint method (which uses the midpoint of an income interval and then distributes the observations within the interval across the interval in a random manner).
In section 2.4, two proposed random distribution methods, the random Pareto- and random lognormal methods, were discussed. For these methods, a random conditional value is assigned to each observation in an interval according to a distribution.

All these methods will be compared in Chapters 5 and 6, based on their performance in the calculation of inequality measures.

CHAPTER 3

MEASURES OF INEQUALITY

3.1 INTRODUCTION

In this study, we focus on inequality and not on poverty. Inequality is in a sense a broader concept than poverty, as inequality is defined over the entire population, while poverty concerns only those individuals whose income falls under a certain poverty line, and who are subsequently considered poor. Inequality of income and wealth have been studied by various researchers. In order to measure inequality, a scale of inequality is necessary to evaluate it (Nishino & Kakamu, 2011). The simplest measure of inequality is to sort the population from poorest to richest, divide the population in fifths and report the proportion of people falling within each category. Various known inequality measures are available, including the Gini coefficient, the Theil measure and the Atkinson measure (Atkinson, 1970). Among all of them, the Gini coefficient is the most famous and well-known measure.

A desirable feature of inequality is mean independence, which implies that the measure does not depend on the mean of the distribution (Haughton & Khandker, 2009).

According to Haughton and Khandker, the criteria for a good measure of inequality are:

1. Mean independence:
   This implies that the measure would not change if all incomes were doubled.

2. Population size independence:
   If the population changes, the measure of inequality would not change.

3. Symmetry:
   The measure of inequality should not change if any two persons swap incomes.

4. Pigou-Dalton Transfer sensitivity:
   The transfer of income from rich to poor reduces the measured inequality.

5. Decomposability:
   Inequality can be broken down by population groups or some other dimensions.

6. Statistical testability:
   This implies that significance of changes of the index over time can be tested.

These criteria are considered when each of the measures of inequality, is studied in the following sections.

In this study, we will consider the following measures of inequality:

- Gini coefficient
- Quintile share ratio
- Theil measure
- Atkinson measure

In the next section, each of these measures is studied. The finite and infinite formulas to calculate the measures are given, and the formulas to calculate the measurement for the Pareto-, lognormal- and Burr distributions are derived.

Throughout this chapter, the information is obtained from Haughton and Khandker (2009), unless otherwise indicated. This reference indicates the Handbook on Poverty and Inequality of the World Bank, which is regarded as a reliable and authoritative source.

3.2 MEASURES OF INEQUALITY

3.2.1 Gini coefficient

The Gini coefficient is the most widely-used measure of inequality. It ranges from zero (perfect equality) to one (perfect inequality). Perfect equality means that the wealth of the population is uniformly distributed, while perfect inequality means that the wealth of the entire population belongs to a single person. The Gini coefficient is calculated from the Lorenz curve. The Lorenz curve sorts the population from poorest to richest, and indicates the cumulative proportion of the population on the x-axis and the cumulative proportion of income on the y-axis. An example of a Lorenz curve is given on the following page.

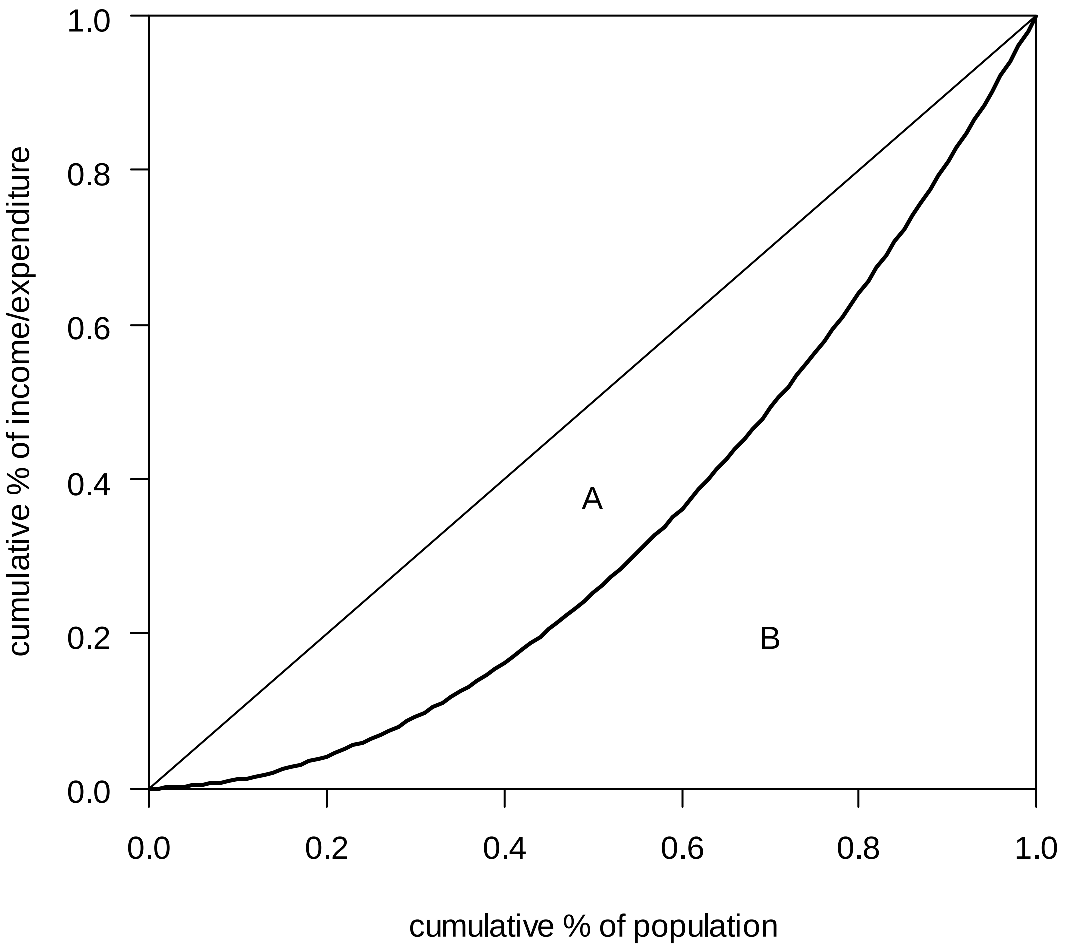


Figure 3.2.1: Lorenz curve

The diagonal line in the figure represents perfect equality. The Gini coefficient can then be calculated with the following formula:

$$I_G = \frac{A}{A+B} = 2A.$$

Perfect equality is obtained when $A = 0$ and the Gini coefficient becomes zero, while perfect inequality is obtained when $B = 0$ and Gini coefficient becomes one.

If x_i represents a point on the x-axis and y_i a point on the y-axis, the formal definition for the Gini coefficient is

$$I_G = 1 - \sum_{i=1}^{N} (x_i - x_{i-1})(y_i + y_{i-1}), \qquad (3.2.1)$$

over N points.

There are many alternative expressions for the Gini coefficient in the literature. The expression for the Gini coefficient that is used in this study is defined by Kpanzou (2011) as

$$I_G = \frac{1}{\mu}\int_0^\infty F(x)\big(1 - F(x)\big)\,dx, \tag{3.2.2}$$

where $F(x)$ is the cumulative distribution function and μ is the expected value of the function.

When the Gini coefficient is compared to the criteria for a good measure of inequality, it satisfies mean independence, population size independence, symmetry and Pigou-Dalton Transfer sensitivity, but does not satisfy decomposability and statistical testability.

The general formula to compute the Gini coefficient numerically for a distribution is derived as follows:

Let $u = F(x)$, then $x = F^{-1}(u) = Q(u)$ and $dx = q(u)du$. The formula for the Gini coefficient then becomes

$$\begin{aligned} I_G &= \frac{1}{\mu}\int_0^1 u(1-u)\,dQ(u) \\ &= \frac{1}{\mu}\int_0^1 u(1-u)\,q(u)du. \end{aligned}$$

Since $q(u) = \frac{1}{f(Q(u))}$,

$$I_G = \frac{1}{\mu}\int_0^1 u(1-u)\frac{1}{f\big(Q(u)\big)}du. \tag{3.2.3}$$

For the lognormal and Burr distributions, the Gini coefficient has to be calculated numerically from formula (3.2.3), by using the distribution formulas in the table below.

Table 3.2.1: Distribution formulas to calculate the Gini coefficient numerically

	f(x)	**F(x)**	**Q(u)**
Lognormal	$f(x) = \frac{1}{x\sigma\sqrt{2\pi}}\exp\left\{\frac{-(\ln(x)-\mu)^2}{2\sigma^2}\right\}$	$F(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right)$	$Q(u) = \exp\{\mu + \sigma\Phi^{-1}(u)\}$
Burr	$f(x) = ckx^{c-1}(1+x^c)^{-k-1}$	$F(x) = 1-(1+x^c)^{-k}$	$Q(u) = \left((1-u)^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}}$

For the Pareto distribution, the Gini coefficient can be calculated exactly. We begin by simplifying the integral part of formula (3.2.2) for the Pareto distribution:

$$\int_0^\infty F(x)\big(1-F(x)\big)\,dx = \int_k^\infty \left(1-\left(\frac{k}{x}\right)^\alpha\right)\left(\frac{k}{x}\right)^\alpha dx.$$

Let $y = \frac{k}{x}$, then $x = \frac{k}{y}$ and $dx = -\frac{k}{y^2}dy$. Regarding the bounds, as $x \to \infty, y \to 0$ and as $x \to k, y \to 1$.

Thus,

$$I_G = \int_1^0 (1-(y)^\alpha)\,(y)^\alpha\left(-\frac{k}{y^2}\right)dy$$

$$= k\left[\frac{1}{\alpha-1}y^{\alpha-1} - \frac{1}{2\alpha-1}y^{2\alpha-1}\right]_0^1$$

$$= k\left[\frac{1}{\alpha-1} - \frac{1}{2\alpha-1}\right]$$

$$= k\left[\frac{2\alpha-1-\alpha+1}{(\alpha-1)(2\alpha-1)}\right] = \frac{k\alpha}{(\alpha-1)(2\alpha-1)}.$$

Next, we consider the expected value, μ, in formula (3.2.2):

$$E(X) = \mu = \int_k^\infty x f(x)dx$$

$$= \int_k^\infty x\alpha k^\alpha x^{-\alpha-1}dx = \frac{k\alpha}{\alpha-1}.$$

Now we can determine the formula to calculate the Gini coefficient for the Pareto distribution:

$$I_G = \frac{(\alpha-1)}{k\alpha}\frac{k\alpha}{(\alpha-1)(2\alpha-1)},$$

which simplifies to

$$I_G = \frac{1}{(2\alpha-1)}. \qquad (3.2.4)$$

3.2.2 Quintile share ratio

The quintile share ratio (QSR) can be defined as the ratio of the total income received by 20% of a country's population with the highest income, to the total income received by 20% of a country's population with the lowest income (Eurostat, 2003). The more the calculated value differs from one, the greater the spread of income.

To calculate the quintile share ratio, the population has to be divided into quintiles. To do this, the population first has to be sorted in ascending order, from smallest to largest, according to income. The first quintile then equals the total income received by the 20% of individuals at the lower end of the distribution, i.e. the total income of the 20% of individuals with the lowest income. The last, or 5th, quintile is equal to the total income received by the 20% of individuals at the upper end of the distribution, i.e. the total income of the 20% of individuals with the highest income. If there are no weights, the quintile share ratio is simply the last quintile divided by the first quintile (Eurostat, 2003), i.e.

$$QSR = \frac{5th\ quintile}{1st\ quintile}. \qquad (3.2.5)$$

For a finite population, $x_1, x_2, \ldots, x_N$, the QSR is defined as

$$QSR = \frac{\left[\sum_{i=[0.8N]+1}^{N} X_{i,N}\right]}{\left[\sum_{i=1}^{[0.2N]} X_{i,N}\right]}, \qquad (3.2.6)$$

where $X_{1,N} < X_{2,N} < \cdots < X_{N,N}$ are the ordered income associated with the finite population and $[y]$ is the largest integer smaller than or equal to y.

For an infinite population the quintile share ratio can be defined by the following formula (Kpanzou, 2011):

$$QSR = \frac{\int_{Q(0.8)}^{\infty} xf(x)dx}{\int_{0}^{Q(0.2)} xf(x)dx} = \frac{E(X)I\big(X > Q(0.8)\big)}{E(X)I\big(X \leq Q(0.2)\big)}, \qquad (3.2.7)$$

where $Q(.)$ indicates the quantile function of a distribution and $I(.)$ is an indicator function.

From formula (3.2.7), the general formula to compute the QSR for a distribution numerically is derived as follows:

Let $u = F(x)$, then $x = F^{-1}(u) = Q(u)$ and $dx = q(u)du$. The formula for the QSR in this case becomes

$$QSR = \frac{\int_{0.8}^{1} Q(u)du}{\int_{0}^{0.2} Q(u)du}. \tag{3.2.8}$$

For the lognormal and Burr distributions, the QSR has to be calculated numerically using formula (3.2.8). For the Pareto distribution, the QSR can be calculated exactly, using the following formula:

$$QSR = \frac{\int_{Q(0.8)}^{\infty} xf(x)dx}{\int_{k}^{Q(0.2)} xf(x)dx},$$

where

$$\begin{aligned}\int_{Q(0.8)}^{\infty} xf(x)dx &= \alpha k^{\alpha} \int_{Q(0.8)}^{\infty} x^{-\alpha} dx \\ &= -\frac{\alpha k^{\alpha}}{-\alpha + 1} Q(0.8)^{-\alpha+1},\end{aligned}$$

and

$$\begin{aligned}\int_{k}^{Q(0.2)} xf(x)dx &= \alpha k^{\alpha} \int_{k}^{Q(0.2)} x^{-\alpha} dx \\ &= \frac{\alpha k^{\alpha}}{-\alpha + 1}\left(Q(0.2)^{-\alpha+1} - k^{-\alpha+1}\right).\end{aligned}$$

Thus,

$$QSR = \frac{\int_{Q(0.8)}^{\infty} xf(x)dx}{\int_{0}^{Q(0.2)} xf(x)dx} = -\frac{\alpha k^{\alpha}}{-\alpha + 1} Q(0.8)^{-\alpha+1} \div \frac{\alpha k^{\alpha}}{-\alpha + 1}\left(Q(0.2)^{-\alpha+1} - k^{-\alpha+1}\right),$$

which simplifies to

$$\frac{Q(0.8)^{-\alpha+1}}{k^{-\alpha+1} - Q(0.2)^{-\alpha+1}}. \tag{3.2.9}$$

3.2.3. Theil measure

Another class of measures of inequality is the general entropy measures. The most widely-used measures in this class are the Theil indices, that satisfy all six of the above criteria (section 3.1) for a good measure of inequality. The formula for the general entropy measure for a finite population is

$$GE(\alpha) = \frac{1}{\alpha(a-1)}\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i}{\bar{x}}\right)^{\alpha} - 1\right], \tag{3.2.10}$$

where $\bar{x}$ is the mean income per person.

The formula for the general entropy measure for an infinite population is

$$GE(\alpha) = \int_0^{\infty} \frac{1}{\alpha(a-1)}\left[\left(\frac{x}{\mu}\right)^{\alpha} - 1\right] dF(x), \tag{3.2.11}$$

where $\mu = E(X)$ (Cowell & Flachaire, 2007).

The values of the general entropy inequality measure vary between zero and infinity. A value of zero indicates an equal distribution, and larger values indicate higher levels of inequality. The parameter α represents the weight given to distances between incomes at different parts of the income distribution. Smaller values of α are more sensitive to changes in the lower tail of the distribution, while larger values of α are more sensitive to changes in the upper tail.

The Theil measure, also known as Theil's T, is a special case of the general entropy measure where $\alpha = 1$. The formula for a finite population can be written as

$$GE(1) = \frac{1}{N}\sum_{i=1}^{N}\frac{x_i}{\bar{x}}\ln\left(\frac{x_i}{\bar{x}}\right). \tag{3.2.12}$$

The Theil measure allows one to decompose inequality into parts: the part that is due to inequality within areas (for example, provinces), the part that is due to differences between areas (for example, the income gap between provinces), and the source of changes in inequality over time.

The Theil measure of inequality for a distribution can be calculated with the following formula:

$$GE(1) = \int_0^\infty \frac{x}{\mu} \ln\left(\frac{x}{\mu}\right) dF(x) = \frac{\nu}{\mu} - \ln \mu, \tag{3.2.13}$$

where $\nu = \int_0^\infty x \ln x\, f(x) dx$, μ is equal to $E(X)$ and $f(x)$ is the density function of the distribution (Kpanzou, 2011).

The general formula to compute the Theil measure for a distribution numerically is derived from formula (3.2.13) as follows:

$$GE(1) = \frac{1}{\mu}\int_0^\infty x \ln x\, dF(x) - \ln \mu.$$

Let $u = F(x)$, then $x = F^{-1}(u) = Q(u)$ and $dx = q(u)du$. The formula for $GE(1)$ then becomes

$$GE(1) = \frac{1}{\mu}\int_0^1 Q(u) \ln\big(Q(u)\big)\, du - \ln \mu. \tag{3.2.14}$$

For the lognormal and Burr distributions, the Theil measure has to be calculated numerically using formula (3.2.14). For the Pareto distribution, the exact value of the Theil measure can be calculated using formula (3.2.14). This gives

$$\int_k^\infty x \ln(x) f(x)dx = \alpha k^\alpha \int_k^\infty x^{-\alpha} \ln(x) dx$$

$$= \alpha k^\alpha \left[\left[\frac{1}{-\alpha+1} x^{-\alpha+1} \ln(x)\right]_k^\infty - \int_k^\infty \frac{1}{-\alpha+1} x^{-\alpha+1} \frac{1}{x} dx \right]$$

$$= \alpha k^\alpha \left[\left(0 - \frac{k^{-\alpha+1}}{-\alpha+1} \ln(k)\right) - \int_k^\infty \frac{1}{-\alpha+1} x^{-\alpha} dx \right]$$

$$= \alpha k^\alpha \left[\frac{k^{-\alpha+1}}{\alpha-1} \ln(k) - \left[\frac{1}{(-\alpha+1)^2} x^{-\alpha+1}\right]_k^\infty \right]$$

$$= \alpha k^\alpha \left[\frac{k^{-\alpha+1}}{\alpha-1} \ln(k) - \left(0 - \frac{1}{(-\alpha+1)^2} k^{-\alpha+1}\right) \right]$$

$$= \alpha k^\alpha \left[\frac{k^{-\alpha+1}}{\alpha-1} \ln(k) + \frac{k^{-\alpha+1}}{(\alpha-1)^2} \right]$$

$$= \frac{\alpha k}{\alpha-1}\left[\ln(k) + \frac{1}{\alpha-1}\right].$$

Thus, the exact value of the Theil measure becomes

$$GE(1) = \frac{1}{\mu}\int_k^{\infty} x\ln(x) f(x)dx - \ln(\mu)$$

$$= \frac{\alpha - 1}{k\alpha}\frac{\alpha k}{\alpha - 1}\left[\ln(k) + \frac{1}{\alpha - 1}\right] - \ln\left(\frac{k\alpha}{\alpha - 1}\right)$$

$$= \ln(k) + \frac{1}{\alpha - 1} - \ln\left(\frac{k\alpha}{\alpha - 1}\right). \qquad (3.2.15)$$

3.2.4 Atkinson measure

Another inequality measure is the Atkinson measure. The Atkinson measure can be calculated using the following formula for a finite population:

$$A_\varepsilon = \begin{cases} 1 - \left[\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(\dfrac{x_i}{\bar{x}}\right)^{1-\varepsilon}\right]^{\frac{1}{1-\varepsilon}} & for\ \varepsilon \neq 1 \\ 1 - \dfrac{\prod_{i=1}^{N}\left(x_i^{\frac{1}{N}}\right)}{\bar{x}} & for\ \varepsilon = 1. \end{cases} \qquad (3.2.16)$$

The parameter ε is a measure of the degree of inequality aversion, or the relative sensitivity to transfers at different income levels. As ε become larger, more weight is attached to transfers at the lower tail of the distribution, and less weight to transfers at the upper tail. If $\varepsilon \to \infty$, only the transfers to the very lowest income group will be taken into account, while $\varepsilon = 0$ will rank distributions solely according to total income (Atkinson, 1970).

The Atkinson measure can also be written in the following form, which is suitable when using a distribution (Cowell & Flachaire, 2007):

$$A_\varepsilon = 1 - \left[\int_0^{\infty}\left(\frac{x}{\mu}\right)^{1-\varepsilon} f(x)dx\right]^{\frac{1}{1-\varepsilon}} = 1 - \frac{\mu_{1-\varepsilon}^{\frac{1}{1-\varepsilon}}}{\mu}\ for\ \varepsilon > 0, \varepsilon \neq 1, \qquad (3.2.17)$$

where $\mu_{1-\varepsilon} = \int_0^{\infty} x^{1-\varepsilon} f(x)dx$.

For the case where $\varepsilon = 1$, the formula is given as

$$A_1 = 1 - \frac{e^{\int_0^{\infty}\ln(x)dx}}{\mu}.$$

For this study, the calculation for the Atkinson measure is carried out with the computer package R, which makes use of an $\varepsilon = 0.5$.

The general formula to compute the Atkinson measure for a distribution numerically is derived as

$$A_\varepsilon = 1 - \left[\left(\frac{1}{\mu}\right)^{1-\varepsilon} \int_0^\infty x^{1-\varepsilon} dF(x)\right]^{\frac{1}{1-\varepsilon}}.$$

Let $u = F(x)$, then $x = F^{-1}(u) = Q(u)$ and $dx = q(u)du$. The formula for A_ε then becomes

$$A_\varepsilon = 1 - \left[\left(\frac{1}{\mu}\right)^{1-\varepsilon} \int_0^1 Q(u)^{1-\varepsilon} du\right]^{\frac{1}{1-\varepsilon}}. \quad (3.2.18)$$

For the lognormal and Burr distributions, the Atkinson measure has to be calculated numerically with the formula (3.2.18). For the Pareto distribution, the Atkinson measure can be calculated exactly with the following formula:

$$\begin{aligned} A_\varepsilon &= 1 - \left[\int_k^\infty \left(\frac{x}{\mu}\right)^{1-\varepsilon} f(x)dx\right]^{\frac{1}{1-\varepsilon}} \\ &= 1 - \left[\frac{1}{\mu^{1-\varepsilon}} \int_k^\infty x^{1-\varepsilon} \alpha k^\alpha x^{-\alpha-1} dx\right]^{\frac{1}{1-\varepsilon}} \\ &= 1 - \left[\frac{\alpha k^\alpha}{\mu^{1-\varepsilon}} \int_k^\infty x^{-\alpha-\varepsilon} dx\right]^{\frac{1}{1-\varepsilon}}. \end{aligned}$$

Since

$$E(X) = \mu = \frac{k\alpha}{\alpha - 1}$$

and

$$\begin{aligned} \int_k^\infty x^{-\alpha-\varepsilon} dx &= \left[\frac{1}{-\alpha-\varepsilon+1} x^{-\alpha-\varepsilon+1}\right]_k^\infty \\ &= \left(0 - \frac{1}{-\alpha-\varepsilon+1} k^{-\alpha-\varepsilon+1}\right) \\ &= \frac{k^{-\alpha-\varepsilon+1}}{\alpha+\varepsilon-1}, \end{aligned}$$

it is found that the exact Atkinson measure becomes

$$A_\varepsilon = 1 - \left[\frac{\alpha k^\alpha}{\mu^{1-\varepsilon}} \int_k^\infty x^{-\alpha-\varepsilon} dy\right]^{\frac{1}{1-\varepsilon}}$$

$$= 1 - \left[\alpha k^{\alpha}\left(\frac{\alpha-1}{k\alpha}\right)^{1-\varepsilon}\frac{k^{-\alpha-\varepsilon+1}}{\alpha+\varepsilon-1}\right]^{\frac{1}{1-\varepsilon}}$$

$$= 1 - \left[\left(\frac{\alpha-1}{k\alpha}\right)^{1-\varepsilon}\frac{\alpha k^{-\varepsilon+1}}{\alpha+\varepsilon-1}\right]^{\frac{1}{1-\varepsilon}}. \qquad (3.2.19)$$

3.3 SUMMARY

In this chapter, the measures of inequality used in this study were examined. The measures included the Gini coefficient, QSR, Theil measure and Atkinson measure. Some background information on each of the measures was given, and a formula was derived for each of the measures for each of the distributions that was simulated from. The table below summarises the formulas that are used to calculate the measures for each of the distributions. The measures for the Pareto distribution can be calculated exactly while the measures for the lognormal and Burr distributions have to be calculated numerically.

Table 3.3.1: Formulas to calculate the measures of inequality for each distribution

	Pareto	**Lognormal**	**Burr**
Gini	$\frac{1}{(2\alpha-1)}$	$\frac{1}{\mu}\int_0^1 u(1-u)\frac{1}{f(Q(u))}du$	$\frac{1}{\mu}\int_0^1 u(1-u)\frac{1}{f(Q(u))}du$
QSR	$-\frac{Q(0.8)^{-\alpha+1}}{Q(0.2)^{-\alpha+1}-k^{-\alpha+1}}$	$\frac{\int_{0.8}^1 Q(u)du}{\int_0^{0.2} Q(u)du}$	$\frac{\int_{0.8}^1 Q(u)du}{\int_0^{0.2} Q(u)du}$
Theil	$\ln(k)+\frac{1}{\alpha-1}-\ln\left(\frac{k\alpha}{\alpha-1}\right)$	$\frac{1}{\mu}\int_0^1 Q(u)\ln(Q(u))\,du-\ln(\mu)$	$\frac{1}{\mu}\int_0^1 Q(u)\ln(Q(u))\,du-\ln(\mu)$
Atkinson	$1-\left[\left(\frac{\alpha-1}{k\alpha}\right)^{1-\varepsilon}\frac{\alpha k^{-\varepsilon+1}}{\alpha+\varepsilon-1}\right]^{\frac{1}{1-\varepsilon}}$	$1-\left[\left(\frac{1}{\mu}\right)^{1-\varepsilon}\int_0^1 Q(u)^{1-\varepsilon}du\right]^{\frac{1}{1-\varepsilon}}$	$1-\left[\left(\frac{1}{\mu}\right)^{1-\varepsilon}\int_0^1 Q(u)^{1-\varepsilon}du\right]^{\frac{1}{1-\varepsilon}}$

The methods discussed in this chapter will be implemented on simulated and real datasets in Chapters 5 and 6, and will be used to calculate the true values. The measures obtained for each method will then be compared to the true measures. In the next chapter, the simulation process will be studied, and parameters for each of the distributions that are simulated from will be selected.

CHAPTER 4
SIMULATION STUDY

4.1 INTRODUCTION

In this chapter, the focus of the study turns to the simulation process, with a diagram that explains the process. We also consider the distributions that are simulated from, and calculate the 90th percentile, median, expected value and mode of each of the distributions, in order to compare the heaviness of the tails of the distributions.

4.2 SIMULATION PROCESS

Before the simulation process can commence, it is necessary to decide which distributions, $F(x)$, will be simulated from; in addition, the values for the parameters, θ, must be chosen. It is also necessary to select the samples sizes, n, and the number of replications, R, that will be used; furthermore, the interval widths (with their lower and upper bounds) for the grouped data must be decided upon beforehand.

The simulation process will be as follows.

1. Choose the distribution, F, to simulate from. Since we work with measures of inequality that are related to income data we will simulate from the Pareto, lognormal and Burr distributions. These are all heavy tail distributions.

2. Choose the corresponding parameters (θ) of F, denoted by $F(x;\theta)$. The chosen parameters for this study are given in section 4.4.

3. Choose n, the sample sizes of simulated observations. The different values we use for n are 1 000, 5 000, 10 000 and 15 000.

4. Choose R, the number of repetitions. We keep R fixed through the simulation process at 1 000.

5. Simulate nR observations from $F(x;\theta)$.

6. Allocate n observations to each of the R replicates.

7. For each replicate, classify the simulated observations into the predetermined intervals.

8. Execute the calculations for the different methods on the interval datasets.

    8.1 Calculate the midpoint of each interval for the midpoint method. Allocate the midpoint to each observation as a point value.

8.2 Estimate the parameters of the Pareto distribution and calculate the Pareto mean for each interval.

8.3 Estimate the parameters of the lognormal distribution and calculate the lognormal mean for each interval.

8.4 Calculate the random midpoint for each observation.

8.5 Use the estimated Pareto parameters and calculate a random Pareto point for each observation.

8.6 Use the estimated lognormal parameters and calculate a random lognormal point for each observation.

9. Calculate the measures of inequality and store the results.

9.1 Calculate these measures on each dataset created by each method. Each measure is calculated per replication so that there are R of each measure per method.

9.2 Calculate these measures on the raw simulated data of step 4 per replicate.

10. If the simulation process

- is completed for all values of n, simulate from new θ of $F(x;\theta)$ i.e. step 2;

- otherwise, choose different n and repeat the simulation process from step 3 for a new n.

11. If the simulation process

- is completed for all values of θ of $F(x;\theta)$, simulate from a new F i.e. step 1;

- otherwise, choose a different θ of $F(x;\theta)$ and repeat the simulation process from step 2 for a new θ.

12. If the simulation process

- is completed for all F, the data can be analysed;

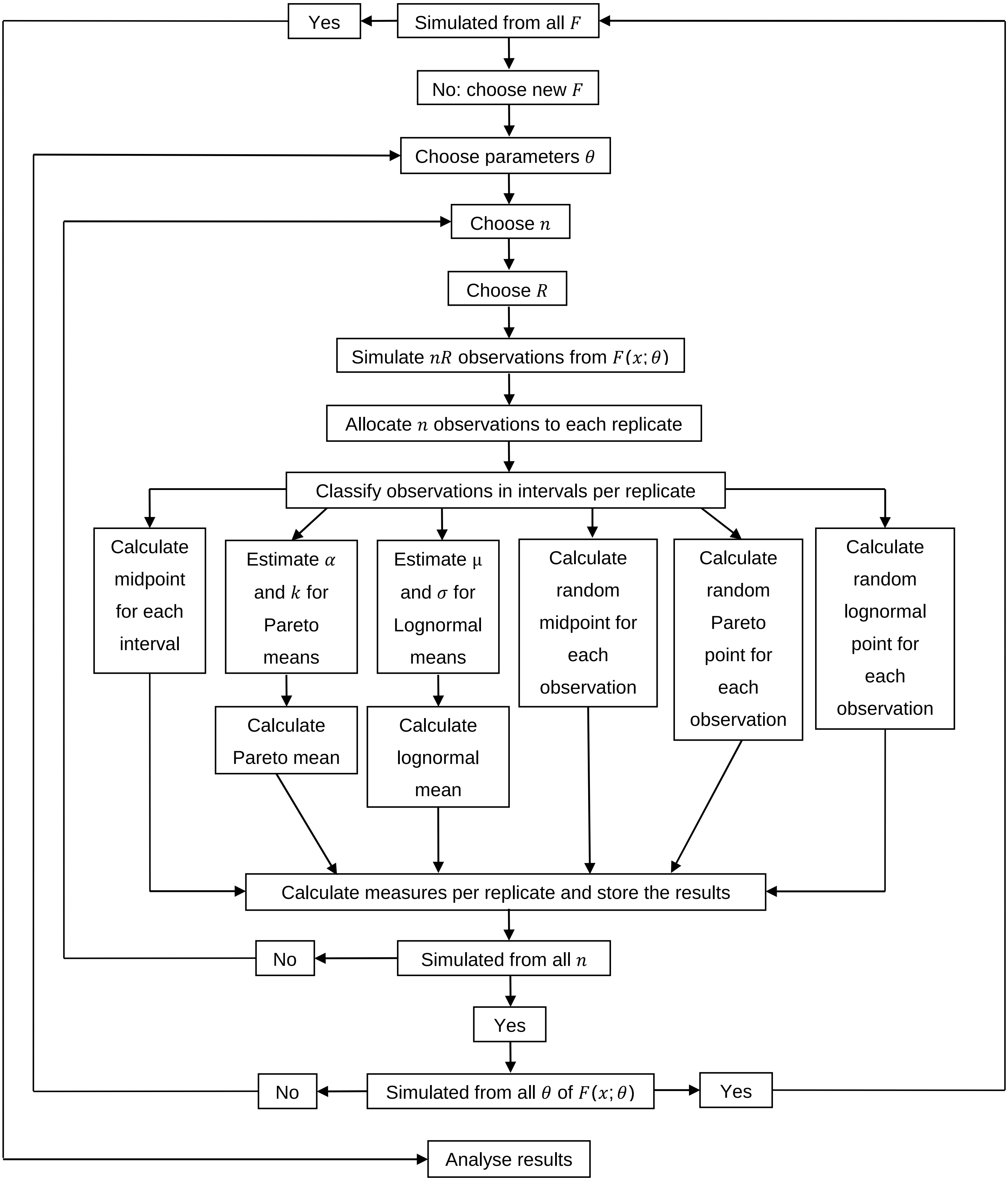
- otherwise, choose a different F and repeat the simulation process from step 1.

The true values of the measures of inequality must be calculated for the distributions that are simulated from (the derivations for these calculations were introduced in section 3.2), either during the simulation process, or after the process is completed. The measures calculated from each method are then compared to the true measure. This forms part of the process of analysis, which is the final phase in the simulation process.

The estimates in the simulation process are calculated using statistical programming packages R and SAS, and the data is stored in Microsoft Excel format, where the final summarised analyses are undertaken, processed, and presented in tabular format.

This simulation process is now given in diagrammatic form.

4.3 DIAGRAM OF SIMULATION PROCESS:



4.4 PARAMETERS USED IN THE STUDY FOR THE DIFFERENT DISTRIBUTIONS

In this section, we discuss the values of the parameters for the distributions that will be simulated from. Different Pareto-, lognormal- and Burr distributions will be used. Hereafter, the 90th percentile, median, expected value and mode of each distribution are calculated in order to compare the heaviness of the tails of the different distributions.

4.4.1 Pareto

The Pareto distribution has parameters α and k, with the following density and distribution functions:

$$f(x) = \frac{\alpha k^{\alpha}}{x^{\alpha+1}} \text{ for } x \geq k \text{ and } \alpha > 0$$

$$F(x) = 1 - \left(\frac{k}{x}\right)^{\alpha} \text{ for } x \geq k \text{ and } \alpha > 0.$$

In this study, we use $k = 1$. With $k = 1$, the density and distribution functions become

$$f(x) = \frac{\alpha}{x^{\alpha+1}} \text{ for } x \geq 1 \text{ and } \alpha > 0$$

$$F(x) = 1 - \left(\frac{1}{x}\right)^{\alpha} \text{ for } x \geq 1 \text{ and } \alpha > 0.$$

In the calculations carried out for the measures of inequality, the expected value of the function must exist. For the expected value to exist, the following has to exist:

$$E(X) = \int_1^{\infty} x\alpha x^{-\alpha-1}\, dx$$

$$= \alpha \int_1^{\infty} x^{-\alpha}\, dx$$

$$= \frac{\alpha}{-\alpha+1} x^{-\alpha+1}\Big|_1^{\infty} = \frac{\alpha}{\alpha-1}. \tag{4.4.1}$$

This means that for the expected value to exist,

$$1 - \alpha > 0$$

i.e. $\alpha > 1$.

Since α has to be larger than one for the expected value to exist, we make use of the following values of α in this study:

$$\alpha = 3,\ 1.5,\ 1.1.$$

Now that we have decided on the values for the parameters of the Pareto distribution, the 90th percentile can be calculated. We will begin by deriving a general formula to calculate the pth percentile. Since

$$F(x) = 1 - \left(\frac{1}{x}\right)^{\alpha} = 1 - p,$$

it follows that

$$\mathrm{x_{percentile}} = \left(\frac{1}{p}\right)^{\frac{1}{\alpha}}. \qquad (4.4.2)$$

Thus, for the 90th percentile $p = 0.1 \Rightarrow 1 - p = 0.9$

$$x_{percentile} = \left(\frac{1}{0.1}\right)^{\frac{1}{\alpha}} = 10^{\frac{1}{\alpha}}. \qquad (4.4.3)$$

The 90th percentile for the different values of α is equal to the following:

For $\alpha = 3$: $x_{percentile} = 10^{\frac{1}{3}} = 2.1544$.

For $\alpha = 1.5$: $x_{percentile} = 10^{\frac{1}{1.5}} = 4.6416$.

For $\alpha = 1.1$: $x_{percentile} = 10^{\frac{1}{1.1}} = 8.1113$.

Next, we calculate the median, which is a special case of the percentile with $p = 0.5$.

$$F(x) = 1 - \left(\frac{1}{x}\right)^{\alpha} = 0.5.$$

From this it follows that

$$x_{median} = 2^{\frac{1}{\alpha}}. \qquad (4.4.4)$$

The median for the different values of α is equal to the following:

For $\alpha = 3$: $Median = 1.2599$.

For $\alpha = 1.5$: $Median = 1.5874$.

For α = 1.1: $Median$ = 1.8779.

Next, we consider the expected value derived in formula (4.4.1). The expected values for the different values of alpha will be the following:

For α = 3: $E(X)$ = 1.5.

For α = 1.5: $E(X)$ = 3.

For α = 1.1: $E(X)$ = 11.

It is clear from the form of the Pareto density that it is monotone decreasing. Its mode is therefore at its lower end-point, in our case at one.

4.4.2 Lognormal

The mean of the lognormal distribution will be denoted by 'η', while the standard deviation of the lognormal distribution will be denoted by 'τ'. The mean of the standard normal distribution will be denoted by 'μ', and the standard deviation of the standard normal distribution will be denoted by 'σ'.

The lognormal distribution has the following density and distribution function:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{\frac{-(\ln(x) - \mu)^2}{2\sigma^2}\right\}; x > 0$$

$$F(x) = \Phi\left(\frac{\ln(x) - \mu}{\sigma}\right),$$

where Φ is the cumulative distribution function of the standard normal distribution.

Since all the moments of the lognormal distribution exist, the expected value will exist for all values of μ and σ.

Let $X \sim lognormal(\eta, \tau^2)$ and $Y \sim normal(\mu, \sigma^2)$, then $X = \exp(Y)$ and $Y = \ln(X)$.

Next, the formula for converting the parameters of the lognormal distribution to the parameters of the standard normal distribution are explicated. We will start by solving σ^2.

Since

$$E(X) = \eta = E(\exp(Y)) = m_Y(1) = \exp\left(\mu + \frac{1}{2}\sigma^2\right) \text{ and}$$

$$E(X^2) = E(\exp(2Y)) = m_Y(2) = \exp(2\mu + 2\sigma^2),$$

it follows that the variance is equal to

$$\begin{aligned}
Var(X) = \tau^2 &= E(X^2) - \left(E(X)\right)^2 \\
&= \exp(2\mu + 2\sigma^2) - \left(\exp\left(\mu + \frac{1}{2}\sigma^2\right)\right)^2 \\
&= \exp(2\mu + 2\sigma^2) - \exp\left(2\left(\mu + \frac{1}{2}\sigma^2\right)\right) \\
&= \exp(\sigma^2)\exp(2\mu + \sigma^2) - \exp(2\mu + \sigma^2) \\
&= \exp(\sigma^2)\eta^2 - \eta^2.
\end{aligned}$$

The previously stated formula of the variance can also be written as

$$\tau^2 = \eta^2(\exp(\sigma^2) - 1),$$

from which it follows that

$$\exp(\sigma^2) = \frac{\tau^2}{\eta^2} + 1,$$

and thus

$$\sigma^2 = \ln\left(\frac{\tau^2}{\eta^2} + 1\right). \tag{4.4.5}$$

Next, we derive a formula for μ.

$$\eta^2 = \exp(2\mu + \sigma^2) = \exp(2\mu)\exp(\sigma^2);$$

thus,

$$\exp(2\mu) = \frac{\eta^2}{\exp(\sigma^2)} = \eta^2 \div \left(\frac{\tau^2}{\eta^2} + 1\right) = \eta^2\left(\frac{\eta^2}{\tau^2 + \eta^2}\right) = \frac{\eta^4}{\tau^2 + \eta^2},$$

$$\exp(\mu) = \left(\frac{\eta^4}{\tau^2 + \eta^2}\right)^{\frac{1}{2}} = \frac{\eta^2}{(\tau^2 + \eta^2)^{\frac{1}{2}}},$$

and this leads to

$$\mu = \ln\left(\frac{\eta^2}{(\tau^2 + \eta^2)^{\frac{1}{2}}}\right). \tag{4.4.6}$$

We make use of the following values of μ and σ in this study:

$$\mu = 0.5, \sigma = 0.5;$$

$$\mu = 0.6, \sigma = 1;$$

$$\mu = 1.5, \sigma = 0.8.$$

Having decided the values for the parameters of the lognormal distribution, we calculate the 90th percentile. We start by deriving the following general formula to calculate the pth percentile:

$$F(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right) = 1 - p$$

$$\ln x = \sigma\Phi^{-1}(1 - p) + \mu.$$

It then follows that

$$x_{percentile} = \exp(\sigma\Phi^{-1}(1 - p) + \mu). \tag{4.4.7}$$

For the 90th percentile: $p = 0.1 \Rightarrow 1 - p = 0.9$ and

$\Phi^{-1}(1 - p) = \Phi^{-1}(0.9) = 1.2816$.

The 90th percentile for the different values of μ and σ is:

For $\mu = 0.5, \sigma = 0.5$: $x_{percentile} = \exp(0.5 * 1.2816 + 0.5) = 3.1293$.

For $\mu = 0.6, \sigma = 1$: $x_{percentile} = \exp(1 * 1.2816 + 0.6) = 6.564$.

For $\mu = 1.5, \sigma = 0.8$: $x_{percentile} = \exp(0.8 * 1.2816 + 1.5) = 12.4944$.

Taking $p = 0.5$ in formula (4.4.7) gives the median as

$$x_{median} = \exp(\sigma\Phi^{-1}(0.5) + \mu) = \exp(\mu). \tag{4.4.8}$$

The median for the different values of μ and σ is:

For $\mu = 0.5, \sigma = 0.5$: $x_{med} = \exp(0.5) = 1.6487$.

For $\mu = 0.6, \sigma = 1$: $x_{med} = \exp(0.6) = 1.8221$.

For $\mu = 1.5, \sigma = 0.8$: $x_{med} = \exp(1.5) = 4.4817$.

Next, we consider the expected value. The moments for the lognormal distribution can be calculated with the following formula:

$$m_Y(t) = \exp\left(\mu t + \frac{1}{2}t^2\sigma^2\right).$$

Thus,

$$E(X) = m_Y(1) = \exp\left(\mu + \frac{1}{2}\sigma^2\right). \tag{4.4.9}$$

The expected values for the different values of μ and σ are:

For $\mu = 0.5, \sigma = 0.5$: $E(X) = \exp\left(0.5 + \frac{1}{2}0.5^2\right) = 1.8682$.

For $\mu = 0.6, \sigma = 1$: $E(X) = \exp\left(0.6 + \frac{1}{2}1^2\right) = 3.0042$.

For $\mu = 1.5, \sigma = 0.8$: $E(X) = \exp\left(1.5 + \frac{1}{2}0.8^2\right) = 6.1719$.

Finally, we consider the mode for the lognormal distribution. Note that

$$F(x) = \Phi\left(\frac{\ln(x) - \mu}{\sigma}\right),$$

$$f(x) = \frac{1}{\sigma x}\phi\left(\frac{\ln(x) - \mu}{\sigma}\right),$$

and for $f'(x) = 0$, it follows that

$$f'(x) = \frac{1}{\sigma}\left[-\frac{1}{x^2}\phi\left(\frac{\ln(x) - \mu}{\sigma}\right) + \frac{1}{x}\left(-\left(\frac{\ln(x) - \mu}{\sigma}\right)\phi\left(\frac{\ln(x) - \mu}{\sigma}\right)\frac{1}{x\sigma}\right)\right] = 0$$

$$\frac{1}{x^2}\phi\left(\frac{\ln(x) - \mu}{\sigma}\right) + \frac{1}{x^2\sigma}\left(\frac{\ln(x) - \mu}{\sigma}\right)\phi\left(\frac{\ln(x) - \mu}{\sigma}\right) = 0$$

$$1 + \frac{\ln(x) - \mu}{\sigma^2} = 0;$$

thus

$$x_{mode} = \exp(\mu - \sigma^2). \tag{4.4.10}$$

The mode for the different values of μ and σ is:

For $\mu = 0.5, \sigma = 0.5$: $x_{mode} = \exp(0.5 - 0.5^2) = 1.284$.

For $\mu = 0.6, \sigma = 1$: $x_{mode} = \exp(0.6 - 1^2) = 0.6703$.

For $\mu = 1.5, \sigma = 0.8$: $x_{mode} = \exp(1.5 - 0.8) = 2.3632$.

4.4.3 Burr

The Burr distribution has the parameters c and k, with the following density and distribution functions:

$$f(x) = ckx^{c-1}(1 + x^c)^{-k-1} \quad \text{for } x > 0,\ c > 0,\ k > 0$$

$$F(x) = 1 - (1 + x^c)^{-\mathrm{k}} \quad \text{for } x > 0,\ c > 0,\ k > 0.$$

In the calculations carried out for the measures of inequality, the expected value of the function must exist.

$$E(X)I(X > a) = \int_a^{\infty} xf(x)\,dx$$

$$= ck\int_a^{\infty} x^c(1 + x^c)^{-k-1}\,dx.$$

Note that for x large $(1 + x^c)^{-k-1} \simeq (x^c)^{-k-1}$. The integrand $x^c(1 + x^c)^{-k-1}$ then becomes x^{-ck}, which is integrable at ∞ for

$$-ck + 1 < 0$$

$$\text{i.e. } ck > 1.$$

Since ck has to be larger than one for the expected value to exist, we make use of the following values of c and k in this study:

$$c = 1.2, k = 2.5;$$

$$c = 3, k = 1;$$

$$c = 1.3, k = 1.5;$$

$$c = 3, k = \frac{1.1}{3}.$$

Now that we have decided on the values for the parameters of the Burr distribution, we calculate the 90th percentile. We begin by deriving a general formula to calculate the pth percentile:

$$F(x) = 1 - (1 + x^c)^{-k} = 1 - p.$$

From this it follows that

$$x_{percentile} = \left(p^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}}. \tag{4.4.11}$$

For the 90th percentile, $p = 0.1 \Rightarrow 1 - p = 0.9$

$$x_{percentile} = \left(0.1^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}}$$

$$x_{percentile} = \left(10^{\frac{1}{k}} - 1\right)^{\frac{1}{c}}. \tag{4.4.12}$$

The 90th percentile for the different values of c and k will be:

For $c = 1.2, k = 2.5$: $x_{percentile} = 1.4112$.

For $c = 3, k = 1$: $x_{percentile} = 2.08$.

For $c = 1.3, k = 1.5$: $x_{percentile} = 2.7023$.

For $c = 3, k = \frac{1.1}{3}$: $x_{percentile} = 8.1062$.

Taking $p = 0.5$ in formula (4.4.11) gives the median as

$$x_{median} = \left((0.5)^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}}. \tag{4.4.13}$$

The median for the different values of c and k is equal to:

For $c = 1.2, k = 2.5$: $x_{med} = 0.3864$.

For $= 3, k = 1$: $x_{med} = 1$.

For $= 1.3, k = 1.5$: $x_{med} = 0.6641$.

For $= 3, k = \frac{1.1}{3}$: $x_{med} = 1.7781$.

Next, we derive a formula to calculate the expected value.

$$E(X) = \int_0^{\infty} x f(x) dx \text{ for } x > 0, c > 0, k > 0, ck > 1$$

$$= \int_0^{\infty} x c k x^{c-1} (1 + x^c)^{-k-1} dx$$

$$= ck\int_0^\infty x^c(1+x^c)^{-k-1}dx.$$

Let $x^c = t$, then $x = t^{\frac{1}{c}}$ and $dx = \frac{1}{c}t^{\frac{1}{c}-1}dt$. Then

$$E(X) = ck\int_0^\infty x^c(1+x^c)^{-k-1}dx$$

$$= ck\int_0^\infty t(1+t)^{-k-1}\frac{1}{c}t^{\frac{1}{c}-1}dt$$

$$= k\int_0^\infty t^{\frac{1}{c}}(1+t)^{-(k+1)}dt.$$

The general formula for the Beta function is

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt\ ,$$

which can also be written in the following form (Gradshteyn & Ryzhik, 1965):

$$B(\alpha,\beta) = \int_0^\infty t^{\alpha-1}(1+t)^{-(\alpha+\beta)}dt,$$

and it follows that

$$E(X) = k\int_0^\infty t^{\frac{1}{c}}(1+t)^{-(k+1)}dt$$

$$= kB\left(1+\frac{1}{c}, k-\frac{1}{c}\right). \qquad (4.4.14)$$

The expected value for the different values of c and k is equal to:

For $c = 1.2, k = 2.5$: $E(X) = 0.6388$.

For $c = 3, k = 1$: $E(X) = 1.2092$.

For $c = 1.3, k = 1.5$: $E(X) = 1.3046$.

For $c = 3, k = \frac{1.1}{3}$: $E(X) = 10.8412$.

Finally, the mode of the Burr distribution is derived by considering the function

$$f(x) = ckx^{c-1}(1+x^c)^{-k-1}.$$

If $f'(x) = 0$, then

$$f'(x) = \text{ck}[(c-1)x^{c-2}(1+x^c)^{-k-1} + x^{c-1}(-k-1)(1+x^c)^{-k-2}cx^{c-1}] = 0$$

$$x^{c-2}(c-1)(1+x^c)^{-k-1} + cx^{2c-2}(-k-1)(1+x^c)^{-k-2} = 0$$

$$x^{c-2}(1+x^c)^{-k-2}[(c-1)(1+x^c) + cx^c(-k-1)] = 0$$

$$(c-1) + x^c(c-1) - cx^c(k+1) = 0$$

$$x^c[(c-1) - c(k+1)] = 1 - c$$

$$x^c = \frac{(1-c)}{[(c-1) - c(k+1)]}.$$

From this equation, it follows that

$$x_{mode} = \left(\frac{c-1}{ck+1}\right)^{\frac{1}{c}}. \qquad (4.4.15)$$

The mode for the different values of c and k is equal to:

For $c = 1.2, k = 2.5$: $x_{mode} = 0.0824$.

For $c = 3, k = 1$: $x_{mode} = 0.7937$.

For $c = 1.3, k = 1.5$: $x_{mode} = 0.1723$.

For $c = 3, k = \frac{1.1}{3}$: $x_{mode} = 0.9839$.

4.5 SUMMARY

In this chapter, the planned simulation process was discussed. Parameters were assigned to each of the distributions that will be simulated from. The Pareto and Burr distributions contain restrictions ($\alpha > 1$ for the Pareto distribution and $ck > 1$ for the Burr distribution) for their expected values to exist. These restrictions are necessary for the calculation of the measures of inequality. Parameters for these distributions were chosen according to the restrictions.

The expected value, median, mode and 90th percentile were calculated for each of the distributions with the assigned parameters. These calculated values are summarised in the following table (4.5.1):

Table 4.5.1: Summary of the expected value, median, mode and 90th percentile for each distribution

		$E(X)$	Median	Mode	90th percentile
Pareto	$\alpha = 3$	1.5	1.2599	1	2.1544
	$\alpha = 1.5$	3	1.5874	1	4.6416
	$\alpha = 1.1$	11	1.8779	1	8.1113
Lognormal	$\mu = 0.5, \sigma = 0.5$	1.8682	1.6487	1.284	3.1293
	$\mu = 0.6, \sigma = 1$	3.0042	1.8221	0.6703	6.564
	$\mu = 1.5, \sigma = 0.8$	6.1719	4.4817	2.3632	12.4944
Burr	$c = 1.2, k = 2.5$	0.6388	0.3864	0.0824	1.4112
	$c = 3, k = 1$	1.2092	1	0.7937	2.08
	$c = 1.3, k = 1.5$	1.3046	0.6641	0.1723	2.7023
	$c = 3, k = 1.1/3$	10.8412	1.7781	0.9839	8.1062

The parameters that resulted in the heaviest tail for each of the Pareto-, lognormal- and Burr distributions are indicated in Figure 4.5.1 below. The mode, expected value and 90th percentiles are indicated with vertical lines.

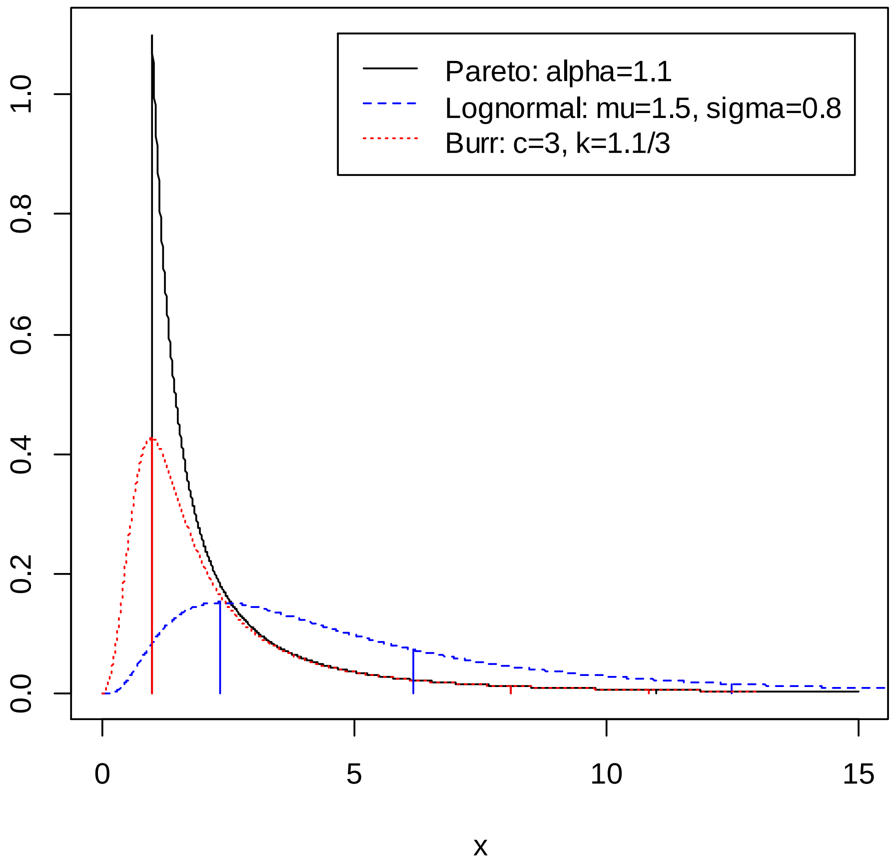


Figure 4.5.1: Heaviest tails of each distribution

From the graph on the previous page, it is clear that the Pareto distribution has a much higher peak for smaller values, with the tail areas of the Pareto and Burr distributions approximately equal. Against this, the lognormal distribution is more flat for smaller values, with a heavier tail.

The parameters that resulted in the lightest tail for each of the Pareto-, lognormal- and Burr distributions are indicated in the figure below. The mode, expected value and 90th percentiles are indicated with vertical lines. When comparing this graph to the graph with the heaviest tails, it is important to notice the scale difference for the x- and y-axes.

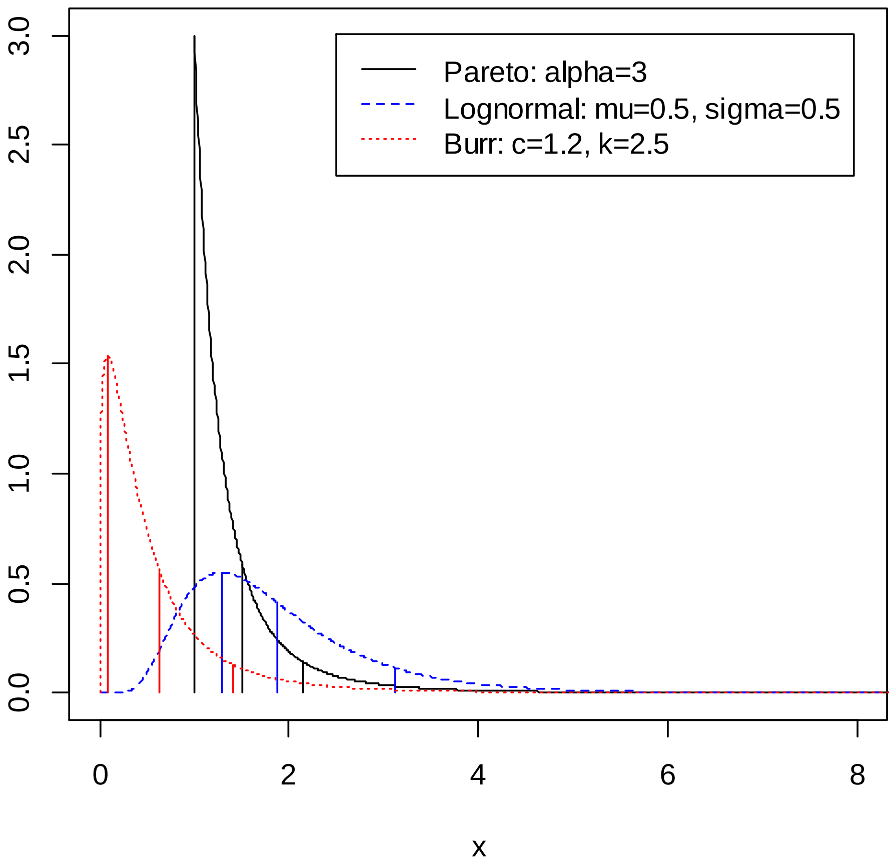


Figure 4.5.2: Lightest tails of each distribution

The Pareto- and Burr distributions are again much higher for smaller values, while the lognormal distribution is much flatter.

CHAPTER 5

ANALYSIS OF RESULTS OF SIMULATED DATA

5.1 INTRODUCTION

In the previous chapter, the distributions from which data was simulated, were determined. Four sample sizes, n=1 000, n=5 000, n=10 000 and n=15 000, are used for the simulation process. In the present chapter, the simulated data is categorised into intervals, and the methods described in sections 2.3 and 2.4 are applied in order to convert the interval data back to point data. The inequality measures (the Gini-, QSR-, Theil- and Atkinson measures) are then calculated from the point data for each method. A simulation study consisting of 1 000 replications is also undertaken. This means that 1 000 estimates per measurement per method is obtained.

In this chapter, the formulas for the different measures of performance are studied, and the results obtained from the simulated data are analysed.

5.2 MEASURES OF PERFORMANCE

Descriptive statistics are used to describe the data; this process is always the first step in the analysis of data. This study includes graphical and numerical summaries which provide an overview of the data. Measures of central tendency and measures of variability are used in order to obtain an overview of the dataset. Statistics such as the mean, median and mode are included in measures of location (central tendency), while statistics such as the range, variance and standard deviation are included in measures of variability.

We divided the 1 000 estimates per measure of inequality per method into 10 blocks, each containing 100 estimates; this was done in order that standard errors could be calculated. After this division, the mean, median, bias, root mean square error (RMSE) and median absolute deviation (MAD) are calculated for each block, as is the standard error over the blocks.

In the following section, the formulas used to calculate the RMSE, MAD and the standard error of the mean, RMSE and MAD, are defined.

5.2.1 Root mean square error (RMSE)

The root mean square error, also called the root mean square deviation, is simply the square root of the mean square error (MSE). For estimates $\hat{\theta}_1 \dots \hat{\theta}_n$ of θ, the RMSE is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_i - \theta\right)^2}. \tag{5.2.1}$$

5.2.2 Median absolute deviation (MAD)

The MAD of $\hat{\theta}_1 \dots \hat{\theta}_n$ is defined as

$$MAD = \underset{i}{median}\left|\hat{\theta}_i - \theta\right|. \tag{5.2.2}$$

Remark. In the following applications of these formulae, θ will be replaced by an average estimate.

5.2.3 Standard errors

5.2.3.1 The (estimated) standard error for the mean

We begin by calculating the estimate of each block. Let the mean for block b be indicated by $\bar{\theta}_b$. Hereafter, the mean of the block estimate is calculated.

$$\bar{\theta} = \frac{1}{B}\sum_{b=1}^{B}\bar{\theta}_b,$$

where B represents the number of blocks. The standard error of the mean is then calculated by

$$se(mean) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\bar{\theta}_b - \bar{\theta})^2}. \tag{5.2.3}$$

Some literature uses the fraction $\frac{1}{B}$ instead of $\frac{1}{B-1}$. Usually B is a large number; it then follows that $B - 1 \approx B$. We decided to use $(B - 1)$ in the denominator of the standard error, as we deal with a small B equal to 10.

5.2.3.2 The standard error for the biases

In this study, two biases are considered. The first bias is calculated as the difference between the mean and the true value, while the second bias is calculated as the difference between the median and the true value. Since income forms a positive skew distribution, it is good practice to also consider the median as a descriptive statistic. To calculate the standard error for the biases, we begin by calculating the bias, either mean minus true value or median minus true value, of each block. Let the bias for block b be indicated by $\hat{\theta}_b$. Then the mean of the block biases is given by

$$\bar{\hat{\theta}} = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}_{b,}$$

where B represents the number of blocks. The standard error of the bias can now be calculated by

$$se(bias) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_b - \bar{\hat{\theta}}\right)^2}. \qquad (5.2.4)$$

5.2.3.3 The standard error for the RMSE

In the first place, we calculate the RMSE for each block and let $\widehat{RMSE}_b$ denote the RMSE for block b. The mean of the block RMSEs can be written as

$$\overline{\widehat{RMSE}} = \frac{1}{B}\sum_{b=1}^{B} \widehat{RMSE}_{b,}$$

where B represents the number of blocks. The standard error of the RMSE is calculated by

$$se\left(\widehat{RMSE}\right) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{RMSE}_b - \overline{\widehat{RMSE}}\right)^2}. \qquad (5.2.5)$$

5.2.3.4 The standard error for the MAD

First, we calculate the MAD of each block, and let $\widehat{MAD}_b$ indicates the MAD for block b. The mean of the block MAD’s is given by

$$\overline{\widehat{MAD}} = \frac{1}{B}\sum_{b=1}^{B} \widehat{MAD}_{b,}$$

where B represents the number of blocks. The standard error of the MAD is then calculated by

$$se\left(\widehat{MAD}\right) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{MAD}_b - \overline{\widehat{MAD}}\right)^2}. \tag{5.2.6}$$

The purpose of the MAD is the avoidance of the quadratic deviations and it is also a measure of variability, as an alternative to the standard deviation. Therefore, the variability among the MADs will also be calculated using the following formula:

$$se^*\left(\widehat{MAD}\right) = \underset{b}{median}\left|\widehat{MAD}_b - MED\right|, \tag{5.2.7}$$

where $MED = \underset{j}{median}\left(\widehat{MAD}_j\right)$.

5.3 STATISTICAL ANALYSIS

For the simulation process, we simulated 1 000 times *n* observations from a distribution to form 1 000 replications. We categorised the data in intervals per replication, and applied the different methods to the interval data in order to convert it into point data. We calculated the measures of inequality on the point data per replication per method; this means that we had 1 000 measures of each measure of inequality. Next, we calculated the measures of performance to compare the results of each method with the true values. The following table is an example of results obtained for the bias (mean – true value), with *n*=15 000 for the Gini coefficient. The first value in a cell indicates the calculated bias, while the value (in brackets) indicates the standard error.

Table 5.3.1: Estimated bias with its standard error per method and per distribution (*n*=15 000)

	Parameters	midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	alpha=3	0.01222 (0.00029)	0.01451 (0.00029)	-0.07079 (0.00021)	-0.02027 (0.00033)	0.00455 (0.00028)	–	–	-0.003 (0.00026)	0.01535 (0.00035)	0.00001 (0.00026)	-0.08376 (0.0003)	0.00007 (0.00021)
	alpha=1.5	-0.37022 (0.00089)	-0.36287 (0.001)	-0.21066 (0.00187)	-0.36478 (0.00195)	-0.35943 (0.0025)	-0.35375 (0.00349)	-0.33955 (0.00881)	-0.38288 (0.00082)	-0.15302 (0.00058)	-0.00456 (0.00202)	-0.32419 (0.00104)	-0.00393 (0.00353)
	alpha=1.1	-0.37544 (0.00237)	-0.35923 (0.00252)	-0.12488 (0.00544)	-0.29713 (0.00893)	-0.29517 (0.00937)	-0.2952 (0.0096)	-0.29367 (0.00734)	-0.44716 (0.00201)	-0.25459 (0.00194)	-0.09626 (0.00325)	-0.44141 (0.00234)	-0.07698 (0.00522)
Burr	c=1.2, k=2.5	-0.42448 (0.00037)	-0.42183 (0.00043)	-0.47303 (0.00028)	-0.44157 (0.00042)	-0.42996 (0.00041)	–	–	-0.43537 (0.00034)	-0.42395 (0.00048)	-0.38512 (0.00032)	-0.44722 (0.00047)	-0.00001 (0.00028)
	c=3, k=1	-0.13373 (0.00038)	-0.13133 (0.00033)	-0.21229 (0.00025)	-0.16418 (0.00039)	-0.14149 (0.00033)	–	–	-0.14875 (0.00029)	-0.13121 (0.0004)	-0.13885 (0.00027)	-0.21997 (0.00043)	-0.00007 (0.00028)
	c=1.3, k=1.5	-0.26929 (0.0004)	-0.24445 (0.00051)	-0.3402 (0.00065)	-0.25587 (0.00095)	-0.25164 (0.00093)	–	–	-0.28867 (0.00042)	-0.26643 (0.0004)	-0.28124 (0.00119)	-0.35955 (0.00046)	-0.00052 (0.00098)
	c=3, k=1.1/3	-0.32606 (0.00018)	-0.25541 (0.00027)	-0.04929 (0.00156)	-0.01242 (0.00175)	-0.02971 (0.00232)	–	–	-0.31335 (0.00031)	-0.31682 (0.00017)	-0.09939 (0.00451)	-0.36959 (0.00047)	-0.07376 (0.00531)
Lognormal	μ=0.5, σ=0.5	0.0257 (0.0001)	0.02641 (0.00012)	-0.06498 (0.00013)	-0.01368 (0.00024)	0.01991 (0.0001)	0.02518 (0.00013)	–	0.02021 (0.00009)	0.04399 (0.00011)	-0.02213 (0.00019)	-0.11001 (0.00011)	0.00006 (0.00017)
	μ=0.6, σ=1	-0.02159 (0.00043)	-0.01247 (0.00053)	-0.08004 (0.00075)	-0.02859 (0.0007)	-0.0379 (0.00066)	-0.03322 (0.00058)	–	-0.05189 (0.00048)	-0.00942 (0.00042)	-0.05519 (0.00151)	-0.12231 (0.00063)	0.00001 (0.00041)
	μ=1.5, σ=0.8	0.01526 (0.00019)	0.02081 (0.00023)	0.01438 (0.00034)	-0.01821 (0.00028)	-0.02376 (0.00028)	-0.0211 (0.00026)	–	-0.01886 (0.00021)	0.03844 (0.00018)	0.03878 (0.0011)	-0.0459 (0.00025)	-0.00001 (0.00018)

Note the following abbreviations, which are used to indicate the different methods:

Midpoint1: All of the observations are given the midpoint of the interval as a point value, and the observations in the last open interval are given the lower bound plus ten percent.

Midpoint2: All of the observations are given the midpoint of the interval as point value, and the observations in the last open interval are given the lower bound times two.

Par_0 to Par_4: This refers to the Pareto midpoint method. The number indicates how many of the first intervals are given the normal midpoint as point value. For example, the zero in Par_0 indicates that all of the intervals are given the Pareto midpoint value, and none are given the normal midpoint. Par_1 indicates that the observations in the first interval are given the normal midpoint, while the remaining intervals are given the Pareto midpoint. Par_3 indicates that the observations in the first three intervals are given the midpoint, while the remaining intervals are given the Pareto midpoint.

Logn_midp: This is the lognormal midpoint method. The observations in the first interval are given the normal midpoint, while the observations in the remaining intervals are given the lognormal midpoint.

Rand_midp: This indicates the random midpoint method, and was assigned to all observations in all intervals.

Rand_pareto: This refers to the random Pareto method, and was assigned to all observations in all intervals.

Rand_ln: This refers to the random lognormal method, and was assigned to all observations in all intervals.

Raw_data: This column refers to the raw data, and indicates the results that were obtained for the raw data, i.e. the simulated data. The measures of inequality were also calculated on the raw data per replicate. These measures were also divided into ten blocks as for the other methods, and the measures of performance were calculated on these inequality measurements.

For some distributions, no values are presented for Par_3 and Par_4. Whether Par_3 or Par_4 have values or not depends on the number of intervals into which the data was categorised. We could estimate the parameters of the distributions when there are three or more intervals. Thus, if Par_3 or Par_4 has a value, it means that the simulated data was divided into more intervals than when only Par_0 to Par_2 have values.

As mentioned above, the value in brackets in the table is the standard error of each estimate. One way to determine whether there is a significant difference between two entries is to make use of a confidence interval. An approximate normal confidence interval can be calculated using the following formula:

$$\hat{\theta} \pm 1.96se, \tag{5.2.8}$$

where $\hat{\theta}$ is an estimate, 'mean – true value' in the above table and se is the standard error. The 1.96 is used for a 95% confidence level.

There are two possible outcomes when considering the confidence intervals of two estimates. The first is when the interval bands of the two confidence intervals overlap. If this is the case, then the two estimates "do not differ significantly" from one another. The second case is when the interval bands of the two confidence intervals do not overlap, which indicates that the two estimates "differ significantly" from one another. We use this as a guide in distinguishing between estimates.

Let us consider three estimates obtained from the Burr distribution with parameters c=3 and k=1 from the table above.

Distribution	Parameters	midpoint1	midpoint2	rand_midp
Burr	c=3, k=1	-0.13373 (0.00038)	-0.13133 (0.00033)	-0.13121 (0.0004)

The confidence interval for midpoint1 is: $[-0.13447\,;\,-0.13299]$.

The confidence interval for midpoint2 is: $[-0.13198\,;\,-0.13068]$.

The confidence interval for rand_midp is: $[-0.13199\,;\,-0.13043]$.

It is clear that the confidence intervals of midpoint2 and random midpoint overlap. Thus, the estimates of these two methods do not differ significantly from one another. The confidence interval of midpoint1 does not overlap with either midpoint2 or random midpoint. Thus, the estimate of midpoint1 differs significantly from the estimates of midpoint2 and the random midpoint. Since the estimates of midpoint2 and random midpoint are better than the estimate of midpoint1, we can conclude that midpoint2 and random midpoint are significantly better than midpoint1 for the Burr distribution with $c = 3$ and $k = 1$.

Tables similar to that of Table 5.3.1 were obtained for each measure of inequality per sampling size per measure of performance. These tables are included in Appendix A.

Before we continue, let us consider the true values that were calculated for each distribution with the formulas derived in section 3.2.

Table 5.3.2: True values

	Parameters	EVI	Gini	QSR	Theil	Atk
Pareto	alpha=3	0.3333	0.2000	2.4742	0.0945	0.0400
	alpha=1.5	0.6667	0.5 000	8.1583	0.9014	0.2500
	alpha=1.1	0.9091	0.8333	43.0193	7.6021	0.6944
Burr	c=1.2, k=2.5	0.3333	0.5444	25.4560	0.5544	0.2497
	c=3, k=1	0.3333	0.3333	5.4551	0.2054	0.0931
	c=1.3, k=1.5	0.5128	0.6033	31.6520	0.8093	0.3120
	c=3, k=1.1/3	0.9091	0.8536	71.6374	7.7472	0.7157
Lognormal	mu=0.5, sigma=0.5	0	0.2763	4.0766	0.1250	0.0606
	mu=0.6, sigma=1	0	0.5205	17.1804	0.5 000	0.2212
	mu=1.5, sigma=0.8	0	0.4284	9.6038	0.3200	0.1479

The EVI (extreme value index) column is an indicator of how heavy the tail of the distribution is. A distribution with a higher EVI value has a heavier tail than a distribution with a smaller EVI value. The EVI for the Pareto distribution is calculated by

$$\gamma = \frac{1}{\alpha}, \tag{5.2.9}$$

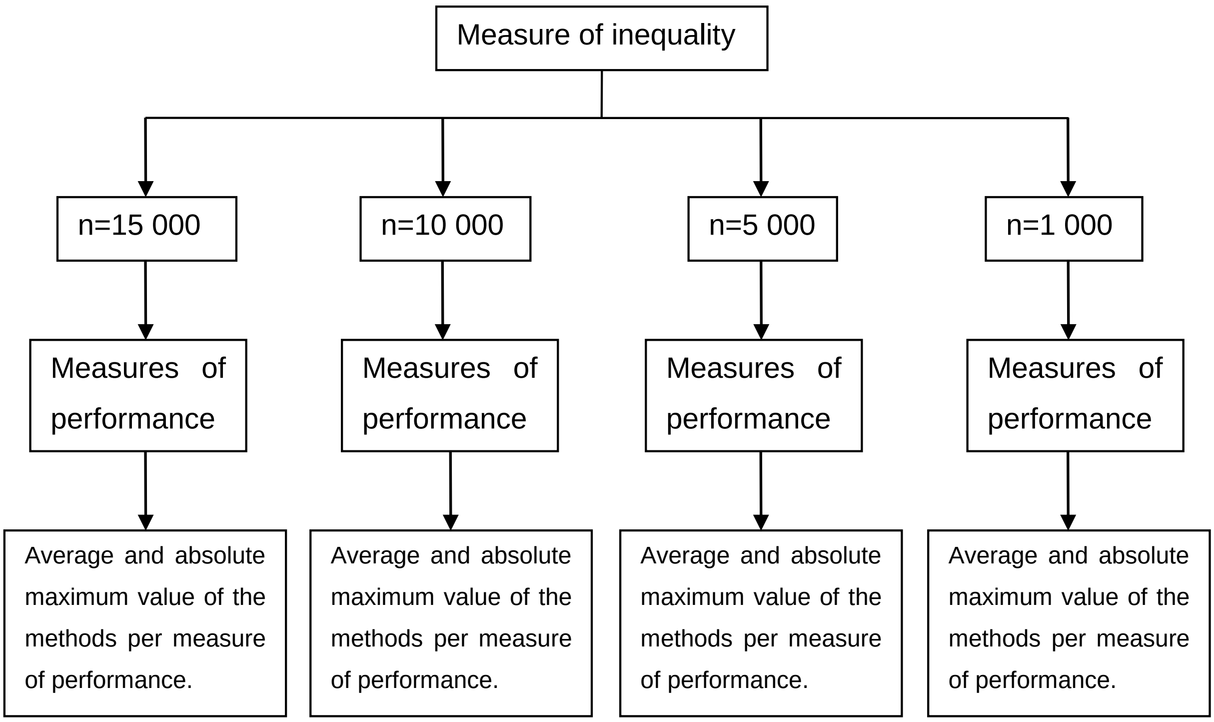
while the EVI for the Burr distribution is calculated by

$$\gamma = \frac{1}{ck}. \tag{5.2.9}$$

The EVI value for the lognormal distribution is always zero. From Table 5.3.2 it is clear that the true values increase as the EVI value becomes larger.

Next, we want to find a method of summarising the results, in order to obtain an overall picture of the performances of the different methods and distributions. To this end, let us study the average and absolute maximum values of each column (method) of each table (measurement of performance). Tables A1 – A64 in Appendix A contain the detailed results, which are the measures of performance obtained for each distribution per method. Tables 5.3.3 – 5.3.18 contain the summarised results of Tables A1 – A64, which are the average and absolute maximum value over the distributions for each method. In Tables 5.3.3 – 5.3.18 the smallest average and the minmax value for each measure of performance over the methods are indicated in bold. The minmax value is the minimum of the absolute maximum values. Thus, the absolute maximum, (i.e. the distribution that performed the worst in each method), is taken and the minimum of all the absolute maxima is determined. Thus, the minmax is the best of the worst methods. The smallest average is the method that, on average, performed best over all the distributions.

The following diagram indicates the order in which the summarised tables, containing the averages and the absolute maximum values, are presented and studied:



5.3.1 Results obtained for the Gini coefficient

The first measure of inequality that we consider is the Gini coefficient. The results obtained for the different *n* values and the different measures of performance per method are indicated in the tables below, with the minimum average and minmax values in bold text.

Table 5.3.3: Gini with *n*=15 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Gini	Mean	-0.18676	-0.17259	-0.16118	-0.16167	-0.15446	-0.13562	-0.31661	-0.20697	-0.14577	**-0.10440**	-0.25239
	|max|	0.42448	0.42183	0.47303	0.44157	0.42996	0.35375	**0.33955**	0.44716	0.42395	0.38512	0.44722
Median – True Gini	Mean	-0.18663	-0.17263	-0.16205	-0.16378	-0.15715	-0.14150	-0.33874	-0.20701	-0.14573	**-0.10681**	-0.25234
	|max|	0.42444	0.42180	0.47310	0.44170	0.43011	0.36202	**0.35769**	0.44681	0.42396	0.38528	0.44727
RMSE	Mean	0.19761	0.18522	0.16612	0.16453	0.16223	0.14923	0.32626	0.21128	0.16554	**0.11694**	0.25254
	|max|	0.42450	0.42186	0.47305	0.44160	0.42999	0.35514	**0.34448**	0.44778	0.42398	0.38514	0.44727
MAD	Mean	0.004109	0.004802	0.009963	0.012198	0.012111	0.016726	0.03734	0.004142	**0.003649**	0.008487	0.005118
	|max|	0.01607	0.01798	0.05353	0.07068	0.06871	0.06439	0.0595	0.01635	**0.01276**	0.02865	0.01775

(Table 5.3.3 is a summary of the results in Table A1 – Table A4 in Appendix A.)

Table 5.3.4: Gini with *n*=10 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Gini	Mean	-0.1877	-0.1732	-0.1615	-0.1620	-0.1548	-0.1367	-0.3151	-0.2076	-0.1465	**-0.1054**	-0.2531
	|max|	0.4250	0.4216	0.4730	0.4412	0.4293	0.3550	**0.3404**	0.4521	0.4245	0.3850	0.4476
Median – True Gini	Mean	-0.1875	-0.1732	-0.1637	-0.1661	-0.1597	-0.1470	-0.3487	-0.2076	-0.1465	**-0.1077**	-0.2530
	|max|	0.4249	0.4219	0.4732	0.4415	0.4297	0.3612	**0.3529**	0.4517	0.4243	0.3852	0.4479
RMSE	Mean	0.1986	0.1860	0.1667	0.1654	0.1696	0.1945	0.3235	0.2120	0.1663	**0.1181**	0.2533
	|max|	0.4250	0.4216	0.4730	0.4412	0.4931	0.5749	**0.3441**	0.4530	0.4246	0.3850	0.4476
MAD	Mean	0.0055	0.0061	0.0099	0.0118	0.0118	0.0137	0.0241	0.0052	**0.0048**	0.0094	0.0064
	|max|	0.0238	0.0243	0.0442	0.0588	0.0565	0.0478	0.0317	0.0209	**0.0188**	0.0311	0.0225

(Table 5.3.4 is a summary of the results in Table A5 – Table A8 in Appendix A.)

Table 5.3.5: Gini with *n*=5 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Gini	Mean	-0.1893	-0.1738	-0.1595	-0.1587	-0.1510	-0.1287	-0.2660	-0.2083	-0.1479	**-0.1065**	-0.2541
	|max|	0.4256	0.4206	0.4725	0.4397	0.4268	0.3424	**0.3262**	0.4582	0.4252	0.3843	0.4526
Median – True Gini	Mean	-0.1895	-0.1739	-0.1575	-0.1570	-0.1502	-0.1273	-0.2837	-0.2082	-0.1481	**-0.1100**	-0.2540
	|max|	0.4255	0.4205	0.4726	0.4401	0.4274	0.3532	**0.3426**	0.4560	0.4252	0.3846	0.4506
RMSE	Mean	0.2003	0.1872	0.1663	0.1642	0.3396	0.2620	0.2720	0.2130	0.1678	**0.1210**	0.2546
	|max|	0.4257	0.4207	0.4725	0.4398	1.6497	0.9313	**0.3297**	0.4600	0.4253	0.3844	0.4547
MAD	Mean	0.0078	0.0086	0.0134	0.0177	0.0183	0.0247	0.0217	0.0074	**0.0069**	0.0119	0.0090
	|max|	0.0347	0.0366	0.0591	0.0957	0.0982	0.0971	0.0335	0.0314	**0.0271**	0.0348	0.0334

(Table 5.3.5 is a summary of the results in Table A9 – Table A12 in Appendix A.)

Table 5.3.6: Gini with *n*=1 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Gini	Mean	-0.2005	-0.1805	-0.1617	-0.1629	-0.1417	**-0.0767**	–	-0.2155	-0.1567	-0.1120	-0.2642
	|max|	0.4814	0.4523	0.4683	0.4306	0.3918	**0.2672**	–	0.5221	0.4242	0.3801	0.5298
Median – True Gini	Mean	-0.2009	-0.1815	-0.1682	-0.1740	-0.1597	**-0.0811**	–	-0.2164	-0.1572	-0.1186	-0.2657
	|max|	0.4843	0.4577	0.4695	0.4332	0.4334	**0.2720**	–	0.5289	0.4249	0.3825	0.5338
RMSE	Mean	0.2123	0.1979	0.1792	0.1751	0.2312	0.1446	–	0.2218	0.1773	**0.1362**	0.2654
	|max|	0.4846	0.4558	0.4684	0.4312	0.9746	0.5263	–	0.5245	0.4246	**0.3805**	0.5330
MAD	Mean	0.0125	0.0137	0.0238	0.0259	0.0234	0.0149	–	0.0118	**0.0115**	0.0196	0.0149
	|max|	0.0372	0.0342	0.0728	0.0833	0.0570	0.0301	–	0.0304	**0.0291**	0.0481	0.0363

(Table 5.3.6 is a summary of the results in Table A13 – Table A16 in Appendix A.)

For *n*=15 000, *n*=10 000 and *n*=5 000, the same methods have the smallest average and the minmax values. According to both bias measures, mean – true Gini and median – true Gini, and RMSE for *n*=15 000, *n*=10 000 and *n*=5 000, random Pareto has the smallest average of all the methods, and Par_4 has the minmax value. For *n*=1 000, the smallest averages and the minmax value for both biases are at Par_3. For the RMSE, the random Pareto method has the smallest average as well as the minmax value. For *n*=1 000, Par_4 contains no values because after simulating a mere 1 000 values, none of the values were large enough to fall in the last interval. When we had simulated 5 000, 10 000 and 15 000 values, there were values large enough to be categorised into the last interval, and we were able to perform calculations on these values. For the MAD, all the minimum values for all *n* were at random midpoint. The MAD is a more robust method and is thus not affected by outliers as much as the RMSE.

For *n*=15 000 and *n*=10 000, the method with the largest average for all measures of performance is Par_4, which is the method that also has the most minmax values. For *n*=5 000, the largest averages for both the biases are also at Par_4, but for the RMSE and the MAD the largest averages is at Par_2 and Par_3. For *n*=1 000, the largest average is at random lognormal for both biases and RMSE, while the Par_1 contains the largest average for the MAD. All the averages for both biases and all *n* values have a negative sign. This indicates that all the methods on average had under-predicted the true value of the Gini coefficient, meaning that the predicated value of the Gini coefficient on average was smaller than the true value of the Gini coefficient.

5.3.2 Results obtained for the QSR

The next measure of inequality for considerion is the QSR. The results obtained for the different *n* values and the different measures of performance per method are indicated in the tables below, with the minimum average and minmax values in bold text.

Table 5.3.7: QSR with *n*=15 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True QSR	Mean	-13.6062	-12.6445	-6.1221	**-4.0724**	-5.3071	-5.8190	-12.8431	-14.4845	-14.4951	-12.2546	-17.8938
	|max|	57.0252	50.3081	28.0803	38.7634	26.5928	**24.8069**	24.9239	56.0562	61.1310	43.4508	62.9493
Median – True QSR	Mean	-13.6076	-12.6524	-6.5893	**-4.9864**	-6.4210	-6.5264	-14.9813	-14.4906	-14.4988	-12.6685	-17.8958
	|max|	57.0250	50.3076	28.0820	32.9384	**27.8165**	27.9110	28.0687	56.0577	61.1323	45.6347	62.9491
RMSE	Mean	14.9646	14.1047	9.5299	13.4466	12.6037	**6.7830**	14.9935	15.2098	15.2736	13.1241	17.8950
	|max|	57.0252	50.3088	28.0805	47.0467	40.1679	26.5658	**26.3830**	56.0565	61.1312	44.2298	62.9496
MAD	Mean	0.1331	0.1826	1.4808	1.9339	1.8317	0.8192	1.7657	0.1224	0.1348	0.6925	**0.0944**
	|max|	0.6725	0.8124	7.5590	14.4524	13.7022	3.2198	2.9442	0.5156	0.7743	3.6897	**0.4202**

(Table 5.3.7 is a summary of the results in Table A17 – Table A20 in Appendix A.)

Table 5.3.8: QSR with *n*=10 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True QSR	Mean	-13.6380	-12.6678	-6.0584	**-3.7886**	-4.8502	-5.7695	-12.9106	-14.5003	-14.6968	-12.2136	-17.8181
	|max|	57.0227	50.2992	28.0729	42.2674	29.9039	**24.5429**	24.9226	56.0499	61.1237	43.4555	62.9402
Median – True QSR	Mean	-13.6426	-12.6804	-6.8206	**-5.0105**	-6.4320	-6.8148	-15.5037	-14.5091	-14.6993	-12.6669	-17.8246
	|max|	57.0237	50.3027	**28.0783**	33.9311	29.1610	29.3744	29.3540	56.0519	61.1190	46.0632	62.9431
RMSE	Mean	14.9981	14.1337	9.7421	14.0225	14.0931	**7.0532**	14.4540	15.2267	15.3056	13.2404	17.9059
	|max|	57.0228	50.3002	28.0732	55.1159	51.8689	27.7270	**26.0258**	56.0503	61.1239	44.5465	62.9406
MAD	Mean	0.1815	0.2259	1.4303	2.0942	2.0420	0.6549	1.1563	0.1524	0.1844	0.7298	**0.1160**
	|max|	0.9968	1.0684	7.2489	16.8263	16.5660	2.3225	1.6329	0.6786	1.1467	3.8312	**0.5282**

(Table 5.3.8 is a summary of the results in Table A21 – Table A24 in Appendix A.)

Table 5.3.9: QSR with *n*=5 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True QSR	Mean	-13.6949	-12.6905	-5.5359	**-2.8502**	-2.8988	-5.1005	-9.6082	-14.5218	-14.5843	-12.2859	-17.9201
	|max|	57.0230	50.3003	28.0779	48.5456	46.1752	22.1094	**18.9191**	56.0564	61.1459	44.4177	62.9650
Median – True QSR	Mean	-13.2210	-12.2937	-5.8040	-4.1172	-5.5659	**-4.0540**	-9.4201	-13.9436	-14.2299	-12.7936	-16.7663
	|max|	57.0215	50.2948	28.0892	33.2801	26.6195	24.0647	**21.7039**	56.0562	61.1456	47.8804	62.9596
RMSE	Mean	15.0623	14.1844	10.3807	15.8434	19.4737	**7.3537**	11.4043	15.2535	15.3663	13.4756	17.9233
	|max|	57.0231	50.3022	28.0785	74.4790	108.5270	25.9672	**20.1349**	56.0573	61.1463	46.0177	62.9658
MAD	Mean	0.2523	0.3313	2.3964	3.1947	2.9797	1.5187	1.6528	0.2209	0.2531	0.8465	**0.1639**
	|max|	1.3499	1.6066	11.6493	23.6649	21.6484	6.3319	2.9074	1.0041	1.5372	3.9592	**0.7566**

(Table 5.3.9 is a summary of the results in Table A25 – Table A28 in Appendix A.)

Table 5.3.10: QSR with *n*=1 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True QSR	Mean	-14.2355	-13.1121	-5.7210	2.5951	**-0.0221**	-1.0496	–	-14.8684	-14.9685	-12.7217	-18.1032
	|max|	57.5406	51.3260	28.0451	88.7265	73.3152	**5.3241**	–	56.7433	61.2288	46.7339	63.1166
Median – True QSR	Mean	-14.1477	-13.0506	-8.1372	-7.0649	-8.1845	**-1.9282**	–	-14.8229	-15.0122	-13.8263	-18.1430
	|max|	57.1068	50.5995	28.0997	32.7611	32.9194	**9.4320**	–	56.2452	61.2256	51.9218	63.1192
RMSE	Mean	15.6262	14.8526	15.3945	59.3584	36.8205	**4.3809**	–	15.6436	15.7343	14.9713	18.1115
	|max|	57.5576	51.3906	42.6559	397.0480	275.1180	**12.0243**	–	56.7688	61.2306	50.6096	63.1194
MAD	Mean	0.3492	0.4565	2.9301	4.1637	3.5624	1.6825	–	0.3185	0.3326	0.9903	**0.2369**
	|max|	1.0394	1.0195	18.0424	34.6911	29.7980	5.5693	–	0.7215	1.2638	4.0987	**0.6153**

(Table 5.3.10 is a summary of the results in Table A29 – Table A32 in Appendix A.)

The smallest average for both biases for *n*=15 000 and *n*=10 000 is at Par_1. For *n*=5 000, the smallest average for the mean – true QSR is also at Par_1, but the smallest average for the median – true QSR is at Par_3. For *n*=1 000, the smallest average for mean – true QSR is at Par_2, while the smallest average for the median – true QSR is at Par_3.

The minmax values for the biases for the different *n* values differ between the different Pareto midpoint methods. For *n*=15 000 and *n*=1 000, the minmax values for both biases are at Par_3.

The minmax value for both biases for *n*=5 000 is at Par_4, while the minmax values for *n*= 10 000 are at Par_2 for mean – true QSR and at Par_0 for median – true QSR.

For the Gini coefficient, all the minimum values for the MAD are at the random midpoint method, while all the minimum values for the QSR for all *n* values are at the random lognormal method. For the RMSE at *n*=15 000, *n*=10 000 and *n*=5 000, the smallest average is at Par_3 and the minmax values are at Par_4. For *n*=1 000, both the smallest average and the minmax value are at Par_3 for the RMSE.

All the values of the biases are negative except for the Par_1 method at *n*=1 000. This indicates that the QSR was also under predicted by all the methods except for Par_1 at *n*=1 000 where the QSR was over predicted.

For both the biases for all *n* values the largest average is at the random lognormal method. For the MAD the smallest average is also at the random lognormal method for all n. For the MAD the largest average is at Par_1 for all *n* values. The largest average for the RMSE for both *n*= 15 000 and *n*=10 000 is also at random lognormal while the largest average for *n*=5 000 and *n*=1 000 is at Par_2 and Par_1.

5.3.3 Results obtained for the Theil measure

Next we consider the results obtained for the Theil measure. The results obtained for the different *n* values and the different measures of performance per method are indicated in the tables below with the minimum average and minmax values in bold text.

Table 5.3.11: Theil with *n*=15 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Theil	**Mean**	-1.5139	-1.4563	-1.1587	-1.2429	-1.2587	-1.2073	-3.0055	-1.5644	-1.4886	**-1.1537**	-1.6707
	|max|	7.2202	6.9963	5.8624	5.6910	5.8079	5.4527	5.5007	7.1725	7.2087	**5.2973**	7.2745
Median – True Theil	**Mean**	-1.5133	-1.4568	-1.1673	**-1.2641**	-1.2873	-1.2722	-3.2437	-1.5648	-1.4879	-1.2115	-1.6711
	|max|	7.2204	6.9965	5.8623	5.6997	5.8241	5.7204	5.8398	7.1729	7.2089	**5.5114**	7.2749
RMSE	**Mean**	1.5480	1.5021	**1.1847**	1.2682	1.8721	1.2557	3.1168	1.5837	1.5304	1.2278	1.6709
	|max|	7.2202	6.9963	5.8631	5.6923	6.2444	5.5486	5.5937	7.1725	7.2087	**5.3392**	7.2745
MAD	**Mean**	0.0131	0.0189	0.0946	0.0886	0.1653	0.1394	0.3176	0.0133	0.0135	0.0865	**0.0121**
	|max|	0.0836	0.1141	0.7592	0.6893	0.7758	0.6254	0.5695	0.0846	0.0866	0.4422	**0.0772**

(Table 5.3.11 is a summary of the results in Table A33 – Table A36 in Appendix A.)

Table 5.3.12: Theil with *n*=10 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Theil	**Mean**	-1.5193	-1.4610	**-1.1650**	-1.2485	-1.2644	-1.2190	-2.9969	-1.5677	-1.4937	-1.1706	-1.6734
	|max|	7.2204	6.9964	5.8586	5.6863	5.7988	5.5063	5.4726	7.1725	7.2088	**5.3686**	7.2739
Median – True Theil	**Mean**	-1.5170	-1.4612	**-1.2001**	-1.2947	-1.3170	-1.3344	-3.3445	-1.5680	-1.4926	-1.2299	-1.6735
	|max|	7.2205	6.9964	5.8582	5.7629	5.9000	6.0272	6.0801	7.1727	7.2089	**5.5687**	7.2741
RMSE	**Mean**	1.5535	1.5074	**1.1911**	1.2748	1.8448	1.2701	3.0811	1.5872	1.5357	1.2406	1.6736
	|max|	7.2204	6.9964	5.8595	5.6882	5.9068	5.6068	5.5533	7.1725	7.2088	**5.4129**	7.2739
MAD	**Mean**	0.0195	0.0238	0.0675	0.0666	0.0811	0.0918	0.1741	0.0168	0.0205	0.0887	**0.0149**
	|max|	0.1350	0.1525	0.4672	0.4476	0.4213	0.3862	0.2716	0.1097	0.1442	0.4425	**0.0957**

(Table 5.3.12 is a summary of the results in Table A37 – Table A40 in Appendix A.)

Table 5.3.13: Theil with *n*=5 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Theil	**Mean**	-1.5289	-1.4673	**-1.1485**	-1.2244	-1.2365	-1.1571	-2.5565	-1.5724	-1.5036	-1.1929	-1.6772
	|max|	7.2207	6.9970	5.8654	5.6909	5.7923	5.2637	**4.6523**	7.1732	7.2094	5.4657	7.2754
Median – True Theil	**Mean**	-1.5347	-1.4730	**-1.1383**	-1.2275	-1.2472	-1.1618	-2.7183	-1.5762	-1.5087	-1.2617	-1.6799
	|max|	7.2207	6.9967	5.8680	5.7031	5.8327	5.2153	**4.8604**	7.1728	7.2093	5.7225	7.2752
RMSE	**Mean**	1.5633	1.5152	**1.1829**	1.2589	1.7436	1.2364	2.6325	1.5923	1.5460	1.2685	1.6777
	|max|	7.2207	6.9970	5.8671	5.6944	5.7985	5.3907	**4.7189**	7.1732	7.2094	5.5395	7.2754
MAD	**Mean**	0.0268	0.0357	0.1171	0.1265	0.1311	0.2126	0.2627	0.0249	0.0266	0.0983	**0.0219**
	|max|	0.1814	0.2373	0.8920	0.9654	0.9742	0.9702	0.4892	0.1673	0.1767	0.4271	**0.1456**

(Table 5.3.13 is a summary of the results in Table A41 – Table A44 in Appendix A.)

Table 5.3.14: Theil with *n*=1 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Theil	**Mean**	-1.5809	-1.5123	-1.2348	-1.3041	-1.2438	**-0.8594**	–	-1.6076	-1.5549	-1.2875	-1.7087
	|max|	7.2245	7.0054	5.9327	5.9819	5.8975	**4.0976**	–	7.1809	7.2130	5.9488	7.2851
Median – True Theil	**Mean**	-1.5823	-1.5190	-1.3003	-1.3876	-1.3758	**-0.9164**	–	-1.6124	-1.5562	-1.3791	-1.7140
	|max|	7.2239	7.0031	5.9362	6.5993	6.6543	**4.2794**	–	7.1796	7.2125	6.2201	7.2857
RMSE	**Mean**	1.6162	1.5713	1.3034	1.5435	1.4283	**0.9626**	–	1.6298	1.5980	1.4232	1.7098
	|max|	7.2246	7.0054	5.9386	6.0982	6.0215	**4.1175**	–	7.1809	7.2130	6.0152	7.2851
MAD	**Mean**	0.0258	0.0340	0.0942	0.0981	0.0885	0.1077	–	0.0230	0.0266	0.0976	**0.0210**
	|max|	0.1141	0.1211	0.4012	0.3651	0.2936	0.3564	–	0.0869	0.1166	0.3903	**0.0817**

(Table 5.3.14 is a summary of the results in Table A45 – Table A48 in Appendix A.)

For *n*=15 000, the smallest average and the minmax value for mean – true Theil is obtained with the random Pareto method. For median – true Theil, the smallest average is at Par_1, while the minmax value is also at random Pareto. For *n*=10 000 and *n*=5 000, the smallest average for both biases is at Par_0. The minmax values for both biases for *n*=10 000 are at random Pareto, the same as for *n*=15 000, while the minmax value for *n*=5 000 is at Par_4. For *n*=1 000, both the smallest average value, and the minmax values for both biases are found at Par_3.

For *n*=15 000, *n*=10 000 and *n*=5 000, the smallest average for RMSE is at Par_0. The minmax values for RMSE for *n*=15 000 and *n*=10 000 are at random Pareto, while the minmax value for *n*=5 000 is at Par_4. For *n*=1 000, the smallest average and the minmax value are at Par_3. The MAD has the same results as for the QSR. The minimum average and minmax values for all *n* are found at the random lognormal method.

The largest average value for all measures of performance for *n*=15 000, *n*=10 000 and *n*=5 000 is at Par_4. For *n*=1 000, the largest average for both biases and RMSE is at the random lognormal method, while the largest average for the MAD is at Par_3. All the averages of the biases under-predicted the Theil value, and thus have negative signs.

5.3.4 Results obtained for the Atkinson measure

Finally, we consider the results obtained for the Atkinson measure. The results obtained for the different *n* values and the different measures of performance per method are indicated in the tables below with the minimum average and minmax values in bold text.

Table 5.3.15: Atkinson with *n*=15 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Atk	Mean	-0.1442	-0.1273	**-0.0805**	-0.0925	-0.0914	-0.0875	-0.2207	-0.1613	-0.1288	-0.0829	-0.1916
	|max|	0.4838	0.4119	0.2295	0.2744	0.2805	0.2856	0.2864	0.4707	0.4798	**0.2210**	0.5213
Median – True Atk	Mean	-0.1441	-0.1274	-0.0815	-0.0952	-0.0949	-0.0948	-0.2483	-0.1613	-0.1287	**-0.0875**	-0.1917
	|max|	0.4837	0.4120	0.2297	0.2931	0.3044	0.3127	0.3210	0.4708	0.4798	**0.2213**	0.5213
RMSE	Mean	0.1556	0.1431	**0.0912**	0.0994	1.6705	3.0402	3.1970	0.1643	0.1466	0.0964	0.1973
	|max|	0.4837	0.4119	0.2295	0.2975	6.3147	8.5927	6.0855	0.4707	0.4798	**0.2210**	0.5212
MAD	Mean	0.0032	0.0041	0.0120	0.0137	0.0631	0.2452	0.6205	**0.0032**	0.0032	0.0114	0.0032
	|max|	0.0164	0.0193	0.0739	0.0839	0.1966	1.1039	1.1692	**0.0160**	0.0161	0.0485	0.0164

(Table 5.3.15 is a summary of the results in Table A49 – Table A52 in Appendix A.)

Table 5.3.16: Atkinson with *n*=10 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Atk	Mean	-0.1415	-0.1268	**-0.0835**	-0.0922	-0.0927	-0.0924	-0.2191	-0.1563	-0.1297	-0.0846	-0.1977
	|max|	0.4838	0.4119	0.2296	0.2799	0.2869	0.2908	0.2820	0.4707	0.4798	**0.2210**	0.5211
Median – True Atk	Mean	-0.1412	-0.1268	**-0.0867**	-0.0975	-0.0991	-0.1056	-0.2603	-0.1564	-0.1295	-0.0890	-0.1978
	|max|	0.4838	0.4119	0.2298	0.3249	0.3389	0.3488	0.3519	0.4707	0.4798	**0.2214**	0.5210
RMSE	Mean	0.1566	0.1441	**0.0918**	0.1004	1.6819	3.0479	3.3020	0.1650	0.1475	0.0982	0.1978
	|max|	0.4837	0.4119	0.2296	0.3036	6.3446	8.6011	6.3021	0.4707	0.4798	**0.2210**	0.5210
MAD	Mean	0.0045	0.0051	0.0110	0.0125	0.0714	0.2539	0.7599	0.0040	0.0045	0.0123	**0.0040**
	|max|	0.0251	0.0261	0.0566	0.0654	0.2232	1.1586	1.4847	0.0210	0.0255	0.0511	**0.0209**

(Table 5.3.16 is a summary of the results in Table A53 – Table A56 in Appendix A.)

Table 5.3.17: Atkinson with *n*=5 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Atk	Mean	-0.1433	-0.1276	**-0.0810**	-0.0886	-0.0884	-0.0835	-0.1624	-0.1571	-0.1314	-0.0866	-0.1985
	|max|	0.4840	0.4122	0.2291	0.2494	0.2564	0.2580	**0.1812**	0.4710	0.4801	0.2206	0.5216
Median – True Atk	Mean	-0.1438	-0.1281	**-0.0784**	-0.0873	-0.0885	-0.0819	-0.1825	-0.1574	-0.1319	-0.0929	-0.1987
	|max|	0.4839	0.4121	0.2294	0.2248	0.2374	0.2386	**0.2037**	0.4709	0.4801	0.2210	0.5213
RMSE	Mean	0.1584	0.1455	**0.0922**	0.1 000	3.1878	3.1079	3.5972	0.1660	0.1493	0.1024	0.1987
	|max|	0.4840	0.4122	0.2292	0.2881	8.3445	8.6228	6.9975	0.4710	0.4801	**0.2206**	0.5216
MAD	Mean	0.0064	0.0076	0.0155	0.0198	0.2484	0.3447	0.3408	0.0059	0.0063	0.0151	**0.0058**
	|max|	0.0359	0.0408	0.0805	0.1139	1.4632	1.5247	0.6354	0.0324	0.0347	0.0563	**0.0314**

(Table 5.3.16 is a summary of the results in Table A57 – Table A60 in Appendix A.)

Table 5.3.18: Atkinson with *n*=1 000

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_par	rand_ln
Mean – True Atk	Mean	-0.1543	-0.1354	-0.0858	-0.0954	-0.0826	**-0.0292**	–	-0.1644	-0.1446	-0.0956	-0.2043
	|max|	0.5009	0.4711	0.2266	0.3365	0.3216	**0.0931**	–	0.5326	0.4816	0.2500	0.5775
Median – True Atk	Mean	-0.1549	-0.1368	-0.0956	-0.1098	-0.1047	**-0.0364**	–	-0.1655	-0.1452	-0.1070	-0.2057
	|max|	0.5022	0.4768	0.2281	0.4328	0.4466	**0.1062**	–	0.5386	0.4814	0.2784	0.5831
RMSE	Mean	0.1696	0.1565	**0.1078**	0.5381	3.3846	1.8349	–	0.1742	0.1600	0.1245	0.2065
	|max|	0.5031	0.4738	**0.2267**	4.3977	8.4915	8.8595	–	0.5342	0.4815	0.2792	0.5791
MAD	Mean	0.0083	0.0099	0.0234	0.2211	0.3615	0.0483	–	0.0075	0.0083	0.0209	**0.0073**
	|max|	0.0315	0.0298	0.0801	1.9879	2.3496	0.1607	–	**0.0237**	0.0301	0.0690	0.0243

(Table 5.3.16 is a summary of the results in Table A61 – Table A64 in Appendix A.)

For *n*=15 000, *n*=10 000 and *n*=5 000, the smallest average for the mean – true Atkinson is at Par_0. The smallest average for *n*=15 000 for median – true Atkinson is at random Pareto, while the smallest average for median – true Atkinson for *n*=10 000 and *n*=5 000 is at Par_0. For *n*= 1 000, the smallest average for both biases is at Par_3. The minmax values for *n*=15 000 and *n*=10 000 for both biases are at random Pareto, while the minmax values for both biases for *n*= 5 000 are at Par_4, and for *n*=1 000 at Par_3.

The smallest average for all *n* values for the RMSE is at Par_0, and the minmax values for *n*= 15 000, *n*=10 000 and *n*=5 000 are at random Pareto, while the minmax value for *n*=1 000 is at Par_0. For the MAD, all the smallest average and minmax values are at random midpoint method for the Gini coefficient, and all the smallest average and minmax values are at random lognormal for the QSR and Theil measure. For the Atkinson measure, the smallest average and minmax value for *n*=15 000 are at lognormal midpoint. For *n*=10 000 and *n*=5 000, the smallest average and minmax values are at the random lognormal method, while the smallest average for *n*=1 000 is at random lognormal method, and the minmax value at the lognormal midpoint method.

For *n*=15 000 and *n*=10 000, the largest average values for all the measures of performance are at Par_4. For *n*=5 000 and *n*=1 000, the largest average values for both biases are at random lognormal method. The largest average for *n*=5 000 for RMSE is at Par_4, and the largest average for the MAD is at Par_3. For *n*=1 000, the largest average for the RMSE and the MAD is at Par_2.

For the Atkinson measure, all the signs of both biases are negative. For the Gini coefficient, the QSR, the Theil and the Atkinson measures the signs of all biases, except for the sign of Par_1 for the QSR with *n*=1 000, are negative. This means that the predicted values of all the methods for all the measures of inequality are smaller than the true value, and that the methods tend to under-predict the true values.

5.3.5 Summary of the methods with the minima

The following tables summarise the methods with the smallest average and the minmax values in each of the tables above (the bold text values in each of the tables, Table 5.3.3 – 5.3.18). The first table contains the methods with the smallest average.

Table 5.3.19: Method with smallest average

		Gini	QSR	Theil	Atk
n=15 000	**Mean − True Gini**	Rand_pareto	Par_1	Rand_pareto	Par_0
	Median − True Gini	Rand_pareto	Par_1	Par_1	Rand_pareto
	RMSE	Rand_pareto	Par_3	Par_0	Par_0
	MAD	rand_midp	rand_ln	rand_ln	logN_midp
n=10 000	**Mean − True Gini**	Rand_pareto	Par_1	Par_0	Par_0
	Median − True Gini	Rand_pareto	Par_1	Par_0	Par_0
	RMSE	Rand_pareto	Par_3	Par_0	Par_0
	MAD	rand_midp	rand_ln	rand_ln	rand_ln
n=5 000	**Mean − True Gini**	Rand_pareto	Par_1	Par_0	Par_0
	Median − True Gini	Rand_pareto	Par_3	Par_0	Par_0
	RMSE	Rand_pareto	Par_3	Par_0	Par_0
	MAD	rand_midp	rand_ln	rand_ln	rand_ln
n=1 000	**Mean − True Gini**	Par_3	Par_2	Par_3	Par_3
	Median − True Gini	Par_3	Par_3	Par_3	Par_3
	RMSE	Rand_pareto	Par_3	Par_3	Par_0
	MAD	rand_midp	rand_ln	rand_ln	rand_ln

From the above table, it is clear that the random Pareto method contained the smallest average most of the time for the Gini coefficient for all *n*, except *n*=1 000. For the QSR, Par_1 and Par_3 contained the smallest average most of the time. For large *n*, *n*=15 000 and *n*=10 000, Par_1 contained the smallest average more than Par_3, but for smaller *n*, Par_3 contained the smallest average more than did Par_1.

For the Theil- and Atkinson measures, Par_0 contained the smallest average most of time. For the Theil measure, a different method contained the smallest average for every measure of performance at *n*=15 000. For *n*=10 000 and *n*=5 000, Par_0 contained the smallest average for both biases and the RMSE, while the smallest average for the MAD is at the random lognormal method. For *n*=1 000, Par_3 contained the smallest average most of the time. For the Atkinson measure, Par_0 contained the smallest average most of the time for *n*=15 000, *n*=10 000 and *n*=5 000, but Par_3 contained the smallest average most of the time for *n*=1 000.

Overall, the Pareto methods, Pareto midpoint and random Pareto contained the smallest average more than the other methods. For the two biases and RMSE, a Pareto method contained the smallest average every time. For MAD, the lognormal methods contained the smallest average most of the time. Except for the Gini coefficient, the lognormal methods contained all the smallest averages for the MAD.

The table on the following page contains the methods with the minimum of the maximum value over all the distributions, the method with the minmax value of the tables above.

Table 5.3.20: Method with best minmax

		Gini	QSR	Theil	Atk
n=15 000	**Mean − True Gini**	Par_4	Par_2	Rand_pareto	Rand_pareto
	Median − True Gini	Par_4	Par_2	Rand_pareto	Rand_pareto
	RMSE	Par_4	Par_4	Rand_pareto	Rand_pareto
	MAD	rand_midp	rand_ln	rand_ln	logN_midp
n=10 000	**Mean − True Gini**	Par_4	Par_3	Rand_pareto	Rand_pareto
	Median − True Gini	Par_4	Par_0	Rand_pareto	Rand_pareto
	RMSE	Par_4	Par_4	Rand_pareto	Rand_pareto
	MAD	rand_midp	rand_ln	rand_ln	rand_ln
n=5 000	**Mean − True Gini**	Par_4	Par_4	Par_4	Par_4
	Median − True Gini	Par_4	Par_4	Par_4	Par_4
	RMSE	Par_4	Par_4	Par_4	Rand_pareto
	MAD	rand_midp	rand_ln	rand_ln	rand_ln
n=1 000	**Mean − True Gini**	Par_3	Par_3	Par_3	Par_3
	Median − True Gini	Par_3	Par_3	Par_3	Par_3
	RMSE	Rand_pareto	Par_3	Par_3	Par_0
	MAD	rand_midp	rand_ln	rand_ln	logN_midp

For the Gini coefficient, it is clear that Par_4 contained the minmax values most of the time. For both biases and the RMSE, Par_4 contained the minmax value for all *n* except *n*=1 000, where Par_3 contained the minmax value most of the time. For the QSR with *n*=15 000 and *n*=10 000, a different method contained the minmax value for each *n* for each measure of performance. Over all the *n* values, Par_3 and Par_4 contained the minmax value most of the time.

For the Theil- and Atkinson measures, random Pareto contained the minmax values most of the time. For both measurements for *n*=15 000 and *n*=10 000, random Pareto contained the minmax value most of the time. For *n*=5 000, Par_4 contained the minmax value most of the time and for *n*=1 000, Par_3 contained the minmax for both measurements most of the time.

Overall, the Pareto methods also contained the minmax values more than the other methods. For the two biases and RMSE, a Pareto method contained the minmax every time. For MAD, the lognormal methods contained the minmax most the time. Except for the Gini coefficient, the lognormal methods contained all the minmax values for the MAD. The same results were obtained for smallest averages and minmax values for the MAD, at Atkinson with *n*=1 000.

5.3.5 Raw data

Next, we consider the results that were obtained for the raw data, i.e. the simulated data. The following tables contain the average and the absolute maximum values of the raw data of all the distributions simulated from. The full results obtained for the measures of inequality per sample size for each measure of performance of the raw data appear in Appendix A in Tables A1 –A64 under column name "raw_data".

Table 5.3.21: Raw data

		Gini				QSR			
		n=15 000	n=10 000	n=5 000	n=1 000	n=15 000	n=10 000	n=5 000	n=1 000
Mean – True Gini	Mean	-0.0155	-0.0160	-0.0180	-0.0252	-2.6711	1.8180	-2.4053	-4.0786
	|max|	0.0770	0.0843	0.0921	0.1217	19.1564	25.9298	13.4033	26.4478
Median – True Gini	Mean	-0.0185	-0.0189	-0.0220	-0.0294	-4.7553	-4.7618	-5.0580	-6.1799
	|max|	0.0908	0.0950	0.1047	0.1374	29.5717	29.1398	32.1811	38.4007
RMSE	Mean	0.0240	0.0255	0.0298	0.0419	12.6099	55.9096	14.9584	10.8329
	|max|	0.0956	0.1016	0.1126	0.1469	63.6937	480.7240	98.5723	61.8498
MAD	Mean	0.0086	0.0094	0.0118	0.0173	1.1378	1.2370	1.4085	1.7574
	|max|	0.0302	0.0298	0.0365	0.0501	5.7994	6.5149	6.8826	7.0151

		Theil				Atkinson			
		n=15 000	n=10 000	n=5 000	n=1 000	n=15 000	n=10 000	n=5 000	n=1 000
Mean – True Gini	Mean	-1.1160	-1.0155	-1.0547	-1.1683	-0.0287	-0.0295	-0.0329	-0.0445
	|max|	5.0634	5.0678	5.2262	5.8619	0.1413	0.1524	0.1654	0.2186
Median – True Gini	Mean	-1.0847	-1.0989	-1.1439	-1.2422	-0.0343	-0.0353	-0.0402	-0.0527
	|max|	5.3677	5.4009	5.6395	6.1341	0.1663	0.1749	0.1906	0.2478
RMSE	Mean	1.0734	1.0844	1.1291	1.2452	0.0406	0.0425	0.0485	0.0641
	|max|	5.1786	5.1897	5.3627	5.9339	0.1717	0.1814	0.1988	0.2514
MAD	Mean	0.1039	0.1071	0.1112	0.1114	0.0129	0.0138	0.0164	0.0217
	|max|	0.4573	0.4748	0.4532	0.4386	0.0510	0.0516	0.0584	0.0763

When we consider the results obtained for the raw data and compare it to the other methods used, we expect that the raw data will be closer to the true values than the other methods and will have the smallest average and minmax value.

When we consider the results of the raw data obtained for the Gini coefficient, the average and absolute maximum value of the raw data is consistently smaller than the values of the other methods, except for the MAD, where they are not the smallest for any value of *n*. When we consider the results of the QSR, the raw data did not have the smallest average or the minimum absolute maximum value most of the time. There are even cases where the raw data had the largest value of all the methods for both the average and absolute maximum values. For *n*=

15 000, five of the eight times the raw data did not have the smallest value, and for *n*=1 000, not one of the values of the raw data was the smallest.

For the Theil measure, the raw data contains the smallest average and minimum of the absolute maximum values for both biases and the RMSE, at *n*=15 000 and *n*=10 000. For the MAD, the raw data does not have the smallest average or minimum absolute maximum value. For *n*= 5 000, only two of the eight values of the raw data are the smallest, and for *n*=1 000, none of the answers of the raw data are the smallest.

For the Atkinson measure, the raw data contains the smallest average and minimum absolute maximum value for *n*=15 000, *n*=10 000 and *n*=5 000, for all the measures of performance except for the MAD. For *n*=1 000 the raw data has the smallest value only once.

For the QSR-, Theil- and Atkinson measures, the raw data performed worse when *n* became smaller. For *n*=1 000, only one value of the raw data for the Atkinson measure is the smallest, and none of the values of the raw data for the QSR and Theil are the smallest. For the MAD, not one of the results of the raw data for any of the measures of inequality resulted in the smallest value.

The sign of the biases are consistently negative, except for QSR with *n*=10 000. The raw data, on average, under-predicted the true value. The sign of the biases for the other methods was also almost always negative.

5.4 CONCLUSION

In this chapter, the performances of the methods were studied for the Gini coefficient, QSR, and the Theil- and Atkinson measures. The bias (mean – true value and median – true value), RMSE and MAD were calculated after overall performance measures were obtained by calculating the average and absolute maximum values for each method. The smallest average and minmax values of all methods were then determined.

In the analysis of results, the Pareto methods yielded the best results for all measures of performance except for the MAD, where the lognormal methods yielded the best results. For the Gini cofficient and the Atkinson measure, the raw data performed most of the time better than any other method, but not for the QSR and Theil measure.

In this chapter the methods were tested on simulated data. For simulated data, the true value can be determined, and the methods can be compared to the true value. In Chapter 6, the methods will be tested on real data.

CHAPTER 6
ANALYSIS OF IES DATA

6.1 INTRODUCTION

In this chapter, the methods studied in Chapter 3 are applied to real data, and the measurements studied in Chapter 4 are calculated from each of the methods. The real data is obtained from the Income and Expenditure Survey (IES) 2005/2006. First, we consider some background to the IES data; thereafter, we study the analysis of the results obtained from the real data.

6.2 BACKGROUND TO IES DATA

The dataset that is used in this study is the Income and Expenditure Survey (IES) 2005/2006 data. This data was obtained from a survey conducted by Statistics South Africa (StatsSA), which was undertaken between September 2005 and August 2006. The IES is a representative sample of households that covers the entire South African population. The aim of this survey was to obtain information regarding items and services from households, the various sources of income for these households, as well as how these income sources are spent. This was accomplished by obtaining details of all the expenditures of each household, together with the purchases of goods and services for their own consumption within a specific given reference period. The results obtained through this survey are used to identify the goods and services that should be included into the basket of goods and services for the Consumer Price Index (CPI) (Statistics South Africa, 2008).

The sampling frame used for IES 2005/2006 is based on a designed master sample that consists of enumeration areas (EAs) of the 2001 population census. An EA is the smallest geographical unit that a country is divided into for survey purposes. A stratified multistage sample of 3000 EA’s were obtained from StatsSA’s Master Sample and was used to obtain the information in the report. The EA’s are the primary sampling units (PSUs). From each of these PSUs, eight dwelling units (DUs) were selected systematically for participation. A DU is defined as a structure, part of a structure or a group of structures that is/are occupied or meant to be occupied by one or more households. The occupants of 24 000 DUs were interviewed (Statistics South Africa, 2008).

During the twelve months of data collection, a total of 25 192 households were covered, but 4 048 households were rejected and excluded from the study. Reasons for rejection include refusal, vacant dwellings and non-useable information. Only the data from the occupants of households who completed at least two diaries and the main questionnaire was used. The final sample size for the IES 2005/2006 was 21 144 households (Yu, 2008).

For IES 2005 StatsSA decided to make use of a combination of the recall method and the diary method (the details of which appear in the following paragraph). In the recall method, a single questionnaire was handed out to a selected dwelling unit. The members of a household were required to state their income, as well as the amount spent on all non-durable and semi-durable goods purchased in the month prior to the survey. In the case of both durable goods (long-lasting items or long-term services – for example, cars and furniture) and semi-durable goods (items that require replacement more often - for example, clothing and shoes) the household was also asked to provide details of the purchases of such goods for the twelve months prior to the survey. Data regarding both income and details on how the income was spent was also obtained. The survey period lasted four weeks (Statistics South Africa, 2008).

Each household was given a diary, and required to keep a record of their daily acquisitions for the duration of the survey period. The diaries were collected on a weekly basis to try and ensure that the information was recorded as close as possible to the time of transaction. The information collected was based on the acquisition approach, in order to account for the total value of all goods and services acquired during the period of the survey (Statistics South Africa, 2008).

The raison d'etre of IES is principally the monitoring of consumption expenditure, but the survey is also very useful for the study of household income. Other variables such as household size, population group and provincial demographics are important variables used in the classification of household expenditure. In addition, the monitoring of income is an important factor in studies concerning relative income inequality and poverty (Statistics South Africa, 2008).

In this study, household income is the main dependent variable. These income values gathered in the IES survey is the total annual household income, inflated/deflated to March 2006 prices using CPI indices. There were 39 out of 21 144 households that reported an income of zero and an expenditure value that varied between 1 063 and 76 540. In these cases the expenditure was taken as the income. Hence the variable 'hhincexp' is equal to the income of each household, except where the income is zero; in this case, 'hhincexp' is equal to the expenditure of the household.

The hhincexp values were categorised into intervals. The intervals used are obtained from Community Survey 2006. Table 6.2.1 below is a summary of the IES 2005/2006 data for variable hhincexp represented in a frequency table. The column 'IncCat' refers to income category, 'Lbound' to the lower bound of the interval, 'Ubound' to the upper bound of the interval and 'COUNT' to the frequency for the specific category

Table 6.2.1: IES data in intervals

IncCat	Lbound	Ubound	COUNT
1	0	4800	1481
2	4800	9600	2351
3	9600	19200	5788
4	19200	38400	4940
5	38400	76800	2923
6	76800	153600	2012
7	153600	307200	1126
8	307200	614400	394
9	614400	1228800	101
10	1228800	2457600	19
11	2457600	Infinity	9

6.3 RESULTS OBTAINED FOR IES

For the IES dataset that consist of 21 144 entries, 100 samples of size 10 000, 5 000, 1 000 and 500 were drawn. These samples were categorised into intervals, and each method was applied to the interval data. The measures of inequality were calculated on each dataset obtained from each method per sample. Therefore, 100 estimates were calculated from the samples for each method. The measures of inequality were also calculated on the entire IES dataset, and are perceived as the true values. For the estimates the mean, mean – true value, median, median – true value, standard error, RMSE and MAD were calculated per method. The formulas to calculate the RMSE and MAD are indicated in section 5.2.1 and 5.2.2.

The standard error for the mean of the 100 estimates, $se\left(\hat{\bar{\theta}}\right)$, is calculated for each measure as follows (Lohr, 2010):

$$se\left(\hat{\bar{\theta}}\right) = \frac{1}{10} stdev(\hat{\theta}),$$

where $stdev(\hat{\theta})$ is the standard deviation and

$$stdev(\hat{\theta}) = \sqrt{\frac{1}{100}\sum_{i=1}^{100}\left(\hat{\theta}_i - \hat{\bar{\theta}}\right)^2}$$

with

$$\hat{\bar{\theta}} = \frac{1}{100}\sum_{i=1}^{100}\hat{\theta}_i.$$

After samples were taken from the IES dataset, the entire IES dataset of 21 144 entries was also categorised into intervals and the methods were applied to the categorised data. The measures of inequality were calculated for each method, and the true value was subtracted from the estimate for a measure of performance.

The values calculated from the entire IES 2005/2006 dataset are indicated in the following table.

Table 6.3.1: Values calculated form entire IES 2005/2006 dataset

	True value
Gini coefficient	0.6466
QSR	31.3138
Theil measure	0.8939
Aktinson measure	0.3480

In this study, we consider further the results obtained for the different measures of inequality. The first four tables for each measure of inequality indicate the measures of performance obtained from the drawn samples. The last table of each measure of inequality indicates the estimates obtained from the entire dataset for each method. The bold text values in the tables indicate the method that yielded the best results for each measure of performance. We begin by examining the results of the Gini coefficient.

6.3.1 Gini coefficient

Table 6.3.2: Measures of performance obtained from estimated Gini coefficients based on the 100 samples with *n*=10 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.64076 (0.00064)	**0.64637 (0.00077)**	0.66558 (0.00179)	0.65584 (0.00136)	0.64493 (0.00121)	0.6358 (0.00113)	0.63122 (0.00106)	0.63181 (0.001)	0.63765 (0.00095)	0.64471 (0.00103)	0.62557 (0.0007)	0.65176 (0.00061)	0.66749 (0.00105)	0.63507 (0.00068)
mean-true Gini	-0.00585	**-0.00024**	0.01897	0.00923	-0.00168	-0.01081	-0.01539	-0.01480	-0.00895	-0.00189	-0.02104	0.00516	0.02089	-0.01153
RMSE	0.00864	**0.00763**	0.02604	0.01641	0.01213	0.01558	0.01865	0.01782	0.01299	0.01031	0.02216	0.00798	0.02338	0.01338
median	0.64034	**0.64517**	0.66299	0.65422	0.64282	0.63434	0.62969	0.63035	0.63630	0.64320	0.62455	0.65129	0.66783	0.63426
median-true Gini	-0.00626	**-0.00144**	0.01638	0.00762	-0.00378	-0.01227	-0.01692	-0.01625	-0.01031	-0.00341	-0.02206	0.00468	0.02122	-0.01235
MAD	0.00458	0.00547	0.01321	0.01047	0.00902	0.00846	0.00792	0.00730	0.00721	0.00729	0.00493	**0.00412**	0.00890	0.00466

Table 6.3.3: Measures of performance obtained from estimated Gini coefficients based on the 100 samples with *n*=5 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.64016 (0.00087)	**0.64617 (0.00101)**	0.66839 (0.00244)	0.65711 (0.00177)	0.64566 (0.00157)	0.63622 (0.00147)	0.6314 (0.0014)	0.63185 (0.00135)	0.63801 (0.00133)	0.64856 (0.00173)	0.62519 (0.00093)	0.65115 (0.00084)	0.66936 (0.00126)	0.63427 (0.00087)
mean-true Gini	-0.00645	**-0.00043**	0.02178	0.01051	-0.00094	-0.01039	-0.01521	-0.01476	-0.00859	0.00195	-0.02142	0.00455	0.02276	-0.01233
RMSE	0.01082	0.01001	0.03259	0.02048	0.01561	0.01794	0.02062	0.01995	0.01573	0.01607	0.02333	**0.00950**	0.02597	0.01506
median	0.64145	0.64784	0.66478	0.65630	**0.64574**	0.63655	0.63196	0.63247	0.63804	0.64544	0.62610	0.65259	0.67019	0.63509
median-true Gini	-0.00515	0.00124	0.01818	0.00969	**-0.00087**	-0.01006	-0.01465	-0.01414	-0.00857	-0.00117	-0.02051	0.00598	0.02358	-0.01151
MAD	0.00549	0.00528	0.01322	0.00959	0.00834	0.00813	0.00763	0.00730	0.00676	0.00634	0.00539	**0.00478**	0.00901	0.00558

Table 6.3.4: Measures of performance obtained from estimated Gini coefficients based on the 100 samples with *n*=1 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.63475 (0.00201)	**0.64709 (0.00212)**	0.72727 (0.0066)	0.68458 (0.00384)	0.66361 (0.00331)	0.64891 (0.00311)	0.64029 (0.00299)	0.63845 (0.00292)	0.65361 (0.00275)	0.68951 (0.00241)	0.62305 (0.00206)	0.64589 (0.00207)	0.68436 (0.00329)	0.63326 (0.00209)
mean-true Gini	-0.01185	**0.00048**	0.08066	0.03797	0.01700	0.00230	-0.00632	-0.00815	0.00700	0.04290	-0.02356	-0.00072	0.03775	-0.01335
RMSE	0.02328	0.02114	0.10400	0.05384	0.03706	0.03104	0.03045	0.03022	0.02456	0.02639	0.03121	**0.02060**	0.04994	0.02470
median	0.63759	**0.64775**	0.71447	0.68575	0.66417	0.64894	0.63747	0.63467	0.65329	0.68507	0.62487	0.64858	0.67948	0.63271
median-true Gini	-0.00902	**0.00114**	0.06786	0.03915	0.01756	0.00233	-0.00914	-0.01194	0.00668	0.03846	-0.02174	0.00198	0.03288	-0.01389
MAD	0.01414	0.01566	0.04081	0.02381	0.01934	0.01948	0.01942	0.01882	0.01784	**0.00953**	0.01511	0.01436	0.02276	0.01541

Table 6.3.5: Measures of performance obtained from estimated Gini coefficients based on the 100 samples with *n*=500

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.63184 (0.00256)	0.6475 (0.00274)	0.78317 (0.00925)	0.70594 (0.00516)	0.67667 (0.0045)	0.6577 (0.00424)	**0.64588 (0.0041)**	0.64268 (0.00411)	0.68596 (0.00442)	0.74931 (0.00232)	0.62278 (0.00263)	0.64259 (0.00249)	0.69851 (0.005)	0.63205 (0.00269)
mean-true Gini	-0.01476	0.00089	0.13657	0.05933	0.03006	0.01110	**-0.00073**	-0.00392	0.03935	0.10270	-0.02383	-0.00401	0.05190	-0.01456
RMSE	0.02943	0.02732	0.16469	0.07845	0.05391	0.04366	0.04080	0.04106	0.04074	0.03322	0.03537	**0.02510**	0.07193	0.03046
median	0.62593	0.64399	0.76419	0.69664	0.66895	**0.64853**	0.63739	0.63349	0.66915	0.75065	0.61919	0.63792	0.68306	0.62842
median-true Gini	-0.02068	-0.00262	0.11758	0.05003	0.02234	**0.00192**	-0.00922	-0.01312	0.02254	0.10404	-0.02742	-0.00869	0.03645	-0.01818
MAD	0.01827	0.02009	0.05025	0.04017	0.03036	0.02812	0.02651	0.02551	0.03102	0.02122	0.01897	**0.01757**	0.03174	0.02020

Table 6.3.6: Results obtained for estimated Gini coefficients based on the entire dataset

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
Est Gini	0.6412	**0.6469**	0.6651	0.6558	0.6451	0.6361	0.6316	0.6322	0.6377	0.6436	0.6260	0.6522	0.6608	0.6334
Est Gini – true Gini	-0.0054	**0.0003**	0.0185	0.0092	-0.0015	-0.0105	-0.0150	-0.0144	-0.0089	-0.0031	-0.0206	0.0056	0.0142	-0.0132

The results in Tables 6.3.2 to 6.3.5 are based on the 100 samples, while the results of Table 6.3.6 are based on the entire dataset. The midpoint2 method in Table 6.3.6 is the method that obtained the estimated Gini coefficient closest to the true Gini. The midpoint2 method obtained the midpoint of the interval as a point value in all the observations, and the observations in the last open interval obtain the lower bound times two. The estimate of midpoint2 is only 0.04% larger than the true estimate. The estimate of the lognormal method with the biggest bias is 3.1876% smaller than the true estimate.

Table 6.3.2 is based on a large *n*, a little less than half the dataset. It is clear that the estimated Gini coefficients calculated from the 100 samples of midpoint2 were the closest to the Gini coefficient calculated from the entire IES dataset (the true Gini). The midpoint2 method also has the smallest RMSE, while the random midpoint method has the smallest standard error and MAD. In Table 6.3.3, the midpoint2 method obtained the mean estimate closest to the true value, and the Par_3 method obtained the median estimate closest to the true value. The random midpoint method contains the smallest RMSE and MAD values.

Tables 6.3.4 and 6.3.5 are based on smaller samples; namely *n*=1 000 and *n*=500. The midpoint2 method in Table 6.3.4 contains the mean and median estimates closest to the true value. The random midpoint method contains the smallest RMSE, while the Par_8 method contains the smallest MAD. The Par_5 method in Table 6.3.5 contains the mean estimate closest to the true value, while the Par_4 method contains the median estimate closest to the true value. The random midpoint method contains the smallest RMSE and smallest MAD values.

The results in Tables 6.3.2 – Table 6.3.4 relate to the results of Table 6.3.6 that are based on the entire dataset. In all of these tables, the midpoint2 method contains the mean estimate closest to the true value. The results for Table 6.3.5 are quite different from those of the other tables.

Some bias values are positive while other are negative. A positive sign indicates that the estimate is over-predicted, and a negative sign indicates that an estimate is under-predicted. Since the values under review are so small, it may appear that all of the estimates are close to the true value for all sample sizes. For the estimates in Table 6.3.2 (based on the largest sample), the estimates differ by between 0.0371% and 3.2541% from the true value. For the estimates in Table 6.3.5 (based on the smallest sample size), the estimates differ by between 0.1125% and 15.883% from the true value.

6.3.2 QSR

Table 6.3.7: Measures of performance obtained from estimated QSRs based on the 100 samples with *n*=10 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	20.8305 (0.0574)	21.3319 (0.07)	23.2073 (0.1849)	22.026 (0.1305)	19.9733 (0.1055)	19.6478 (0.0949)	19.3763 (0.0886)	19.8335 (0.0836)	20.5253 (0.0814)	21.1927 (0.0919)	18.836 (0.0557)	35.4377 (0.1234)	42.879 (0.2397)	**29.979 (0.1091)**
mean-true QSR	-10.4832	-9.9818	-8.1064	-9.2877	-11.3404	-11.6659	-11.9374	-11.4803	-10.7885	-10.1210	-12.4778	4.1240	11.5652	**-1.3348**
RMSE	10.4988	10.0061	8.3126	9.3780	11.3889	11.7041	11.9699	11.5103	10.8189	10.0601	12.4901	4.3028	11.8085	**1.7203**
median	20.8741	21.2428	22.9809	21.8352	19.7846	19.4756	19.2303	19.6851	20.4181	21.1116	18.8052	35.3915	43.2317	**30.0137**
median-true QSR	-10.4396	-10.0710	-8.3329	-9.4786	-11.5292	-11.8381	-12.0835	-11.6287	-10.8956	-10.2022	-12.5085	4.0778	11.9179	**-1.3000**
MAD	0.4425	0.5022	1.2770	0.9992	0.8098	0.6898	0.6420	0.5945	0.5974	0.6163	**0.3665**	0.8852	1.9516	0.8146

Table 6.3.8: Measures of performance obtained from estimated QSRs based on the 100 samples with *n*=5 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	20.8284 (0.0808)	21.3698 (0.0959)	23.6553 (0.2849)	22.2386 (0.1829)	20.1111 (0.1455)	19.7495 (0.1313)	19.4507 (0.1239)	19.894 (0.1196)	20.6209 (0.1235)	21.6587 (0.2021)	18.8509 (0.0764)	35.4423 (0.1663)	43.5534 (0.3032)	**29.952 (0.1422)**
mean-true QSR	-10.4854	-9.9439	-7.6584	-9.0751	-11.2027	-11.5643	-11.8631	-11.4198	-10.6928	-9.6551	-12.4629	4.1285	12.2396	**-1.3617**
RMSE	10.5161	9.9896	8.1662	9.2558	11.2959	11.6378	11.9269	11.4816	10.7093	9.1455	12.4860	4.4478	12.6059	**1.9638**
median	20.8513	21.4404	23.0035	21.9846	19.9921	19.7130	19.4568	19.8987	20.5958	21.2446	18.9164	35.2678	43.0980	**29.8871**
median-true QSR	-10.4625	-9.8734	-8.3102	-9.3291	-11.3216	-11.6008	-11.8569	-11.4151	-10.7179	-10.0692	-12.3973	3.9541	11.7842	**-1.4266**
MAD	0.5591	0.5623	1.3257	0.9995	0.8090	0.7502	0.7145	0.6953	0.6536	0.5575	**0.4564**	1.1201	1.9221	0.9278

Table 6.3.9: Measures of performance obtained from estimated QSRs based on the 100 samples with *n*=1 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	21.4826 (0.4509)	22.661 (0.4808)	36.7541 (1.9817)	26.8305 (0.6686)	23.224 (0.5636)	22.2011 (0.524)	21.4816 (0.5038)	21.7026 (0.5053)	23.4464 (0.5499)	28.1196 (0.715)	19.6825 (0.3999)	34.5466 (0.4097)	46.7964 (0.8733)	**29.6944 (0.3203)**
mean-true QSR	-9.8312	-8.6527	5.4403	-4.4832	-8.0898	-9.1127	-9.8322	-9.6112	-7.8673	-3.1941	-11.6313	3.2328	15.4827	**-1.6194**
RMSE	10.8063	9.8871	20.4547	8.0225	9.8434	10.4985	11.0365	10.8466	8.3497	4.1563	12.2932	5.2030	17.7544	**3.5752**
median	20.7094	21.6356	29.8599	25.3083	22.0378	20.9694	20.2320	20.4661	22.1910	25.6147	18.8600	34.5816	46.0351	**29.4288**
median-true QSR	-10.6044	-9.6781	-1.4539	-6.0055	-9.2759	-10.3443	-11.0817	-10.8476	-9.1227	-5.6991	-12.4537	3.2678	14.7214	**-1.8850**
MAD	1.4004	1.5167	6.1242	2.9892	2.0967	1.9668	1.7757	1.7780	1.6876	**1.1569**	1.2034	3.0239	5.8191	2.4015

Table 6.3.10: Measures of performance obtained from estimated QSRs based on the 100 samples with *n*=500

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	24.4019 (0.867)	26.1858 (0.9416)	115.5462 (23.7271)	34.6706 (1.3991)	28.804 (1.1344)	26.8205 (1.0212)	25.508 (0.9547)	25.5191 (0.9389)	**30.6671 (0.9426)**	35.9587 (0.4773)	22.5383 (0.7839)	34.3864 (0.4943)	51.2951 (1.4648)	29.8827 (0.4464)
mean-true QSR	-6.9119	-5.1279	84.2325	3.3568	-2.5097	-4.4932	-5.8058	-5.7947	**-0.6467**	4.6450	-8.7754	3.0727	19.9814	-1.4311
RMSE	11.0540	10.6805	250.6585	14.3199	11.5631	11.1096	11.1329	10.9935	6.4778	**2.0513**	11.7405	5.7989	24.7319	4.6666
median	20.7441	22.3914	46.7122	**30.2809**	24.5025	23.0660	21.5981	21.4983	29.1338	35.9876	19.3982	33.8998	48.2469	29.2852
median-true QSR	-10.5696	-8.9223	15.3985	**-1.0328**	-6.8112	-8.2477	-9.7156	-9.8154	-2.1799	4.6738	-11.9156	2.5860	16.9331	-2.0285
MAD	2.7101	3.2797	17.2448	7.5309	5.5111	4.8328	4.3317	4.2225	7.1083	4.2024	**2.6408**	3.0800	8.5592	2.8395

Table 6.3.11: Results obtained for estimated QSRs based on the entire dataset

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
Est QSR	20.8842	21.3880	23.0885	22.0014	19.9851	19.6747	19.4139	19.8687	20.5351	21.0847	18.8802	35.6336	41.7076	**29.7770**
Est QSR – true QSR	-10.4296	-9.9258	-8.2252	-9.3123	-11.3287	-11.6390	-11.8999	-11.4451	-10.7787	-10.2291	-12.4336	4.3198	10.3939	**-1.5368**

The results in Tables 6.3.7 to 6.3.10 are based on the 100 samples, while the results of Table 6.3.11 are based on the entire dataset. For the QSR, the random lognormal method contains the estimate closest to the true value in Table 6.3.11. The estimate of the random lognormal method is 4.9077% smaller than the true value. The random midpoint method contains the second-closest estimate to the true value, and is 13.7952% greater than the true value. The lognormal midpoint contains the estimate that is furthest from the true value - 39.7064% smaller than the true value. These percentages are much larger than the percentage obtained when using the Gini coefficient.

For the estimates in Tables 6.3.7 to 6.3.9, the random lognormal obtained the mean estimated QSR and median estimated QSR closest to the true QSR. The random lognormal method also contains the smallest RMSE. In Tables 6.3.7 and 6.3.8 it is the lognormal midpoint method that contains the smallest MAD, while the Par_8 method contains the smallest MAD in Table 6.3.9. For the estimates based on n=500 in Table 6.3.10, the Par_7 method contains the mean estimated QSR closest to the true QSR, and Par_2 the median estimated QSR closest to the true QSR. The Par_8 method contains the smallest RMSE, while the lognormal method contains the smallest MAD.

The bias values for the QSR are much larger than for the other measures of inequality. This results in much larger values for the RMSE, since the RMSE can be calculated through the square root of the sum of the bias and the variance. The variance of the estimates is much smaller than the bias of the estimates. A small variance and a large bias indicate that the estimates are close to one another but far from the true value.

The estimates based on n=10 000 in Table 6.3.7 differ by between 4.2626% and 39.8476% from the true value. The estimates in Table 6.3.10, (based on a sample size of 500) differ by between 2.0652% and 268.995% from the true value. The estimate of the Par_7 method, which is 2.0652% smaller than the true value in Table 6.3.9, is 34.4528% smaller than the true value in Table 6.3.7.

6.3.3 Theil measure

Table 6.3.12: Measures of performance obtained from estimated Theil measures based on the 100 samples with *n*=10 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.87546 (0.003)	0.92644 (0.00462)	1.14947 (0.01786)	1.04896 (0.01199)	0.98766 (0.00952)	0.93862 (0.00809)	0.90303 (0.00712)	0.88504 (0.00659)	0.89303 (0.00652)	0.92761 (0.00864)	0.83707 (0.0034)	**0.89448 (0.003)**	1.03173 (0.00982)	0.85155 (0.0035)
mean-true Theil	-0.01846	0.03253	0.25555	0.15504	0.09374	0.04470	0.00911	-0.00888	-0.00088	0.03369	-0.05685	**0.00056**	0.13781	-0.04237
RMSE	0.03508	0.05628	0.31126	0.19560	0.13327	0.09207	0.07142	0.06620	0.06490	0.09143	0.06616	**0.02981**	0.16895	0.05485
median	0.87462	0.92481	1.13332	1.04413	0.98702	0.93754	0.90160	0.88364	0.89207	0.91645	0.83334	**0.89389**	1.02425	0.84940
median-true Theil	-0.01930	0.03090	0.23940	0.15022	0.09310	0.04363	0.00768	-0.01028	-0.00185	0.02254	-0.06058	**-0.00003**	0.13034	-0.04452
MAD	**0.02117**	0.03438	0.11873	0.08214	0.06934	0.05878	0.05155	0.04700	0.04619	0.04888	0.02468	0.02310	0.07800	0.02173

Table 6.3.13: Measures of performance obtained from estimated Theil measures based on the 100 samples with *n*=5 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.87131 (0.0043)	0.92476 (0.00623)	1.17821 (0.02481)	1.06089 (0.01603)	0.99435 (0.0128)	0.94263 (0.011)	0.90522 (0.00984)	0.88656 (0.00935)	0.89769 (0.00988)	0.96354 (0.01764)	0.8345 (0.00468)	**0.8905 (0.00425)**	1.05342 (0.01241)	0.84625 (0.00446)
mean-true Theil	-0.02261	0.03085	0.28429	0.16697	0.10044	0.04871	0.01131	-0.00736	0.00377	0.06963	-0.05942	**-0.00342**	0.15950	-0.04767
RMSE	0.04838	0.06923	0.37654	0.23093	0.16222	0.11976	0.09854	0.09331	0.09783	0.17498	0.07547	**0.04245**	0.20170	0.06514
median	0.87741	0.93329	1.12853	1.04284	0.98214	0.93320	**0.89883**	0.87808	0.88604	0.91462	0.84264	0.89922	1.04411	0.84731
median-true Theil	-0.01650	0.03937	0.23462	0.14893	0.08822	0.03929	**0.00492**	-0.01584	-0.00788	0.02071	-0.05128	0.00531	0.15019	-0.04661
MAD	0.02600	0.03811	0.14844	0.10106	0.07835	0.06585	0.05780	0.05269	0.05138	0.05448	0.02215	**0.02151**	0.08759	0.02966

Table 6.3.14: Measures of performance obtained from estimated Theil measures based on the 100 samples with *n*=1 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.84801 (0.00946)	0.94015 (0.01218)	1.80094 (0.06996)	1.31773 (0.03389)	1.15324 (0.02637)	1.05196 (0.02285)	0.98291 (0.0208)	0.9483 (0.02016)	1.02409 (0.02237)	1.35761 (0.02436)	0.83135 (0.00979)	**0.86652 (0.00978)**	1.20343 (0.03109)	0.85149 (0.01098)
mean-true Theil	-0.04590	0.04623	0.90702	0.42382	0.25932	0.15804	0.08899	0.05438	0.13018	0.46370	-0.06257	**-0.02740**	0.30952	-0.04243
RMSE	0.10471	0.12967	1.14337	0.54157	0.36892	0.27692	0.22526	0.20786	0.22456	0.28102	0.11576	**0.10114**	0.43762	0.11717
median	0.83815	0.91631	1.66186	1.32365	1.14198	1.00719	0.94113	**0.89751**	0.98463	1.30221	0.81974	0.86087	1.13141	0.83570
median-true Theil	-0.05577	0.02239	0.76794	0.42973	0.24806	0.11327	0.04721	**0.00360**	0.09071	0.40829	-0.07417	-0.03304	0.23749	-0.05822
MAD	0.07179	0.08037	0.43997	0.18995	0.17122	0.14081	0.11738	0.11242	0.13955	0.11476	**0.06776**	0.07373	0.21006	0.07003

Table 6.3.15: Measures of performance obtained from estimated Theil measures based on the 100 samples with *n*=500

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.82544 (0.01207)	**0.93169 (0.01591)**	2.41643 (0.10817)	1.4933 (0.04798)	1.24519 (0.03759)	1.10676 (0.03271)	1.01547 (0.02985)	0.97586 (0.03001)	1.28556 (0.04579)	1.88544 (0.03439)	0.82101 (0.01278)	0.84266 (0.01215)	1.32118 (0.04991)	0.8407 (0.01455)
mean-true Theil	-0.06847	**0.03777**	1.52251	0.59938	0.35127	0.21284	0.12156	0.08194	0.39165	0.99152	-0.07291	-0.05126	0.42726	-0.05322
RMSE	0.13820	0.16277	1.86449	0.76628	0.51311	0.38892	0.32090	0.30959	0.41494	0.33008	0.14660	**0.13126**	0.65514	0.15428
median	0.78326	**0.90160**	2.07663	1.34193	1.15123	1.02386	0.93607	0.88508	1.03111	1.93747	0.78856	0.80716	1.12693	0.79683
median-true Theil	-0.11065	**0.00769**	1.18271	0.44801	0.25731	0.12994	0.04215	-0.00884	0.13719	1.04355	-0.10536	-0.08676	0.23302	-0.09709
MAD	**0.06580**	0.09490	0.64719	0.30974	0.18214	0.15907	0.13988	0.12672	0.12926	0.32299	0.07260	0.06739	0.24867	0.08612

Table 6.3.16: Results obtained for estimated Theil measures based on the entire dataset

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
Est Theil	0.8779	0.9296	1.1397	1.0459	0.9866	0.9384	0.9033	0.8850	**0.8909**	0.9133	0.8394	0.8988	0.9521	0.8428
Est Theil – true Theil	-0.0160	0.0357	0.2458	0.1520	0.0926	0.0445	0.0094	-0.0089	**-0.0030**	0.0194	-0.0545	0.0049	0.0581	-0.0511

The results in Tables 6.3.12 to 6.3.15 are based on the 100 samples, while the results of Table 6.3.16 are based on the entire dataset. For the Theil measure, the Par_7 method contains the estimated Theil closest to the true value (as shown in Table 6.3.16). The estimate of Par_7 is 0.3329% smaller than the true value. The Par_1 method contains the estimate with the biggest difference between the estimate and the true value. This estimate is 27.4993% greater than the true value. Eight of the methods contain estimates with a less than 5% difference between the estimate and the true value.

The random midpoint method contains the mean estimate closest to the true value for the estimates based on sample sizes 10 000, 5 000 and 1 000. For the estimates based on a sample size of 500, the midpoint2 method contains the mean estimate closest to the true value. The random midpoint method also contains the median estimate closest to the true value for the estimates based on n=10 000 (as shown in Table 6.3.12). For n=5 000 in Table 6.3.13, the Par_5 method contains the median estimate closest to the true value, while the Par_6 method contains the median estimate closest to the true value for n=1 000 in Table 6.3.14.

The random midpoint method contains the smallest RMSE value in Table 6.3.12, and the midpoint1 method contains the smallest MAD value. The estimates in this table differ by between 0.0625% and 28.5874% from the true value. Eight of the estimates differ by less than 5% from the true value. In Table 6.3.13, the random midpoint method contains both the smallest RMSE and the smallest MAD value. The estimates differ by between 0.3833% and 31.803% from the true value, and six of the estimates differ less than 5% from the true value.

The random midpoint method contains the smallest RMSE value in Table 6.3.14, while the lognormal midpoint method contains the smallest MAD value. The estimates in this table differ by between 3.0652% and 101.4661% from the true value, and only two of the estimates differ by less than 5% from the true value. The random midpoint method also contains the smallest RMSE in Table 6.3.15, while the midpoint1 method contain the smallest MAD value. The estimates differ by between 4.2252% and 170.3188% from the true value, and only the estimate of the midpoint2 method differs by less than 5% from the true value.

6.3.4 Atkinson measure

Table 6.3.17: Measures of performance obtained from estimated Atkinson measures based on the 100 samples with *n*=10 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.34718 (0.00076)	0.35579 (0.00099)	0.38707 (0.00277)	0.37175 (0.00202)	0.3586 (0.00171)	**0.34789 (0.00154)**	0.34159 (0.00141)	0.34068 (0.00132)	0.34587 (0.00128)	0.35429 (0.00149)	0.3313 (0.00083)	0.35458 (0.00075)	0.39008 (0.00156)	0.33455 (0.00083)
mean-true Atk	-0.00087	0.00774	0.03902	0.02370	0.01056	**-0.00015**	-0.00645	-0.00736	-0.00217	0.00624	-0.01675	0.00653	0.04204	-0.01350
RMSE	**0.00760**	0.01252	0.04778	0.03107	0.02004	0.01529	0.01542	0.01507	0.01289	0.01589	0.01867	0.00992	0.04480	0.01584
median	**0.34681**	0.35437	0.38448	0.36919	0.35732	0.34637	0.34004	0.33934	0.34468	0.35242	0.33038	0.35394	0.38970	0.33386
median-true Atk	**-0.00124**	0.00632	0.03643	0.02114	0.00928	-0.00168	-0.00801	-0.00871	-0.00337	0.00437	-0.01766	0.00590	0.04165	-0.01419
MAD	**0.00530**	0.00737	0.01973	0.01456	0.01336	0.01185	0.01077	0.00995	0.00977	0.00993	0.00636	0.00530	0.01236	0.00566

Table 6.3.18: Measures of performance obtained from estimated Atkinson measures based on the 100 samples with *n*=5 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.3464 (0.00105)	0.35556 (0.00131)	0.39163 (0.00382)	0.37379 (0.00265)	0.3598 (0.00225)	**0.34863 (0.00203)**	0.34197 (0.00189)	0.34088 (0.00182)	0.34657 (0.00185)	0.36018 (0.0027)	0.33084 (0.00111)	0.35381 (0.00104)	0.39332 (0.00189)	0.33349 (0.00106)
mean-true Atk	-0.00164	0.00751	0.04358	0.02574	0.01175	**0.00058**	-0.00607	-0.00717	-0.00147	0.01213	-0.01721	0.00576	0.04528	-0.01455
RMSE	**0.01062**	0.01503	0.05782	0.03685	0.02529	0.02025	0.01978	0.01946	0.01838	0.02728	0.02047	0.01181	0.04903	0.01800
median	0.34840	0.35697	0.38537	0.37094	0.35868	**0.34827**	0.34199	0.34064	0.34546	0.35398	0.33253	0.35584	0.39278	0.33429
median-true Atk	0.00035	0.00892	0.03733	0.02289	0.01064	**0.00022**	-0.00606	-0.00740	-0.00259	0.00594	-0.01551	0.00779	0.04473	-0.01375
MAD	0.00610	0.00663	0.02181	0.01621	0.01234	0.01091	0.01017	0.00927	0.00885	0.00927	0.00623	**0.00540**	0.01376	0.00667

Table 6.3.19: Measures of performance obtained from estimated Atkinson measures based on the 100 samples with *n*=1 000

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.34024 (0.00239)	0.35787 (0.00269)	0.48843 (0.01076)	0.41728 (0.00574)	0.38773 (0.00472)	0.36822 (0.00428)	0.35583 (0.00403)	0.3515 (0.00393)	0.36956 (0.00396)	0.42371 (0.00375)	0.32912 (0.00242)	**0.34766 (0.0025)**	0.41707 (0.00495)	0.33338 (0.00258)
mean-true Atk	-0.00780	0.00982	0.14039	0.06924	0.03969	0.02017	0.00779	0.00345	0.02151	0.07567	-0.01892	**-0.00039**	0.06902	-0.01467
RMSE	0.02505	0.02851	0.17653	0.08973	0.06151	0.04714	0.04083	0.03928	0.03911	0.04531	0.03064	**0.02491**	0.08479	0.02953
median	0.34263	0.35591	0.46450	0.41851	0.38921	0.36544	0.35193	**0.34496**	0.36650	0.41865	0.32960	0.35140	0.40611	0.33195
median-true Atk	-0.00541	0.00786	0.11645	0.07047	0.04117	0.01740	0.00388	**-0.00308**	0.01846	0.07060	-0.01844	0.00335	0.05807	-0.01609
MAD	0.01711	0.01895	0.06262	0.03566	0.02813	0.02652	0.02443	0.02357	0.02471	**0.01628**	0.01737	0.01761	0.03372	0.01795

Table 6.3.20: Measures of performance obtained from estimated Atkinson measures based on the 100 samples with *n*=500

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
mean (se)	0.33626 (0.00307)	0.35788 (0.00353)	0.58342 (0.01596)	0.44976 (0.00789)	0.40646 (0.00657)	0.38024 (0.006)	0.36338 (0.00566)	0.35754 (0.00568)	0.41662 (0.00702)	0.51455 (0.00428)	0.32825 (0.00315)	**0.34316 (0.00308)**	0.43785 (0.00787)	0.33177 (0.00338)
mean-true Atk	-0.01178	0.00983	0.23537	0.10172	0.05841	0.03219	0.01533	0.00949	0.06857	0.16651	-0.01979	**-0.00488**	0.08981	-0.01628
RMSE	0.03273	0.03646	0.28393	0.12850	0.08770	0.06781	0.05838	0.05730	0.06762	0.05420	0.03708	**0.03102**	0.11913	0.03737
median	0.32679	**0.35142**	0.54149	0.43044	0.39284	0.36748	0.34968	0.34146	0.38124	0.51828	0.32121	0.33640	0.41618	0.32535
median-true Atk	-0.02125	**0.00338**	0.19344	0.08240	0.04480	0.01943	0.00163	-0.00659	0.03319	0.17023	-0.02683	-0.01164	0.06813	-0.02269
MAD	**0.01937**	0.02380	0.08509	0.05738	0.03839	0.03672	0.03342	0.02960	0.03418	0.03649	0.02102	0.02047	0.04741	0.02358

Table 6.3.21: Results obtained for estimated Atkinson measures based on the entire dataset

	midpoint1	midpoint2	Par_1	Par_2	Par_3	Par_4	Par_5	Par_6	Par_7	Par_8	logN_midp	rand_midp	Rand_pareto	rand_ln
Est Atk	0.3477	0.3564	0.3860	0.3715	0.3586	**0.3481**	0.3419	0.3410	0.3458	0.3523	0.3318	0.3555	0.3792	0.3325
Est Atk – true Atk	-0.0003	0.0084	0.0379	0.0235	0.0106	**0.0001**	-0.0061	-0.0071	-0.0023	0.0042	-0.0162	0.0074	0.0312	-0.0156

The results in Tables 6.3.17 – 6.3.20 are based on the 100 samples, while the results of Table 6.3.21 are based on the entire dataset. For the Atkinson measure, the Par_4 method obtained the estimate closest to the true value in Table 6.3.21. The estimate of the Par_4 method is 0.0202% larger than the true value. The Par_1 method contains the estimate with the biggest difference between the estimate and the true value. The estimate is 10.8958% bigger than the true value. Eleven of the thirteen estimates differ less than 5% from the true value.

For Tables 6.3.17 and 6.3.18, the Par_4 method contains the mean estimated Atkinson closest to the true value. For Tables 6.3.19 and 6.3.20, the random midpoint method contains the mean estimate closest to the true value. The midpoint1 method in Table 6.3.17 contains the median estimate closest to the true value, and the Par_4 method contains the median estimate closest to the true value in Table 6.3.18. The Par_6 method contains the median estimate closest to the true value in Table 6.3.19, while the midpoint2 method contains the median estimate closest to the true value in Table 6.3.20.

The midpoint1 method contains the smallest RMSE and smallest MAD value in Table 6.3.17. For Table 6.3.19, the midpoint1 method contains the smallest RMSE, while the random midpoint method contains the smallest MAD. The random midpoint method contains the smallest RMSE values in both Tables 6.3.19 and 6.3.20. For Table 6.3.19, the Par_8 method contains the smallest MAD value, while the midpoint1 method contains the smallest MAD value in Table 6.3.20.

The estimates based on n=10 000 differ between 0.0434% and 12.0777% from the true value. Eleven of the estimates differ by less than 5% from the true value. For n=5 000, the estimates differ between 0.1666% and 13.0092% from the true value, and eleven estimates differ by less than 5% from the true value. For n=1 000, the estimates differ by between 0.1122% and 40.3355% from the true value and six of the estimates differ by less than 5% from the true value. For the smallest sample size, n=500, the estimates differ by between 1.4024% and 67.6263% from the true value and six estimates differ less than 5% from the true value. There is thus a large increase between the percentage differences between n=5 000 and n=1 000. The number of estimates that differ by less than 5% also fall from eleven to six estimates.

6.4 CONCLUSION

Regarding the simulated data in Chapter 5, the Pareto methods yielded the best overall results. Regarding all the measures of inequality, a Pareto method yielded the best results for all measures of performance except the MAD. Regarding the real data, the method that yielded the best results for each measure of inequality differs from the other measures of inequality. The tables below summarise the method that obtained the mean estimates closest to the true value for each of the measures of inequality. The sample sizes are indicated just below the measure of inequality, and the column "all" refers to the estimates obtained from the entire dataset.

Table 6.3.22 Method with mean estimate closest to the true value

	Gini					QSR				
	all	**n=10 000**	**n=5 000**	**n=1 000**	**n=500**	**all**	**n=10 000**	**n=5 000**	**n=1 000**	**n=500**
Method	Midpoint2	Midpoint2	Midpoint2	Midpoint2	Par_5	rand_ln	rand_ln	rand_ln	rand_ln	Par_7

	Theil					Atk				
	all	**n=10 000**	**n=5 000**	**n=1 000**	**n=500**	**all**	**n=10 000**	**n=5 000**	**n=1 000**	**n=500**
Method	Par_7	rand_midp	rand_midp	rand_midp	midpoint2	Par_4	Par_4	Par_4	rand_midp	rand_midp

The range between the mean estimate with the smallest percentage difference and the largest percentage difference increases significantly between sample sizes 5 000 and 1 000. This is especially true for the Gini coefficient and the Theil- and Atkinson measures. Regarding the QSR, there is a large difference between the sample sizes of 1 000 and 500.

Since the results of the simulated data and the IES data differ, it probably indicates that the IES data differ significantly from the distributions used in the simulations. It is known that the IES data has a rather uncommon form due to its high proportion of low income values.

CHAPTER 7
CONCLUSIONS

7.1 SUMMARY

Some datasets exist in the form of grouped data, with entries in interval form. This format is especially prevalent where more sensitive information, such as income data, is concerned. The task of processing and carrying out the necessary calculations when working with grouped data is very often much more difficult than applying the same procedures to ungrouped/raw data; in some cases it is impossible. There are different methods for converting grouped data into data where each observation has a specific value or a random value. In this study, different methods of converting grouped data to point data were tested on simulated and real datasets.

Income data was the central focus of this study; since one of the most frequently-occurring characteristics of income data is a heavy-tailed distribution (manifesting in a skewed distribution with a long/longer tail to the right), we simulated from heavy-tailed distributions. These distributions are the Pareto- (α=3; α=1.5; α=1.1), lognormal- (μ=0.5, σ=0.5; μ=0.6, σ=1; μ=1.5, σ=0.8) and Burr (c=1.2, k=2.5; c=3, k=1; c=1.3, k=1.5; c=3, k=1.1/3) distributions. For the real data, the IES 2005/2006 dataset was used.

Six methods were used in this study to convert grouped data into point or random data. The midpoint-, Pareto means- and lognormal means methods are methods in which all observations in an interval obtain the same value, i.e. the midpoint or the conditional mean. The random midpoint-, random Pareto- and random lognormal methods are methods in which each observation in an interval obtains a random value between the lower and upper bound of the interval. For the Pareto- and lognormal methods a value is assigned to an observation according to a distribution fitted to the interval data.

After the methods were applied to the interval data, a new dataset was obtained for each method, in which each observation has a point or random value. The Gini coefficient, QSR (quintile share ratio), Theil measure and Atkinson measure were then calculated from these values for each method. Since we know which distribution is simulated from, the true measure in inequality could be calculated.

From each distribution with corresponding parameters, 1 000 data sets were generated so that 1 000 estimates of each measure of inequality could be obtained. Four samples sizes were used; these were 1 000, 5 000, 10 000 and 15 000. From the 1 000 estimates, the bias (mean – true value and median – true value), RMSE and MAD (median absolute deviation) were calculated. The average was calculated over all the distributions simulated, separately from each sample size and each measure of performance. We also calculated the absolute maximum value over all the distributions (the distribution that performed the worst for each method) and determined the minimum of all the absolute maxima to obtain the minmax. The minmax thus indicates the best of the worst methods.

Regarding the simulated data, the Pareto methods, Pareto means and random Pareto, yielded the best results overall, with the largest number of smallest averages and the largest number of minmax values. For the two biases and RMSE, a Pareto method contained the smallest average and the minmax value every time. For the MAD, the lognormal methods, lognormal means and random lognormal contained all the smallest averages and all the minmax values except for the Gini coefficient, where the random midpoint contained the smallest averages and minmax values. The RMSE (a function of averages) is a less robust function that is fairly susceptible to influences from larger values; the MAD, on the other hand, is a more robust function, and not as vulnerable to influence by larger values because of the deviation around the median. For this reason, the MAD may be more suitable for use with the Pareto distribution, and the RMSE for the lognormal distribution, which has a shorter tail than the Pareto distribution.

After the methods were tested on simulated data, we applied the method to a real data set. The dataset that was used is the IES 2005/2006 dataset, which consists of 21144 observations. Each observation in the dataset has a point value. The household income is used as the dependent variable. There are 39 cases where the income of the household is equal to zero but the household expenditure is not equal to zero. In these cases the household expenditure was used as the household income.

First, 100 samples of 500, 1 000, 5 000 and 10 000 observations were drawn. The observations for each sample size were then categorised into intervals per sample. The methods were applied to each sample in order to obtain a new dataset for each method with a new point value for each observation. The measures of inequality were calculated per sample for each method to yield 100 estimates per method. The measures of inequality were also calculated on the raw 21144 observations of the IES dataset, and perceived as the true values. For the 100 estimates, the mean, mean – true value, median, median – true value, standard deviation, RMSE and MAD were calculated.

After the results were obtained from the samples we categorised the entire IES dataset of 21 144 observations into intervals. The methods were applied to the resultant interval data in order to obtain a new dataset, with new values for each method. The measures of inequality were calculated for each method and compared to the true values obtained from the raw data.

The methods that performed the best for the real data differed for the various measures of inequality. Regarding the Gini coefficient, the midpoint2 method yielded the estimate closest to the true value for the estimates based on the entire dataset, as well as the mean of the estimates for all samples sizes, except n=500. Regarding the QSR, the random lognormal method yielded the estimates closest to the true value for the estimates based on the entire dataset, as well as the mean of the estimates for all samples sizes, except n=500.

The Par_7 method contains the estimated Theil measure closest to the true value for the estimates based on the entire dataset. For the Par_7 method, the observations in the first seven intervals obtained the midpoint as a point value, while the remaining four intervals obtained the Pareto midpoint as a point value. The random midpoint method contains the mean estimate closest to the true value, for all sizes except n=500, for the Theil measure. Regarding the Atkinson measure, the Par_4 method obtained the estimate closest to the true value for the estimates based on the entire dataset, as well as for the mean of the estimates based on samples sizes of 10 000 and 5 000. For sample sizes 1 000 and 500, the random midpoint method contains the mean estimated Atkinson closest to the true value.

The Pareto methods performed best when processing the simulated data, as they contain most of the smallest averages and minmax values. However, this was not true regarding the real data; different methods performed best for different measures of inequality. This indicates that the IES data is probably not of the same form as the distributions simulated from due to its high proportion of low income values.

7.2 FURTHER RESEARCH

Since different results were obtained for the simulated and real data, further examination is necessary. The different methods with the different distributions are only evaluated on one real data set, namely the IES 2005/2006. It is advisable that these techniques be tested on more real data sets which follow different heavy-tailed distributions. One could also test the methods on datasets obtained by simulating from distributions other than the Pareto-, lognormal- and Burr distributions.

Another aspect that would benefit from further examination is the effect of different number and widths of intervals on the estimates. Seiver (1979) found that the widths and number of intervals exercise an influence upon the results of the income distribution. Fewer and wider intervals resulted in over-estimation of inequality measures. However, the influence of different intervals, i.e. different widths and different numbers of intervals, on poverty and inequality estimates is a field of research in need of further investigation. Thus there is a need to expand the current body of knowledge regarding the number and the widths of intervals to be used and such further research would be of value to scholars and practitioners alike.

References

Atkinson, A.B. 1970. On the measurement of inequality. *Journal of Economic Theory*, 2:244-263.

Cowell, F.A. and Flachaire, E. 2007. Income Distribution and Inequality Measurement: The Problem of Extreme Values. *Journal of Econometrics*, 141:1044–1072.

Dagsvik, J.K. and Vatne, B.H. 1999. Is the Distribution of Income Compatible with a Stable Distribution? Discussion Paper No. 246. Statistics Norway, Research Department. Kongsvinger.

Eurostat. 2003. 'Laeken' Indicators, Detailed Calculation Methodology. Working Group Statistics on Income, Poverty and Social Exclusion.

Fields, G.S. 1989. A compendium of data on inequality and poverty for the developing world. Unpublished report. Cornell University, New York.

Gradshteyn, I.S. and Ryzhik, I.M. 1965. *Table of Integrals, Series, and Products.* 4th edition. Academic Press.

Gustavsson, M. 2004. Trends in the Transitory Variance of Earnings: Evidence from Sweden 1960-1990 and a Comparison with the United States. Working Paper 2004:11. Uppsala University, Sweden.

Haughton, J. and Khandker, S.R. 2009. *Handbook on Poverty and Inequality.* The World Bank. Available from: <http://web.worldbank.org/WBSITE/EXTERNAL/TOPICS/EXTPOVERTY/EXTPA/0,,contentMDK:22405907~pagePK:148956~piPK:216618~theSitePK:430367,00.html>. [18 October 2014].

Hofmeyr, J.F. 2001. Segmentation in the South African Labour Market in1999. Paper presented at DPRU/FES Conference on Labour Markets and Poverty in South Africa. Johannesburg, South Africa.

Kpanzou, T.A. 2011. Statistical Inference for Inequality Measures Based on Semi-Parametric Estimators. Dissertation presented for the degree of Doctor of Philosophy at Stellenbosch University

Lohr, S.L. 2010. *Sampling: Design and Analysis.* Brooks/Cole Publishing Company.

Malherbe, J.E. 2007. An analysis of income and poverty in South Africa. Assignment presented in partial fulfilment of the requirements for the degree of Master of Commerce at Stellenbosch University.

Nishino, H. and Kakamu, K. 2011. Grouped data estimation and testing of Gini coefficient using lognormal distributions. *Sankhya Series B,* 73(2), 193-210.

Rospabé, S. 2002. How did Labour Market Racial Discrimination Evolve After the End of Apartheid? *The South African Journal of Economics,* Vol 70, No.1: 185-217.

Seiver, D.A. 1979. A note on the measurement of income inequality with interval data. The *Review of Income and Wealth,* Vol 25, issue 2: 229-233.

Statistics South Africa. 2008. Income and expenditure of households 2005/2006: Statistical release. Technical report, Statistics South Africa.

Von Fintel, D. 2006. Earnings bracket obstacles in household surveys – How sharp are the tools in the shed? Stellenbosch Economic Working Papers.

West, S.A. 1986. Estimation of the mean from censored income data. *Proceedings of the Survey Research Methods Section, American Statistical Association*, 665-670.

Whiteford, A. and McGrath, M. 1994. *The Distribution of Income in South Africa.* Pretoria: Human Sciences Research Council.

Yu, D. 2008. The comparability of Income and Expenditure Surveys 1995, 2000 and 2005/2006. Stellenbosch Economic Working Papers.

Yu, D. 2013. Some factors influencing the comparability and reliability of poverty estimates across household surveys. Stellenbosch Economic Working Papers.

APPENDIX A:

SUMMARISED TABLES

TABLES OBTAINED FOR THE GINI COEFFICIENT FOR n=15 000

Table A1: Mean – true Gini (Gini, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01222 (0.00029)	0.01451 (0.00029)	-0.07079 (0.00021)	-0.02027 (0.00033)	0.00455 (0.00028)	–	–	-0.003 (0.00026)	0.01535 (0.00035)	0.00001 (0.00026)	-0.08376 (0.0003)	0.00007 (0.00021)
	α=1.5	-0.37022 (0.00089)	-0.36287 (0.001)	-0.21066 (0.00187)	-0.36478 (0.00195)	-0.35943 (0.0025)	-0.35375 (0.00349)	-0.33955 (0.00881)	-0.38288 (0.00082)	-0.15302 (0.00058)	-0.00456 (0.00202)	-0.32419 (0.00104)	-0.00393 (0.00353)
	α=1.1	-0.37544 (0.00237)	-0.35923 (0.00252)	-0.12488 (0.00544)	-0.29713 (0.00893)	-0.29517 (0.00937)	-0.2952 (0.0096)	-0.29367 (0.00734)	-0.44716 (0.00201)	-0.25459 (0.00194)	-0.09626 (0.00325)	-0.44141 (0.00234)	-0.07698 (0.00522)
Burr	c=1.2, k=2.5	-0.42448 (0.00037)	-0.42183 (0.00043)	-0.47303 (0.00028)	-0.44157 (0.00042)	-0.42996 (0.00041)	–	–	-0.43537 (0.00034)	-0.42395 (0.00048)	-0.38512 (0.00032)	-0.44722 (0.00047)	-0.00001 (0.00028)
	c=3, k=1	-0.13373 (0.00038)	-0.13133 (0.00033)	-0.21229 (0.00025)	-0.16418 (0.00039)	-0.14149 (0.00033)	–	–	-0.14875 (0.00029)	-0.13121 (0.0004)	-0.13885 (0.00027)	-0.21997 (0.00043)	-0.00007 (0.00028)
	c=1.3, k=1.5	-0.26929 (0.0004)	-0.24445 (0.00051)	-0.3402 (0.00065)	-0.25587 (0.00095)	-0.25164 (0.00093)	–	–	-0.28867 (0.00042)	-0.26643 (0.0004)	-0.28124 (0.00119)	-0.35955 (0.00046)	-0.00052 (0.00098)
	c=3, k=1.1/3	-0.32606 (0.00018)	-0.25541 (0.00027)	-0.04929 (0.00156)	-0.01242 (0.00175)	-0.02971 (0.00232)	–	–	-0.31335 (0.00031)	-0.31682 (0.00017)	-0.09939 (0.00451)	-0.36959 (0.00047)	-0.07376 (0.00531)
Lognormal	μ=0.5, σ=0.5	0.0257 (0.0001)	0.02641 (0.00012)	-0.06498 (0.00013)	-0.01368 (0.00024)	0.01991 (0.0001)	0.02518 (0.00013)	–	0.02021 (0.00009)	0.04399 (0.00011)	-0.02213 (0.00019)	-0.11001 (0.00011)	0.00006 (0.00017)
	μ=0.6, σ=1	-0.02159 (0.00043)	-0.01247 (0.00053)	-0.08004 (0.00075)	-0.02859 (0.0007)	-0.0379 (0.00066)	-0.03322 (0.00058)	–	-0.05189 (0.00048)	-0.00942 (0.00042)	-0.05519 (0.00151)	-0.12231 (0.00063)	0.00001 (0.00041)
	μ=1.5, σ=0.8	0.01526 (0.00019)	0.02081 (0.00023)	0.01438 (0.00034)	-0.01821 (0.00028)	-0.02376 (0.00028)	-0.0211 (0.00026)	–	-0.01886 (0.00021)	0.03844 (0.00018)	0.03878 (0.0011)	-0.0459 (0.00025)	-0.00001 (0.00018)

Table A2: Median – true Gini (Gini, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01231 (0.0004)	0.01461 (0.00047)	-0.07067 (0.00034)	-0.02015 (0.00047)	0.00463 (0.00046)	–	–	-0.00305 (0.00041)	0.01543 (0.00041)	0.00003 (0.00035)	-0.08359 (0.00044)	-0.00001 (0.00032)
	α=1.5	-0.37082 (0.00166)	-0.36409 (0.00192)	-0.21453 (0.00269)	-0.36909 (0.00145)	-0.36528 (0.00128)	-0.36202 (0.00196)	-0.35769 (0.0034)	-0.38359 (0.00163)	-0.15336 (0.00109)	-0.00971 (0.00202)	-0.32469 (0.00169)	-0.00948 (0.00338)
	α=1.1	-0.37377 (0.00264)	-0.3584 (0.00328)	-0.12932 (0.00552)	-0.31245 (0.0066)	-0.31409 (0.00559)	-0.31613 (0.0048)	-0.31978 (0.00482)	-0.44681 (0.00255)	-0.25389 (0.00274)	-0.10391 (0.0021)	-0.4401 (0.00283)	-0.09076 (0.00617)
Burr	c=1.2, k=2.5	-0.42444 (0.00045)	-0.4218 (0.00045)	-0.4731 (0.00036)	-0.4417 (0.0006)	-0.43011 (0.00049)	–	–	-0.43532 (0.00041)	-0.42396 (0.00058)	-0.38528 (0.0004)	-0.44727 (0.00072)	0.00007 (0.0005)
	c=3, k=1	-0.13361 (0.0006)	-0.13128 (0.0005)	-0.21229 (0.00031)	-0.16433 (0.00037)	-0.14138 (0.00037)	–	–	-0.14854 (0.00048)	-0.13107 (0.00056)	-0.1389 (0.00037)	-0.22021 (0.00064)	-0.00027 (0.00032)
	c=1.3, k=1.5	-0.26916 (0.00041)	-0.24455 (0.00069)	-0.34044 (0.00109)	-0.25613 (0.00128)	-0.25206 (0.00095)	–	–	-0.28875 (0.00057)	-0.26646 (0.00052)	-0.28299 (0.00132)	-0.35959 (0.00057)	-0.00174 (0.00097)

	c=3, k=1.1/3	-0.32606 (0.00023)	-0.25538 (0.00042)	-0.04904 (0.00232)	-0.01314 (0.00267)	-0.03129 (0.0033)	–	–	-0.3133 (0.00048)	-0.31683 (0.0002)	-0.10502 (0.00642)	-0.36958 (0.00053)	-0.08255 (0.00653)
Lognormal	μ=0.5, σ=0.5	0.02575 (0.00014)	0.02642 (0.00017)	-0.06493 (0.00015)	-0.01372 (0.00026)	0.01992 (0.00012)	0.02522 (0.00016)	–	0.02024 (0.00008)	0.04401 (0.00016)	-0.02214 (0.00021)	-0.10997 (0.00017)	0.00009 (0.00026)
	μ=0.6, σ=1	-0.02162 (0.00055)	-0.0125 (0.00064)	-0.08022 (0.00094)	-0.02873 (0.00088)	-0.03797 (0.00083)	-0.03336 (0.00079)	–	-0.05197 (0.00065)	-0.00942 (0.00051)	-0.05726 (0.00131)	-0.12232 (0.00079)	-4.74E-06 (0.00036)
	μ=1.5, σ=0.8	0.0151 (0.00037)	0.02069 (0.00032)	0.01407 (0.00034)	-0.01833 (0.00034)	-0.02384 (0.00034)	-0.02122 (0.00028)	–	-0.01902 (0.00033)	0.0383 (0.00034)	0.03712 (0.00047)	-0.04605 (0.0005)	-0.00001 (0.00032)

Table A3: RMSE (Gini, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01284 (0.00028)	0.01512 (0.00027)	0.07086 (0.00021)	0.02086 (0.00037)	0.00612 (0.00024)	–	–	0.00442 (0.00023)	0.0159 (0.00034)	0.00359 (0.00033)	0.08388 (0.00031)	0.00292 (0.00021)
	α=1.5	0.37036 (0.00088)	0.36307 (0.001)	0.21193 (0.00175)	0.36543 (0.00178)	0.3603 (0.0021)	0.35514 (0.00275)	0.34448 (0.00558)	0.383 (0.00082)	0.1532 (0.00057)	0.0296 (0.00608)	0.32451 (0.00102)	0.02916 (0.00803)
	α=1.1	0.37626 (0.00243)	0.36023 (0.00255)	0.1405 (0.00382)	0.31258 (0.00672)	0.31139 (0.00681)	0.31092 (0.007)	0.30804 (0.00554)	0.44778 (0.00203)	0.25535 (0.00199)	0.10268 (0.00257)	0.44218 (0.00236)	0.09564 (0.004)
Burr	c=1.2, k=2.5	0.4245 (0.00037)	0.42186 (0.00043)	0.47305 (0.00028)	0.4416 (0.00041)	0.42999 (0.00041)	–	–	0.43539 (0.00034)	0.42398 (0.00048)	0.38514 (0.00032)	0.44727 (0.00046)	0.00403 (0.00021)
	c=3, k=1	0.13376 (0.00037)	0.13137 (0.00032)	0.21228 (0.00025)	0.16422 (0.00038)	0.14152 (0.00033)	–	–	0.14876 (0.00029)	0.13125 (0.00039)	0.13886 (0.00027)	0.21998 (0.00043)	0.0032 (0.00025)
	c=1.3, k=1.5	0.26933 (0.0004)	0.24454 (0.00051)	0.34029 (0.00065)	0.25612 (0.00094)	0.25187 (0.00092)	–	–	0.28872 (0.00042)	0.26647 (0.0004)	0.28152 (0.00115)	0.35958 (0.00046)	0.00954 (0.00136)
	c=3, k=1.1/3	0.32604 (0.00018)	0.2554 (0.00027)	0.05201 (0.0016)	0.02287 (0.00173)	0.03888 (0.00213)	–	–	0.31334 (0.00031)	0.3168 (0.00017)	0.10894 (0.00383)	0.36959 (0.00047)	0.08748 (0.00302)
Lognormal	μ=0.5, σ=0.5	0.02573 (0.0001)	0.02645 (0.00012)	0.065 (0.00013)	0.01393 (0.00025)	0.01996 (0.0001)	0.02522 (0.00013)	–	0.02024 (0.00009)	0.04401 (0.00011)	0.0222 (0.00019)	0.11003 (0.00011)	0.00169 (0.00015)
	μ=0.6, σ=1	0.0218 (0.00044)	0.01305 (0.00054)	0.08024 (0.00076)	0.02906 (0.0007)	0.03821 (0.00066)	0.03349 (0.00058)	–	0.05201 (0.00049)	0.0099 (0.00042)	0.0568 (0.0014)	0.12239 (0.00063)	0.00345 (0.00029)
	μ=1.5, σ=0.8	0.0155 (0.0002)	0.02107 (0.00023)	0.01506 (0.00035)	0.01861 (0.00028)	0.02409 (0.00029)	0.0214 (0.00026)	–	0.01911 (0.00021)	0.03853 (0.00018)	0.04003 (0.00221)	0.04602 (0.00025)	0.00278 (0.00022)

Table A4: MAD (Gini, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00272 (0.0002) (0.00015)	0.00293 (0.00023) (0.00004)	0.00224 (0.00017) (0.00008)	0.00334 (0.00047) (0.00024)	0.00278 (0.00028) (0.00018)	–	–	0.00221 (0.00022) (0.00021)	0.00282 (0.00031) (0.00026)	0.00242 (0.00025) (0.00011)	0.00296 (0.00052) (0.00029)	0.00194 (0.00019) (0.00007)
	α=1.5	0.00674 (0.00086) (0.00061)	0.00827 (0.00106) (0.00083)	0.01516 (0.00196) (0.00093)	0.01249 (0.00193) (0.00153)	0.01253 (0.00183) (0.00133)	0.0132 (0.00164) (0.00113)	0.01518 (0.00248) (0.00167)	0.00646 (0.00077) (0.00054)	0.00509 (0.00067) (0.00051)	0.01272 (0.0018) (0.00096)	0.00976 (0.00111) (0.00066)	0.01255 (0.00152) (0.00056)
	α=1.1	0.01607 (0.00191) (0.00167)	0.01798 (0.00106) (0.00093)	0.05353 (0.00752) (0.00411)	0.07068 (0.00903) (0.00572)	0.06871 (0.00858) (0.00623)	0.06439 (0.00874) (0.0033)	0.0595 (0.0064) (0.00335)	0.01635 (0.00119) (0.00051)	0.01276 (0.00186) (0.00109)	0.01936 (0.00297) (0.00092)	0.01775 (0.00174) (0.00073)	0.03023 (0.00489) (0.00326)

Burr	c=1.2, k=2.5	0.00316 (0.00034) (0.00026)	0.0034 (0.00042) (0.00032)	0.00233 (0.00026) (0.00012)	0.00369 (0.00032) (0.00028)	0.00328 (0.00034) (0.00025)	–	–	0.00274 (0.00033) (0.00024)	0.0033 (0.00024) (0.00013)	0.0027 (0.00019) (0.00015)	0.00454 (0.00034) (0.00017)	0.00275 (0.00021) (0.00011)
	c=3, k=1	0.00273 (0.0004) (0.00023)	0.00295 (0.0004) (0.00023)	0.00219 (0.00028) (0.00019)	0.0033 (0.00041) (0.00026)	0.00284 (0.00043) (0.00026)	–	–	0.00234 (0.00033) (0.00011)	0.00288 (0.00043) (0.00017)	0.00228 (0.00028) (0.00021)	0.00303 (0.00019) (0.00009)	0.00201 (0.00021) (0.00017)
	c=1.3, k=1.5	0.00351 (0.00048) (0.00024)	0.00458 (0.0006) (0.00026)	0.00533 (0.00061) (0.00033)	0.00737 (0.00076) (0.0006)	0.0073 (0.00068) (0.00053)	–	–	0.00377 (0.00056) (0.00026)	0.00357 (0.00043) (0.00021)	0.00692 (0.00065) (0.00037)	0.00373 (0.00048) (0.00021)	0.00509 (0.00041) (0.00026)
	c=3, k=1.1/3	0.00137 (0.00018) (0.00018)	0.00211 (0.00025) (0.00007)	0.01101 (0.00188) (0.0014)	0.01337 (0.00154) (0.00138)	0.01701 (0.00215) (0.00164)	–	–	0.00235 (0.00036) (0.00014)	0.00133 (0.00019) (0.00009)	0.02865 (0.00306) (0.00129)	0.00309 (0.00047) (0.00034)	0.02611 (0.0038) (0.00109)
Lognormal	μ=0.5, σ=0.5	0.00089 (0.00012) (0.0001)	0.00094 (0.00013) (0.00011)	0.00111 (0.00015) (0.00013)	0.00175 (0.00015) (0.00013)	0.00085 (0.0001) (0.00008)	0.00086 (0.0001) (0.00007)	–	0.00078 (0.00009) (0.00005)	0.00093 (0.00011) (0.00006)	0.00119 (0.00013) (0.00011)	0.00123 (0.00018) (0.00014)	0.00116 (0.00014) (0.00009)
	μ=0.6, σ=1	0.0021 (0.00023) (0.00021)	0.00269 (0.00026) (0.00012)	0.00376 (0.00045) (0.00022)	0.00343 (0.0004) (0.00012)	0.0032 (0.00036) (0.00015)	0.0028 (0.00031) (0.0001)	–	0.00236 (0.00027) (0.00014)	0.00209 (0.00028) (0.0003)	0.00499 (0.00057) (0.00043)	0.003 (0.00029) (0.00033)	0.00231 (0.0003) (0.00025)
	μ=1.5, σ=0.8	0.0018 (0.00026) (0.00019)	0.00217 (0.00022) (0.00018)	0.00297 (0.00022) (0.00015)	0.00256 (0.00022) (0.00011)	0.00261 (0.00023) (0.00012)	0.00238 (0.00021) (0.00017)	–	0.00206 (0.0003) (0.00027)	0.00172 (0.00026) (0.00021)	0.00364 (0.00034) (0.00022)	0.00209 (0.00018) (0.00013)	0.00186 (0.00022) (0.00011)

TABLES OBTAINED FOR THE GINI COEFFICIENT FOR n=10 000

Table A5: Mean – true Gini (Gini, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01197 (0.00039)	0.0147 (0.00042)	-0.07073 (0.00031)	-0.02007 (0.00049)	0.00494 (0.0005)	–	–	-0.003 (0.00033)	0.01509 (0.00043)	0.00009 (0.00032)	-0.0839 (0.00042)	0.00017 (0.00038)
	α=1.5	-0.37127 (0.00106)	-0.36381 (0.00119)	-0.21207 (0.00208)	-0.36612 (0.00192)	-0.36103 (0.00211)	-0.35502 (0.06511)	-0.34039 (0.00443)	-0.38368 (0.00097)	-0.15367 (0.00081)	-0.00755 (0.00271)	-0.32589 (0.00158)	-0.00569 (0.00299)
	α=1.1	-0.38257 (0.00378)	-0.36555 (0.00412)	-0.1277 (0.00913)	-0.3018 (0.01382)	-0.30051 (0.01429)	-0.29959 (0.0139)	-0.28979 (0.01587)	-0.45207 (0.00396)	-0.25989 (0.00294)	-0.10048 (0.00695)	-0.4462 (0.0045)	-0.08425 (0.00601)
Burr	c=1.2, k=2.5	-0.42495 (0.00097)	-0.4216 (0.00104)	-0.47298 (0.0007)	-0.44119 (0.00105)	-0.42929 (0.03346)	–	–	-0.43547 (0.00085)	-0.42452 (0.00089)	-0.38497 (0.00086)	-0.44757 (0.00114)	-0.00019 (0.00074)
	c=3, k=1	-0.13385 (0.00061)	-0.13104 (0.00067)	-0.21213 (0.00049)	-0.16387 (0.00072)	-0.14111 (0.00058)	–	–	-0.14866 (0.0005)	-0.13137 (0.00062)	-0.13871 (0.00053)	-0.21994 (0.0006)	-1.8E-05 (0.00055)
	c=1.3, k=1.5	-0.26944 (0.00073)	-0.24445 (0.00077)	-0.33998 (0.00081)	-0.25537 (0.00096)	-0.25105 (0.00078)	–	–	-0.28873 (0.00073)	-0.26656 (0.00074)	-0.28116 (0.00158)	-0.35949 (0.00069)	-0.00007 (0.00139)
	c=3, k=1.1/3	-0.32609 (0.00022)	-0.25539 (0.00029)	-0.04866 (0.0013)	-0.01188 (0.00183)	-0.0286 (0.00237)	–	–	-0.31333 (0.00033)	-0.3168 (0.00022)	-0.10237 (0.0033)	-0.36937 (0.00044)	-0.07 (0.00473)
Lognormal	μ=0.5, σ=0.5	0.0257 (0.0002)	0.0266 (0.0002)	-0.06499 (0.00023)	-0.01321 (0.00031)	0.02002 (0.00019)	0.02533 (0.00022)	–	0.02021 (0.00016)	0.044 (0.00016)	-0.02194 (0.00022)	-0.10997 (0.00027)	0.00004 (0.0003)
	μ=0.6, σ=1	-0.02164 (0.00027)	-0.01238 (0.00036)	-0.07976 (0.00061)	-0.02835 (0.00052)	-0.03768 (0.00048)	-0.03301 (0.00042)	–	-0.05189 (0.00031)	-0.00942 (0.00032)	-0.0557 (0.00185)	-0.12233 (0.00039)	-0.00016 (0.0004)
	μ=1.5, σ=0.8	0.01526 (0.00022)	0.02071 (0.00026)	0.01423 (0.00037)	-0.01834 (0.0003)	-0.02392 (0.00031)	-0.02126 (0.00028)	–	-0.0189 (0.00025)	0.03844 (0.00029)	0.03851 (0.00113)	-0.04586 (0.00031)	-0.00004 (0.00021)

Table A6: Median – true Gini (Gini, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01193 (0.00035)	0.01457 (0.00039)	-0.07073 (0.0004)	-0.02019 (0.00067)	0.00484 (0.00047)	–	–	-0.00305 (0.00027)	0.01489 (0.00064)	0.00004 (0.00048)	-0.08377 (0.00079)	0.0001 (0.00046)
	α=1.5	-0.3717 (0.0017)	-0.36452 (0.00155)	-0.21476 (0.00206)	-0.36954 (0.00173)	-0.36517 (0.00166)	-0.36121 (0.00193)	-0.35293 (0.00385)	-0.38436 (0.00143)	-0.15411 (0.00115)	-0.01232 (0.00255)	-0.32601 (0.00193)	-0.01127 (0.00273)
	α=1.1	-0.38008 (0.00434)	-0.36395 (0.00589)	-0.14723 (0.01415)	-0.33757 (0.01851)	-0.34133 (0.01516)	-0.34487 (0.01579)	-0.34441 (0.03419)	-0.45168 (0.00556)	-0.25906 (0.00334)	-0.1067 (0.00604)	-0.44489 (0.00622)	-0.09504 (0.00804)
Burr	c=1.2, k=2.5	-0.42491 (0.00108)	-0.42187 (0.00124)	-0.47322 (0.00079)	-0.44153 (0.0011)	-0.42965 (0.00115)	–	–	-0.43564 (0.00087)	-0.42432 (0.0007)	-0.38523 (0.00076)	-0.44791 (0.00125)	-0.00041 (0.00083)
	c=3, k=1	-0.13384 (0.00071)	-0.13097 (0.00074)	-0.21211 (0.0006)	-0.16422 (0.00093)	-0.14129 (0.00068)	–	–	-0.14864 (0.00059)	-0.13141 (0.00061)	-0.13893 (0.00051)	-0.22019 (0.00049)	-0.00002 (0.00049)
	c=1.3, k=1.5	-0.26965 (0.00076)	-0.2446 (0.00092)	-0.34059 (0.00077)	-0.25603 (0.00082)	-0.2518 (0.001)	–	–	-0.28897 (0.00091)	-0.26685 (0.00086)	-0.28252 (0.00111)	-0.35963 (0.00078)	-0.00106 (0.00111)
	c=3, k=1.1/3	-0.326 (0.00029)	-0.2553 (0.00044)	-0.04808 (0.00204)	-0.01251 (0.00228)	-0.03079 (0.003)	–	–	-0.31322 (0.00048)	-0.3167 (0.00026)	-0.10906 (0.00538)	-0.36942 (0.00046)	-0.08116 (0.00551)
Lognormal	μ=0.5, σ=0.5	0.0257 (0.00022)	0.02661 (0.00027)	-0.06496 (0.00023)	-0.0132 (0.0004)	0.02003 (0.00022)	0.02533 (0.00027)	–	0.02022 (0.00021)	0.04398 (0.00019)	-0.02193 (0.00034)	-0.10992 (0.00031)	0.0001 (0.00025)
	μ=0.6, σ=1	-0.02155 (0.00034)	-0.01215 (0.00038)	-0.07949 (0.00067)	-0.02806 (0.00056)	-0.03737 (0.00051)	-0.03271 (0.00046)	–	-0.0517 (0.00034)	-0.00934 (0.00037)	-0.05743 (0.00134)	-0.1222 (0.00035)	-0.00019 (0.0005)
	μ=1.5, σ=0.8	0.01514 (0.00031)	0.02054 (0.00045)	0.01392 (0.0005)	-0.01854 (0.00053)	-0.02407 (0.00052)	-0.02146 (0.00048)	–	-0.01902 (0.00034)	0.03832 (0.00043)	0.03677 (0.0006)	-0.04601 (0.00047)	-0.00018 (0.00035)

Table A7: RMSE (Gini, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01291 (0.00042)	0.01552 (0.00043)	0.07083 (0.00031)	0.02085 (0.00049)	0.00689 (0.00045)	–	–	0.00494 (0.0004)	0.01592 (0.00045)	0.00397 (0.00024)	0.08407 (0.00042)	0.00377 (0.00038)
	α=1.5	0.37147 (0.00106)	0.36407 (0.00118)	0.21355 (0.00197)	0.36681 (0.00183)	0.36189 (0.00196)	0.57487 (0.36707)	0.34406 (0.00364)	0.38385 (0.00097)	0.15393 (0.0008)	0.03137 (0.00851)	0.32633 (0.00156)	0.03094 (0.00584)
	α=1.1	0.3839 (0.00386)	0.36704 (0.0042)	0.14398 (0.00802)	0.31794 (0.01262)	0.31742 (0.01294)	0.31699 (0.0125)	0.30284 (0.01371)	0.45298 (0.00401)	0.26112 (0.00302)	0.10771 (0.00519)	0.44732 (0.00452)	0.10155 (0.0043)
Burr	c=1.2, k=2.5	0.425 (0.00097)	0.42164 (0.00104)	0.473 (0.0007)	0.44123 (0.00104)	0.49313 (0.20211)	–	–	0.4355 (0.00085)	0.42456 (0.00089)	0.385 (0.00084)	0.44763 (0.00114)	0.00504 (0.00034)
	c=3, k=1	0.13392 (0.00061)	0.13111 (0.00067)	0.21214 (0.00049)	0.16394 (0.00072)	0.14117 (0.00057)	–	–	0.14869 (0.0005)	0.13144 (0.00062)	0.13874 (0.00052)	0.21997 (0.0006)	0.0039 (0.00056)
	c=1.3, k=1.5	0.2695 (0.00074)	0.24459 (0.00077)	0.34013 (0.00082)	0.25577 (0.00097)	0.25144 (0.00079)	–	–	0.2888 (0.00073)	0.26662 (0.00074)	0.28154 (0.00142)	0.35955 (0.0007)	0.01194 (0.0047)
	c=3, k=1.1/3	0.32607 (0.00022)	0.25539 (0.00029)	0.0527 (0.00116)	0.02622 (0.00192)	0.04164 (0.00188)	–	–	0.31333 (0.00033)	0.31678 (0.00022)	0.11367 (0.00297)	0.36939 (0.00044)	0.08807 (0.00387)
Lognormal	μ=0.5, σ=0.5	0.02576 (0.0002)	0.02666 (0.0002)	0.06503 (0.00023)	0.01357 (0.0003)	0.0201 (0.00019)	0.02538 (0.00022)	–	0.02027 (0.00016)	0.04404 (0.00016)	0.02207 (0.00022)	0.10999 (0.00027)	0.00201 (0.0001)
	μ=0.6, σ=1	0.02193 (0.00025)	0.01323 (0.00034)	0.08006 (0.0006)	0.02906 (0.0005)	0.03815 (0.00047)	0.03343 (0.00041)	–	0.05206 (0.0003)	0.01009 (0.00028)	0.05725 (0.00119)	0.12244 (0.00039)	0.00429 (0.00024)
	μ=1.5, σ=0.8	0.01559 (0.0002)	0.02107 (0.00025)	0.01514 (0.00039)	0.01889 (0.00031)	0.02437 (0.00031)	0.02166 (0.00028)	–	0.01925 (0.00026)	0.03857 (0.0003)	0.03996 (0.00208)	0.04604 (0.00031)	0.00338 (0.00022)

Table A8: MAD (Gini, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.0033 (0.00037) (0.00023)	0.00347 (0.0003) (0.00023)	0.00263 (0.00024) (0.00021)	0.00403 (0.00044) (0.0004)	0.00336 (0.00051) (0.00039)	–	–	0.0028 (0.00037) (0.00029)	0.00358 (0.00039) (0.00029)	0.00271 (0.00015) (0.00008)	0.00362 (0.00049) (0.00022)	0.00242 (0.00023) (0.00015)
	α=1.5	0.0085 (0.00107) (0.00095)	0.00982 (0.00113) (0.00086)	0.01656 (0.00141) (0.00025)	0.01285 (0.0014) (0.00103)	0.01329 (0.0015) (0.0011)	0.01366 (0.00124) (0.00069)	0.01644 (0.00225) (0.00113)	0.008 (0.00106) (0.00078)	0.00614 (0.00058) (0.00053)	0.0132 (0.00172) (0.00122)	0.01159 (0.00135) (0.00109)	0.01432 (0.00168) (0.00133)
	α=1.1	0.0238 (0.00213) (0.00155)	0.0243 (0.00228) (0.00169)	0.04421 (0.01036) (0.00569)	0.05877 (0.01687) (0.00848)	0.05645 (0.02072) (0.00931)	0.04778 (0.02187) (0.01103)	0.0317 (0.02397) (0.00398)	0.02094 (0.00175) (0.00129)	0.01881 (0.0023) (0.00169)	0.02099 (0.00323) (0.00172)	0.0225 (0.00275) (0.00242)	0.02979 (0.00449) (0.00369)
Burr	c=1.2, k=2.5	0.00406 (0.00047) (0.00026)	0.00416 (0.00035) (0.0003)	0.0029 (0.00034) (0.00013)	0.0042 (0.00046) (0.00032)	0.00397 (0.00039) (0.00028)	–	–	0.00359 (0.00044) (0.00029)	0.00428 (0.00047) (0.00032)	0.00298 (0.00031) (0.0002)	0.00546 (0.00076) (0.00029)	0.0033 (0.00039) (0.00019)
	c=3, k=1	0.00355 (0.00043) (0.00006)	0.00369 (0.00043) (0.00026)	0.00268 (0.00027) (0.00018)	0.00401 (0.00053) (0.00022)	0.00355 (0.00044) (0.0002)	–	–	0.00303 (0.00045) (0.0003)	0.00358 (0.00035) (0.00031)	0.00287 (0.00017) (0.00011)	0.00389 (0.0004) (0.00011)	0.00246 (0.00018) (0.00012)
	c=1.3, k=1.5	0.00436 (0.00025) (0.00011)	0.00585 (0.00041) (0.00032)	0.0069 (0.00071) (0.00065)	0.00953 (0.00112) (0.00109)	0.00915 (0.00111) (0.00096)	–	–	0.00485 (0.00029) (0.00023)	0.00437 (0.00028) (0.00016)	0.00869 (0.00075) (0.00029)	0.00495 (0.0003) (0.00025)	0.00631 (0.00087) (0.00044)
	c=3, k=1.1/3	0.00155 (0.0002) (0.00017)	0.0024 (0.00028) (0.00015)	0.0134 (0.00142) (0.00102)	0.01563 (0.00178) (0.00119)	0.02004 (0.00257) (0.00216)	–	–	0.00268 (0.00025) (0.00018)	0.00153 (0.00016) (0.00011)	0.0311 (0.00382) (0.00327)	0.00378 (0.00038) (0.00036)	0.02922 (0.00322) (0.00222)
Lognormal	μ=0.5, σ=0.5	0.00122 (0.0001) (0.00006)	0.00128 (0.00012) (0.0001)	0.00155 (0.00012) (0.00006)	0.00214 (0.00024) (0.00018)	0.00115 (0.00009) (0.00006)	0.00117 (0.00016) (0.00011)	–	0.00102 (0.0001) (0.00007)	0.00127 (0.0001) (0.00006)	0.00158 (0.00016) (0.00011)	0.00161 (0.0002) (0.00015)	0.00136 (0.00012) (0.00004)
	μ=0.6, σ=1	0.00238 (0.00023) (0.00021)	0.00306 (0.00042) (0.00017)	0.00459 (0.00057) (0.00026)	0.00409 (0.00062) (0.00025)	0.00385 (0.00063) (0.00038)	0.0034 (0.00061) (0.00034)	–	0.00282 (0.00036) (0.00017)	0.00245 (0.00024) (0.00022)	0.00572 (0.00076) (0.00058)	0.00344 (0.00044) (0.00015)	0.00306 (0.00024) (0.00014)
	μ=1.5, σ=0.8	0.00211 (0.00025) (0.00009)	0.00258 (0.00031) (0.0003)	0.00338 (0.00044) (0.00021)	0.00293 (0.0003) (0.00024)	0.00302 (0.00032) (0.00025)	0.00269 (0.0003) (0.00015)	–	0.00239 (0.00029) (0.00012)	0.00203 (0.00027) (0.00011)	0.00406 (0.00031) (0.00021)	0.00272 (0.00028) (0.00027)	0.00225 (0.00017) (0.00017)

TABLES OBTAINED FOR THE GINI COEFFICIENT FOR n=5 000

Table A9: Mean – true Gini (Gini, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.0113 (0.00077)	0.01548 (0.0006)	-0.07039 (0.00047)	-0.01931 (0.0007)	0.00675 (0.00073)	–	–	-0.00283 (0.00058)	0.01449 (0.00083)	0.00051 (0.0006)	-0.08429 (0.0007)	-0.00002 (0.00045)
	α=1.5	-0.37275 (0.00142)	-0.36335 (0.00173)	-0.20765 (0.00334)	-0.36091 (0.00386)	-0.35391 (0.20406)	-0.34244 (0.12945)	-0.32618 (0.00769)	-0.38396 (0.00136)	-0.15508 (0.00103)	-0.00533 (0.00428)	-0.32761 (0.00278)	-0.00587 (0.00233)
	α=1.1	-0.39439 (0.00373)	-0.37376 (0.00391)	-0.11123 (0.00801)	-0.27586 (0.01049)	-0.275 (0.00984)	-0.27215 (0.01063)	-0.20576 (0.006)	-0.45821 (0.00405)	-0.26954 (0.00332)	-0.10118 (0.00483)	-0.45255 (0.00475)	-0.09214 (0.00562)

Burr	c=1.2, k=2.5	-0.42564 (0.00079)	-0.42056 (0.00073)	-0.4725 (0.00052)	-0.43972 (0.00073)	-0.42676 (0.14558)	–	–	-0.43533 (0.00064)	-0.42518 (0.00075)	-0.38434 (0.00072)	-0.44777 (0.00106)	-0.00016 (0.00076)
	c=3, k=1	-0.13446 (0.0005)	-0.13001 (0.00045)	-0.21164 (0.00038)	-0.1627 (0.00055)	-0.13919 (0.00076)	–	–	-0.14839 (0.00031)	-0.13199 (0.00044)	-0.13801 (0.00058)	-0.22011 (0.00068)	-0.00002 (0.0006)
	c=1.3, k=1.5	-0.26961 (0.00094)	-0.24492 (0.00142)	-0.34045 (0.0018)	-0.25593 (0.00256)	-0.25132 (0.00245)	–	–	-0.28899 (0.0011)	-0.26674 (0.00102)	-0.28211 (0.00245)	-0.36001 (0.00121)	-0.0009 (0.00179)
	c=3, k=1.1/3	-0.32637 (0.00028)	-0.25576 (0.0004)	-0.05041 (0.00248)	-0.01338 (0.00253)	-0.02898 (0.00292)	–	–	-0.31374 (0.00046)	-0.31717 (0.00027)	-0.11438 (0.00598)	-0.37007 (0.00067)	-0.08091 (0.00521)
Lognormal	μ=0.5, σ=0.5	0.02554 (0.00025)	0.02704 (0.00026)	-0.06514 (0.00029)	-0.01228 (0.00041)	0.02014 (0.00025)	0.02562 (0.00042)	–	0.02013 (0.00021)	0.04384 (0.00024)	-0.02165 (0.00023)	-0.11021 (0.00034)	-0.00011 (0.0003)
	μ=0.6, σ=1	-0.02181 (0.00042)	-0.01258 (0.00056)	-0.07993 (0.00086)	-0.02852 (0.00077)	-0.03786 (0.00072)	-0.03312 (0.00062)	–	-0.05208 (0.00049)	-0.00956 (0.00042)	-0.05625 (0.00147)	-0.12259 (0.00066)	-0.00004 (0.00041)
	μ=1.5, σ=0.8	0.01494 (0.00045)	0.0204 (0.00054)	0.01408 (0.00071)	-0.01862 (0.00057)	-0.02428 (0.00059)	-0.02159 (0.00056)	–	-0.01924 (0.0005)	0.03817 (0.00046)	0.03821 (0.00125)	-0.04622 (0.00049)	-0.00032 (0.00052)

Table A10: Median – true Gini (Gini, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.011 (0.00105)	0.01514 (0.00077)	-0.07061 (0.00054)	-0.01975 (0.00062)	0.00648 (0.00092)	–	–	-0.00295 (0.00059)	0.01405 (0.00104)	0.00026 (0.00064)	-0.08451 (0.00097)	-0.00036 (0.00052)
	α=1.5	-0.37434 (0.00214)	-0.36521 (0.00228)	-0.21178 (0.00436)	-0.36607 (0.00366)	-0.36031 (0.00533)	-0.35321 (0.00397)	-0.34258 (0.00237)	-0.38574 (0.00187)	-0.15607 (0.00148)	-0.01171 (0.00426)	-0.32877 (0.00302)	-0.01404 (0.00282)
	α=1.1	-0.39495 (0.00643)	-0.37238 (0.00802)	-0.08583 (0.01343)	-0.2511 (0.02063)	-0.2537 (0.01995)	-0.25391 (0.01125)	-0.2249 (0.0052)	-0.45604 (0.00767)	-0.27048 (0.00414)	-0.1105 (0.00533)	-0.4506 (0.00833)	-0.10472 (0.00597)
Burr	c=1.2, k=2.5	-0.42552 (0.0008)	-0.42054 (0.00058)	-0.47263 (0.00058)	-0.44014 (0.00061)	-0.42741 (0.00099)	–	–	-0.43531 (0.00063)	-0.42519 (0.00078)	-0.38463 (0.00089)	-0.44721 (0.00154)	-0.00047 (0.00091)
	c=3, k=1	-0.1345 (0.00051)	-0.12983 (0.00044)	-0.21152 (0.0004)	-0.1626 (0.00058)	-0.13943 (0.00081)	–	–	-0.14847 (0.00021)	-0.13193 (0.00042)	-0.13836 (0.00058)	-0.22035 (0.00086)	-0.00026 (0.00067)
	c=1.3, k=1.5	-0.26967 (0.00103)	-0.24494 (0.00146)	-0.34103 (0.00188)	-0.25708 (0.00295)	-0.25257 (0.00275)	–	–	-0.28923 (0.0011)	-0.2665 (0.00113)	-0.2856 (0.00201)	-0.36001 (0.00163)	-0.00347 (0.0024)
	c=3, k=1.1/3	-0.32624 (0.00047)	-0.25554 (0.00055)	-0.05036 (0.00306)	-0.01383 (0.00361)	-0.03242 (0.0041)	–	–	-0.31349 (0.00075)	-0.31713 (0.00044)	-0.12399 (0.00776)	-0.3696 (0.00098)	-0.0959 (0.00525)
Lognormal	μ=0.5, σ=0.5	0.02541 (0.00028)	0.02692 (0.00033)	-0.06537 (0.00036)	-0.0123 (0.00047)	0.02005 (0.00024)	0.0256 (0.00051)	–	0.02004 (0.00024)	0.04383 (0.00037)	-0.02187 (0.00033)	-0.11029 (0.00043)	-0.00008 (0.00046)
	μ=0.6, σ=1	-0.02181 (0.00061)	-0.01282 (0.00089)	-0.08032 (0.00142)	-0.02891 (0.00123)	-0.03825 (0.00111)	-0.03358 (0.00081)	–	-0.05209 (0.00073)	-0.00957 (0.00052)	-0.05943 (0.00192)	-0.12258 (0.00083)	-0.00025 (0.00065)
	μ=1.5, σ=0.8	0.01518 (0.00043)	0.0206 (0.00067)	0.014 (0.00097)	-0.01855 (0.00085)	-0.02407 (0.00093)	-0.02137 (0.00079)	–	-0.01892 (0.0005)	0.03838 (0.00053)	0.0358 (0.00097)	-0.04609 (0.00074)	-0.00033 (0.00058)

Table A11: RMSE (Gini, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01329 (0.00066)	0.01696 (0.00053)	0.07059 (0.00047)	0.02077 (0.00068)	0.00931 (0.00053)	–	–	0.00617 (0.00043)	0.01629 (0.00072)	0.00578 (0.00034)	0.08464 (0.00069)	0.00508 (0.00032)
	α=1.5	0.37315 (0.00139)	0.36384 (0.00169)	0.21051 (0.00299)	0.36248 (0.00344)	1.64968 (0.61145)	0.93129 (0.55263)	0.32972 (0.0055)	0.38429 (0.00133)	0.15562 (0.001)	0.03617 (0.00644)	0.32849 (0.00274)	0.03716 (0.0032)

	α=1.1	0.39694 (0.00373)	0.37659 (0.00396)	0.13817 (0.00873)	0.30224 (0.0117)	0.30233 (0.01097)	0.29676 (0.00973)	0.21425 (0.00469)	0.45995 (0.004)	0.27179 (0.00332)	0.11388 (0.00383)	0.45474 (0.00473)	0.11255 (0.00464)
Burr	c=1.2, k=2.5	0.42572 (0.00079)	0.42065 (0.00072)	0.47254 (0.00052)	0.4398 (0.00073)	0.90803 (0.56619)	–	–	0.43539 (0.00064)	0.42526 (0.00074)	0.38439 (0.00071)	0.44789 (0.00106)	0.00693 (0.0006)
	c=3, k=1	0.13464 (0.00049)	0.13021 (0.00045)	0.21169 (0.00038)	0.16288 (0.00055)	0.13935 (0.00075)	–	–	0.14848 (0.00031)	0.1322 (0.00045)	0.13812 (0.00058)	0.22023 (0.00068)	0.00558 (0.00055)
	c=1.3, k=1.5	0.26974 (0.00093)	0.2452 (0.0014)	0.34074 (0.00178)	0.25667 (0.00251)	0.25204 (0.0024)	–	–	0.28914 (0.00109)	0.26688 (0.00101)	0.283 (0.00205)	0.36014 (0.0012)	0.0157 (0.00414)
	c=3, k=1.1/3	0.32636 (0.00028)	0.25578 (0.0004)	0.05753 (0.00218)	0.03443 (0.00212)	0.05084 (0.00276)	–	–	0.31376 (0.00046)	0.31716 (0.00027)	0.1278 (0.00607)	0.37013 (0.00067)	0.10143 (0.00394)
Lognormal	μ=0.5, σ=0.5	0.02565 (0.00025)	0.02718 (0.00027)	0.06521 (0.00029)	0.01293 (0.00037)	0.02028 (0.00026)	0.02572 (0.00042)	–	0.02023 (0.00021)	0.04391 (0.00024)	0.02188 (0.00022)	0.11026 (0.00034)	0.00292 (0.00028)
	μ=0.6, σ=1	0.0224 (0.00042)	0.01429 (0.00056)	0.08056 (0.00083)	0.02999 (0.00074)	0.03885 (0.0007)	0.03401 (0.00059)	–	0.05242 (0.00049)	0.01082 (0.0004)	0.05861 (0.00185)	0.1228 (0.00066)	0.00611 (0.00041)
	μ=1.5, σ=0.8	0.01559 (0.00043)	0.02108 (0.00053)	0.01574 (0.00073)	0.01961 (0.00054)	0.0251 (0.00057)	0.02233 (0.00054)	–	0.01989 (0.00049)	0.03843 (0.00046)	0.04016 (0.00208)	0.04653 (0.00051)	0.00455 (0.00028)

Table A12: MAD (Gini, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
	α=3	0.00474 (0.00041) (0.0003)	0.00474 (0.0005) (0.0003)	0.0036 (0.00028) (0.00018)	0.00526 (0.00056) (0.00043)	0.00486 (0.00044) (0.00032)	–	–	0.00373 (0.00042) (0.00039)	0.00494 (0.00048) (0.00033)	0.00386 (0.00038) (0.00018)	0.00529 (0.00048) (0.00016)	0.00352 (0.00019) (0.00012)
Pareto	α=1.5	0.0118 (0.00176) (0.00133)	0.0121 (0.00174) (0.00155)	0.01962 (0.00279) (0.00254)	0.01601 (0.00269) (0.00136)	0.01611 (0.00247) (0.0013)	0.01543 (0.00184) (0.0015)	0.00997 (0.00194) (0.0011)	0.01076 (0.00141) (0.001)	0.00892 (0.00111) (0.00076)	0.01787 (0.00215) (0.00105)	0.01587 (0.0028) (0.00181)	0.01782 (0.00266) (0.00238)
	α=1.1	0.03466 (0.00359) (0.00188)	0.03659 (0.00336) (0.00247)	0.05906 (0.01057) (0.01054)	0.09567 (0.00979) (0.00701)	0.09818 (0.00664) (0.00379)	0.09714 (0.00582) (0.00417)	0.03347 (0.00624) (0.0045)	0.03136 (0.00255) (0.00223)	0.02713 (0.00233) (0.00124)	0.02859 (0.00438) (0.00127)	0.03342 (0.00209) (0.00143)	0.03647 (0.00316) (0.00233)
	c=1.2, k=2.5	0.00553 (0.00078) (0.0003)	0.00581 (0.00061) (0.00043)	0.00399 (0.00043) (0.00035)	0.00587 (0.00085) (0.00049)	0.0054 (0.00085) (0.00033)	–	–	0.00474 (0.00056) (0.00018)	0.00583 (0.00082) (0.00076)	0.00397 (0.00048) (0.00018)	0.007 (0.00087) (0.00067)	0.00469 (0.00066) (0.00041)
	c=3, k=1	0.00501 (0.00055) (0.0004)	0.00527 (0.0004) (0.0002)	0.00396 (0.00034) (0.00021)	0.00541 (0.00067) (0.00053)	0.00509 (0.00064) (0.00027)	–	–	0.00419 (0.00053) (0.0004)	0.00533 (0.00052) (0.00044)	0.00409 (0.00055) (0.00027)	0.00549 (0.00066) (0.00041)	0.0037 (0.00039) (0.00017)
Burr	c=1.3, k=1.5	0.0061 (0.00055) (0.00022)	0.00817 (0.00067) (0.00038)	0.0101 (0.00098) (0.00067)	0.01351 (0.00188) (0.00135)	0.01291 (0.00163) (0.00105)	–	–	0.00671 (0.00064) (0.00032)	0.00634 (0.00045) (0.00042)	0.01096 (0.00109) (0.00065)	0.00696 (0.00066) (0.00057)	0.00852 (0.00118) (0.00039)
	c=3, k=1.1/3	0.00221 (0.00029) (0.00025)	0.0035 (0.00052) (0.00046)	0.02006 (0.00236) (0.00068)	0.0226 (0.00295) (0.00209)	0.02833 (0.00464) (0.00321)	–	–	0.00382 (0.00048) (0.00034)	0.00219 (0.00021) (0.00009)	0.0348 (0.00472) (0.00247)	0.00537 (0.00061) (0.00055)	0.03395 (0.00279) (0.00173)
Lognormal	μ=0.5, σ=0.5	0.00167 (0.00027) (0.00019)	0.00168 (0.00019) (0.00006)	0.00208 (0.00027) (0.00016)	0.00257 (0.00032) (0.00016)	0.00155 (0.00015) (0.00005)	0.00156 (0.00026) (0.00011)	–	0.00138 (0.00024) (0.00008)	0.00168 (0.00022) (0.00016)	0.00208 (0.00018) (0.0001)	0.00225 (0.00031) (0.00016)	0.00195 (0.00019) (0.00009)
	μ=0.6, σ=1	0.00351 (0.0005) (0.0004)	0.00464 (0.00065) (0.00055)	0.0068 (0.00068) (0.00058)	0.00646 (0.00073) (0.00055)	0.00607 (0.0008) (0.00051)	0.00538 (0.00064) (0.0005)	–	0.00411 (0.00058) (0.0004)	0.00348 (0.00056) (0.00039)	0.00745 (0.00041) (0.00023)	0.00503 (0.00065) (0.00063)	0.0041 (0.00035) (0.00021)

	μ=1.5, σ=0.8	0.00295 (0.00025) (0.00018)	0.00367 (0.00046) (0.00036)	0.00492 (0.0006) (0.00041)	0.00411 (0.00053) (0.00029)	0.00426 (0.00053) (0.00026)	0.004 (0.00048) (0.00027)	–	0.00339 (0.00034) (0.00028)	0.00287 (0.00019) (0.00012)	0.00525 (0.00067) (0.00042)	0.00371 (0.00045) (0.0003)	0.00304 (0.00038) (0.0003)

TABLES OBTAINED FOR THE GINI COEFFICIENT FOR n=1 000

Table A13: Mean – true Gini (Gini, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01114 (0.0016)	0.02036 (0.00173)	-0.06685 (0.00128)	-0.01371 (0.00173)	0.02657 (0.00372)	–	–	-0.00109 (0.00145)	0.01439 (0.0017)	0.00389 (0.00121)	-0.08134 (0.00124)	-0.00032 (0.00092)
	α=1.5	-0.39015 (0.00313)	-0.37368 (0.00371)	-0.20876 (0.00753)	-0.3534 (0.01123)	-0.28231 (0.13257)	-0.26718 (0.34953)	–	-0.39529 (0.00288)	-0.16829 (0.00221)	-0.01308 (0.00474)	-0.35486 (0.00325)	-0.00587 (0.00233)
	α=1.1	-0.4814 (0.00405)	-0.45226 (0.00426)	-0.16349 (0.01046)	-0.34984 (0.01332)	-0.32927 (0.00977)	-0.09846 (0.02566)	–	-0.52205 (0.00427)	-0.33683 (0.00309)	-0.14516 (0.00472)	-0.52978 (0.00455)	-0.12166 (0.00898)
Burr	c=1.2, k=2.5	-0.4246 (0.00174)	-0.41428 (0.00236)	-0.46825 (0.00174)	-0.43062 (0.00298)	-0.39184 (0.00794)	–	–	-0.43231 (0.0017)	-0.42421 (0.00194)	-0.38014 (0.00174)	-0.44403 (0.00263)	-0.00078 (0.00142)
	c=3, k=1	-0.13459 (0.00183)	-0.12507 (0.00191)	-0.20805 (0.00151)	-0.15691 (0.00206)	-0.11713 (0.00521)	–	–	-0.14659 (0.00158)	-0.13223 (0.00172)	-0.13475 (0.0017)	-0.21719 (0.00157)	-0.00073 (0.00129)
	c=1.3, k=1.5	-0.27014 (0.00198)	-0.24512 (0.002)	-0.33889 (0.0022)	-0.25198 (0.00308)	-0.24336 (0.00369)	–	–	-0.28907 (0.00184)	-0.26718 (0.00205)	-0.283 (0.00256)	-0.36061 (0.00186)	-0.00224 (0.00248)
	c=3, k=1.1/3	-0.32833 (0.00074)	-0.25918 (0.00104)	-0.06597 (0.00534)	-0.03083 (0.00612)	-0.05046 (0.00643)	–	–	-0.31717 (0.00118)	-0.319 (0.00066)	-0.15198 (0.00677)	-0.37467 (0.00177)	-0.11863 (0.00951)
Lognormal	μ=0.5, σ=0.5	0.02485 (0.0005)	0.03084 (0.00067)	-0.064 (0.00067)	-0.00856 (0.00089)	0.02252 (0.00064)	0.02989 (0.00176)	–	0.02012 (0.0004)	0.04307 (0.0005)	-0.01929 (0.00078)	-0.11075 (0.00061)	-0.00045 (0.00032)
	μ=0.6, σ=1	-0.02374 (0.00134)	-0.01087 (0.00136)	-0.07194 (0.00199)	-0.02262 (0.00177)	-0.03363 (0.0016)	-0.02951 (0.00162)	–	-0.05224 (0.0014)	-0.01172 (0.00123)	-0.05093 (0.00229)	-0.12268 (0.00163)	-0.0003 (0.00092)
	μ=1.5, σ=0.8	0.01184 (0.00148)	0.02384 (0.00105)	0.039 (0.00454)	-0.0101 (0.00148)	-0.01847 (0.00123)	-0.01826 (0.00135)	–	-0.01922 (0.00123)	0.03495 (0.00134)	0.05457 (0.00528)	-0.04621 (0.00101)	-0.00069 (0.00078)

Table A14: Median – true Gini (Gini, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01099 (0.00147)	0.02013 (0.00155)	-0.06711 (0.00144)	-0.01386 (0.00218)	0.02252 (0.00234)	–	–	-0.00092 (0.00103)	0.01442 (0.00194)	0.00331 (0.00129)	-0.08235 (0.00146)	-0.00116 (0.00117)
	α=1.5	-0.39291 (0.00289)	-0.37693 (0.00427)	-0.21819 (0.01242)	-0.37693 (0.00917)	-0.33497 (0.02071)	-0.27198 (0.01546)	–	-0.39823 (0.00267)	-0.17144 (0.00138)	-0.02373 (0.00575)	-0.35984 (0.00335)	-0.01404 (0.00282)
	α=1.1	-0.48428 (0.00338)	-0.4577 (0.00599)	-0.19744 (0.02041)	-0.42672 (0.01799)	-0.43337 (0.03125)	-0.11267 (0.03557)	–	-0.52894 (0.00506)	-0.33972 (0.00324)	-0.15882 (0.00439)	-0.53381 (0.00598)	-0.13744 (0.01257)
Burr	c=1.2, k=2.5	-0.4251 (0.00208)	-0.41486 (0.00214)	-0.46947 (0.00155)	-0.43317 (0.0028)	-0.3994 (0.0033)	–	–	-0.43277 (0.0021)	-0.42493 (0.00208)	-0.38245 (0.0018)	-0.44724 (0.00345)	-0.00134 (0.00161)
	c=3, k=1	-0.13506 (0.00231)	-0.12564 (0.00194)	-0.20898 (0.00146)	-0.15855 (0.00213)	-0.12086 (0.00399)	–	–	-0.1466 (0.00198)	-0.13257 (0.00195)	-0.13643 (0.00151)	-0.21955 (0.00172)	-0.00151 (0.00101)
	c=1.3, k=1.5	-0.26975 (0.00291)	-0.24495 (0.00202)	-0.34088 (0.00267)	-0.25629 (0.00388)	-0.25091 (0.00445)	–	–	-0.28847 (0.00237)	-0.26668 (0.00297)	-0.28802 (0.00242)	-0.36071 (0.00206)	-0.00528 (0.00242)

	c=3, k=1.1/3	-0.32792 (0.00086)	-0.25841 (0.00144)	-0.0634 (0.00701)	-0.02874 (0.00742)	-0.04776 (0.00597)	–	–	-0.31664 (0.00146)	-0.31886 (0.0007)	-0.16368 (0.00756)	-0.37422 (0.00234)	-0.13193 (0.00793)
Lognormal	μ=0.5, σ=0.5	0.0248 (0.00069)	0.03051 (0.00087)	-0.06389 (0.00085)	-0.0082 (0.00128)	0.0225 (0.00102)	0.02931 (0.00167)	–	0.01998 (0.00046)	0.04309 (0.00065)	-0.01953 (0.00107)	-0.11095 (0.00101)	-0.00045 (0.00045)
	μ=0.6, σ=1	-0.0226 (0.00155)	-0.01119 (0.00187)	-0.07567 (0.00296)	-0.02474 (0.00285)	-0.03539 (0.00256)	-0.03146 (0.00241)	–	-0.05229 (0.00191)	-0.01084 (0.00198)	-0.05737 (0.00227)	-0.12225 (0.00172)	-0.00047 (0.00116)
	μ=1.5, σ=0.8	0.01255 (0.00179)	0.02407 (0.00129)	0.02317 (0.00184)	-0.01253 (0.00134)	-0.01938 (0.00132)	-0.01875 (0.00169)	–	-0.01919 (0.00169)	0.03556 (0.00163)	0.04048 (0.0027)	-0.04625 (0.00127)	-0.0006 (0.00119)

Table A15: RMSE (Gini, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01815 (0.00122)	0.02579 (0.00148)	0.06798 (0.00124)	0.02213 (0.00122)	0.03411 (0.00855)	–	–	0.0127 (0.00074)	0.02084 (0.00154)	0.0132 (0.00105)	0.08263 (0.00123)	0.01147 (0.00099)
	α=1.5	0.39111 (0.00304)	0.37495 (0.0036)	0.21764 (0.00675)	0.36689 (0.02778)	0.9746 (0.58232)	0.52633 (0.8139)	–	0.39615 (0.00278)	0.16947 (0.00204)	0.06103 (0.00732)	0.35698 (0.00308)	0.03716 (0.0032)
	α=1.1	0.48459 (0.00389)	0.45579 (0.00417)	0.20423 (0.02749)	0.38788 (0.01152)	0.37037 (0.01018)	0.11081 (0.02524)	–	0.52449 (0.00425)	0.33952 (0.00299)	0.1607 (0.00324)	0.53302 (0.00457)	0.14693 (0.00677)
Burr	c=1.2, k=2.5	0.42494 (0.00172)	0.41471 (0.00232)	0.46844 (0.00172)	0.43117 (0.00289)	0.39396 (0.00595)	–	–	0.43259 (0.00167)	0.42456 (0.00192)	0.38046 (0.00171)	0.44442 (0.00259)	0.01452 (0.00125)
	c=3, k=1	0.13538 (0.00172)	0.12614 (0.00174)	0.20841 (0.00145)	0.15793 (0.0019)	0.11901 (0.0041)	–	–	0.14715 (0.0015)	0.1331 (0.00163)	0.13541 (0.00156)	0.21774 (0.00151)	0.01202 (0.00087)
	c=1.3, k=1.5	0.27093 (0.00204)	0.24657 (0.00204)	0.34043 (0.00219)	0.25615 (0.00291)	0.24874 (0.00276)	–	–	0.28992 (0.00188)	0.268 (0.00209)	0.28546 (0.00247)	0.36136 (0.00186)	0.02809 (0.00334)
	c=3, k=1.1/3	0.32837 (0.00074)	0.25934 (0.00104)	0.08479 (0.00404)	0.06549 (0.00354)	0.08615 (0.00455)	–	–	0.31733 (0.00119)	0.31905 (0.00066)	0.17161 (0.0046)	0.37495 (0.00177)	0.13906 (0.00638)
Lognormal	μ=0.5, σ=0.5	0.02542 (0.00053)	0.0316 (0.00064)	0.06441 (0.00066)	0.01235 (0.0008)	0.02304 (0.00064)	0.03032 (0.00176)	–	0.02062 (0.00043)	0.04343 (0.00052)	0.02097 (0.00062)	0.11102 (0.00061)	0.00643 (0.00053)
	μ=0.6, σ=1	0.02712 (0.00122)	0.01739 (0.00089)	0.07566 (0.00189)	0.03035 (0.00145)	0.03825 (0.00137)	0.03359 (0.00121)	–	0.05387 (0.00132)	0.01759 (0.0011)	0.06049 (0.0026)	0.12371 (0.00158)	0.01255 (0.00062)
	μ=1.5, σ=0.8	0.01738 (0.00128)	0.02681 (0.00098)	0.06027 (0.00894)	0.02099 (0.00158)	0.02407 (0.00089)	0.02197 (0.00097)	–	0.02269 (0.00109)	0.03708 (0.00136)	0.07259 (0.0109)	0.04794 (0.00092)	0.01093 (0.0007)

Table A16: MAD (Gini, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00929 (0.00087) (0.00057)	0.01069 (0.00089) (0.00048)	0.00828 (0.00079) (0.00068)	0.01208 (0.00082) (0.00056)	0.00961 (0.00232) (0.00245)	–	–	0.00851 (0.0008) (0.00042)	0.01039 (0.00078) (0.00067)	0.00823 (0.00073) (0.00057)	0.01049 (0.00104) (0.0005)	0.00705 (0.00083) (0.00046)
	α=1.5	0.01808 (0.00217) (0.00194)	0.02143 (0.00177) (0.00067)	0.04292 (0.00375) (0.00325)	0.03409 (0.00744) (0.00562)	0.04656 (0.01225) (0.01077)	0.02263 (0.00719) (0.00341)	–	0.01756 (0.00218) (0.00149)	0.01301 (0.00204) (0.00097)	0.03127 (0.00256) (0.00147)	0.02632 (0.00378) (0.00247)	0.01782 (0.00266) (0.00238)
	α=1.1	0.03717 (0.00395) (0.00271)	0.03424 (0.00332) (0.00233)	0.07277 (0.01087) (0.00472)	0.08334 (0.01751) (0.00902)	0.05703 (0.02655) (0.01129)	0.03006 (0.03035) (0.02029)	–	0.03044 (0.00231) (0.00182)	0.02906 (0.00325) (0.00223)	0.03789 (0.00539) (0.00281)	0.03632 (0.00265) (0.00238)	0.05008 (0.00827) (0.00422)

Burr	c=1.2, k=2.5	0.0109 (0.00065) (0.00048)	0.01236 (0.00087) (0.00047)	0.00868 (0.0003) (0.00012)	0.01388 (0.00105) (0.00079)	0.0082 (0.0028) (0.00143)	–	–	0.00996 (0.0006) (0.00039)	0.01129 (0.00101) (0.00072)	0.00867 (0.00097) (0.00078)	0.01358 (0.00193) (0.00141)	0.00941 (0.00098) (0.00059)
	c=3, k=1	0.00997 (0.00128) (0.00068)	0.0112 (0.00129) (0.00101)	0.00867 (0.001) (0.00077)	0.01263 (0.00152) (0.00091)	0.00955 (0.00268) (0.00114)	–	–	0.00882 (0.00109) (0.00046)	0.01058 (0.00113) (0.00072)	0.0084 (0.00101) (0.00061)	0.01093 (0.00074) (0.00058)	0.008 (0.00061) (0.00022)
	c=1.3, k=1.5	0.01392 (0.00129) (0.00063)	0.0181 (0.00213) (0.00136)	0.0217 (0.00282) (0.0018)	0.02904 (0.00351) (0.00156)	0.02776 (0.00459) (0.00275)	–	–	0.0151 (0.00195) (0.00053)	0.01444 (0.00145) (0.00129)	0.02094 (0.00225) (0.00121)	0.01569 (0.00221) (0.00071)	0.01652 (0.00215) (0.00095)
	c=3, k=1.1/3	0.00467 (0.00059) (0.00042)	0.00696 (0.00099) (0.00042)	0.03958 (0.00433) (0.00222)	0.04344 (0.00524) (0.00386)	0.04967 (0.00811) (0.00556)	–	–	0.00783 (0.00094) (0.0005)	0.0046 (0.00075) (0.00055)	0.0481 (0.00642) (0.00612)	0.01053 (0.00134) (0.00089)	0.04403 (0.00372) (0.00216)
Lognormal	μ=0.5, σ=0.5	0.00377 (0.00028) (0.0002)	0.00437 (0.00047) (0.00029)	0.00487 (0.00052) (0.00018)	0.00604 (0.00057) (0.00038)	0.00308 (0.00049) (0.00021)	0.00344 (0.00115) (0.00047)	–	0.00318 (0.00019) (0.0001)	0.00391 (0.00051) (0.00042)	0.00532 (0.0006) (0.00034)	0.0053 (0.00064) (0.00042)	0.00444 (0.0005) (0.00024)
	μ=0.6, σ=1	0.00842 (0.00118) (0.00096)	0.0088 (0.00104) (0.00095)	0.01491 (0.00178) (0.00116)	0.01313 (0.00162) (0.0007)	0.01195 (0.00169) (0.00123)	0.01041 (0.0015) (0.00089)	–	0.00877 (0.00092) (0.00072)	0.00872 (0.00095) (0.00096)	0.01368 (0.00178) (0.00099)	0.01049 (0.00103) (0.00084)	0.00864 (0.00074) (0.00074)
	μ=1.5, σ=0.8	0.00919 (0.00111) (0.00063)	0.00836 (0.00084) (0.00079)	0.01574 (0.00203) (0.00115)	0.01127 (0.00086) (0.00052)	0.0101 (0.00127) (0.00077)	0.008 (0.001) (0.00059)	–	0.00817 (0.00096) (0.0008)	0.00919 (0.00097) (0.0006)	0.0135 (0.00249) (0.00161)	0.00887 (0.00057) (0.0002)	0.00712 (0.0009) (0.00043)

TABLES OBTAINED FOR THE QSR FOR n=15 000

Table A17: Mean – true QSR (QSR, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00654 (0.00248)	1.0372 (0.0025)	-0.13867 (0.00159)	0.52445 (0.00441)	0.90292 (0.00311)	–	–	0.8045 (0.00151)	1.00636 (0.00289)	-0.00014 (0.00248)	-0.70373 (0.0025)	0.00069 (0.00214)
	α=1.5	-2.3448 (0.02532)	-2.03141 (0.03695)	6.06403 (0.12977)	-2.10741 (0.097)	-1.86054 (0.13667)	-1.58809 (0.20125)	-0.76224 (0.60452)	-2.87296 (0.0255)	-1.81563 (0.0143)	-0.07782 (0.08565)	-5.74581 (0.01281)	-0.03869 (0.13641)
	α=1.1	-30.75903 (0.11029)	-30.01121 (0.12399)	-6.65243 (1.33767)	-24.73347 (1.61588)	-24.42796 (1.41053)	-24.80685 (1.06268)	-24.92392 (0.69188)	-33.634 (0.07246)	-27.26206 (0.10953)	-17.68044 (0.5183)	-37.17755 (0.0543)	-7.50368 (7.20897)
Burr	c=1.2, k=2.5	-21.64391 (0.00699)	-21.57566 (0.00898)	-22.88023 (0.00567)	-22.09686 (0.00958)	-21.78582 (0.00785)	–	–	-21.92219 (0.00605)	-21.64381 (0.00936)	-23.38224 (0.00267)	-23.8389 (0.00364)	0.01034 (0.05982)
	c=3, k=1	-1.92371 (0.00436)	-1.88893 (0.00416)	-3.08371 (0.0028)	-2.40461 (0.00564)	-2.03683 (0.00394)	–	–	-2.13935 (0.00248)	-1.92307 (0.00527)	-3.04147 (0.00278)	-3.71099 (0.00355)	-0.00127 (0.0078)
	c=1.3, k=1.5	-26.85829 (0.00551)	-26.45857 (0.0096)	-28.08034 (0.0091)	-26.72407 (0.01754)	-26.59284 (0.01773)	–	–	-27.17227 (0.00623)	-26.85879 (0.00624)	-27.70714 (0.01651)	-28.56761 (0.00575)	-0.02476 (0.13435)
	c=3, k=1.1/3	-57.02517 (0.00656)	-50.3081 (0.02344)	-3.51373 (0.87196)	38.76335 (2.18209)	25.32616 (2.60263)	–	–	-56.05616 (0.01646)	-61.13099 (0.0124)	-43.45077 (0.79953)	-62.94934 (0.01707)	-19.1564 (8.44886)
Lognormal	μ=0.5, σ=0.5	1.22832 (0.00238)	1.27636 (0.00419)	-0.06953 (0.0017)	0.888 (0.00392)	0.82034 (0.00483)	1.19281 (0.00257)	–	0.64965 (0.00278)	-0.2479 (0.00162)	-0.95565 (0.00278)	-1.92146 (0.00109)	0.00184 (0.00344)
	μ=0.6, σ=1	-2.27171 (0.02983)	-1.41616 (0.04133)	-6.27875 (0.03951)	-2.87247 (0.05078)	-3.24069 (0.04659)	-3.52788 (0.04443)	–	-4.6554 (0.03034)	-7.93977 (0.01707)	-9.14432 (0.04892)	-11.15914 (0.01535)	-0.00133 (0.02059)
	μ=1.5, σ=0.8	4.52941 (0.00713)	4.93175 (0.01194)	3.41233 (0.0182)	0.03889 (0.01164)	-0.17585 (0.01079)	-0.36478 (0.01141)	–	2.15287 (0.00702)	2.86421 (0.00949)	2.89418 (0.06132)	-3.16325 (0.0074)	0.00234 (0.00813)

Table A18: Median – true QSR (QSR, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00648 (0.00442)	1.03664 (0.00345)	-0.13899 (0.00205)	0.5247 (0.00664)	0.90187 (0.0047)	–	–	0.80312 (0.00301)	1.00637 (0.00461)	-0.00015 (0.00328)	-0.70277 (0.00396)	-0.00002 (0.00333)
	α=1.5	-2.36003 (0.04603)	-2.06193 (0.04288)	5.80954 (0.11207)	-2.29574 (0.0737)	-2.12643 (0.08123)	-1.99848 (0.10404)	-1.89401 (0.15119)	-2.89806 (0.03929)	-1.82527 (0.0239)	-0.27783 (0.06513)	-5.75754 (0.01818)	-0.25385 (0.09685)
	α=1.1	-30.75205 (0.09814)	-30.02866 (0.13543)	-9.97077 (0.66327)	-27.8107 (0.35037)	-27.8165 (0.30922)	-27.911 (0.25481)	-28.06868 (0.20761)	-33.65319 (0.09609)	-27.27338 (0.15248)	-19.259 (0.28436)	-37.18028 (0.06615)	-17.50663 (0.7119)
Burr	c=1.2, k=2.5	-21.64315 (0.00837)	-21.58351 (0.01247)	-22.88431 (0.00886)	-22.10577 (0.01469)	-21.79568 (0.01221)	–	–	-21.92612 (0.00831)	-21.64268 (0.00959)	-23.38355 (0.00354)	-23.8406 (0.00554)	0.00645 (0.07578)
	c=3, k=1	-1.92455 (0.0045)	-1.89133 (0.00605)	-3.08568 (0.00359)	-2.40526 (0.00884)	-2.0391 (0.0068)	–	–	-2.14068 (0.00365)	-1.92331 (0.00657)	-3.04186 (0.00337)	-3.71331 (0.00499)	-0.00778 (0.01113)
	c=1.3, k=1.5	-26.8562 (0.00369)	-26.45868 (0.01044)	-28.08197 (0.00905)	-26.73031 (0.01464)	-26.60206 (0.01513)	–	–	-27.17172 (0.0055)	-26.85657 (0.00575)	-27.73505 (0.01784)	-28.56843 (0.00726)	-0.20571 (0.13733)
	c=3, k=1.1/3	-57.02504 (0.0058)	-50.30763 (0.0207)	-4.56338 (1.2012)	32.93835 (2.71317)	17.88579 (2.98801)	–	–	-56.05765 (0.01862)	-61.13231 (0.01857)	-45.63472 (0.83169)	-62.94908 (0.0177)	-29.57171 (1.4578)
Lognormal	μ=0.5, σ=0.5	1.22756 (0.00441)	1.27158 (0.00666)	-0.06907 (0.00299)	0.88718 (0.00643)	0.81905 (0.00821)	1.1919 (0.00525)	–	0.64969 (0.00607)	-0.24806 (0.00223)	-0.95623 (0.00339)	-1.92082 (0.00191)	0.00077 (0.00546)
	μ=0.6, σ=1	-2.27793 (0.03784)	-1.42521 (0.05012)	-6.30108 (0.0465)	-2.90016 (0.0601)	-3.25725 (0.05523)	-3.54474 (0.05485)	–	-4.66318 (0.03573)	-7.94482 (0.0243)	-9.21148 (0.03622)	-11.1553 (0.01958)	-0.00775 (0.02427)
	μ=1.5, σ=0.8	4.52929 (0.0096)	4.92471 (0.00752)	3.39269 (0.01572)	0.03334 (0.00858)	-0.17967 (0.00768)	-0.36988 (0.00866)	–	2.1515 (0.00904)	2.8525 (0.01334)	2.81448 (0.03292)	-3.16991 (0.01148)	-0.00723 (0.01027)

Table A19: RMSE (QSR, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00718 (0.00245)	1.03806 (0.00245)	0.14128 (0.00177)	0.52729 (0.00432)	0.90379 (0.00311)	–	–	0.80484 (0.0015)	1.00711 (0.00287)	0.03592 (0.00319)	0.70469 (0.00256)	0.02837 (0.00194)
	α=1.5	2.37159 (0.02361)	2.08366 (0.03301)	6.21851 (0.16635)	2.30693 (0.0564)	2.18157 (0.1514)	2.21684 (0.32642)	3.60391 (1.35476)	2.89272 (0.02402)	1.82508 (0.01382)	1.20183 (0.50624)	5.74842 (0.01255)	1.10096 (0.65974)
	α=1.1	30.7772 (0.11062)	30.0362 (0.12359)	15.0073 (1.52122)	29.4073 (5.64426)	28.0544 (2.85431)	26.5658 (0.4472)	26.383 (0.42048)	33.643 (0.07243)	27.2884 (0.11046)	18.5773 (0.36509)	37.1828 (0.05421)	58.7171 (63.9732)
Burr	c=1.2, k=2.5	21.6441 (0.00698)	21.5759 (0.00897)	22.8803 (0.00566)	22.0971 (0.00956)	21.786 (0.00784)	–	–	21.9223 (0.00605)	21.644 (0.00935)	23.3823 (0.00267)	23.839 (0.00363)	0.6048 (0.04411)
	c=3, k=1	1.92412 (0.00436)	1.88949 (0.00415)	3.08385 (0.0028)	2.40531 (0.00566)	2.03733 (0.00394)	–	–	2.13952 (0.00248)	1.92355 (0.00527)	3.04166 (0.00276)	3.71118 (0.00355)	0.07642 (0.00543)
	c=1.3, k=1.5	26.8584 (0.00552)	26.4588 (0.0096)	28.0805 (0.00909)	26.7247 (0.0175)	26.5935 (0.01768)	–	–	27.1724 (0.00623)	26.8589 (0.00625)	27.7078 (0.01641)	28.5677 (0.00575)	1.3924 (0.21664)
	c=3, k=1.1/3	57.0252 (0.00657)	50.3088 (0.0235)	10.1072 (0.70607)	47.0467 (3.47989)	40.1679 (6.62394)	–	–	56.0565 (0.01648)	61.1312 (0.0124)	44.2298 (0.63418)	62.9496 (0.01707)	63.6937 (72.1728)
Lognormal	μ=0.5, σ=0.5	1.22885 (0.00237)	1.27744 (0.00416)	0.07439 (0.00173)	0.88941 (0.00381)	0.82235 (0.00464)	1.19334 (0.00255)	–	0.65119 (0.00272)	0.24878 (0.00167)	0.9561 (0.00278)	1.92156 (0.0011)	0.0345 (0.00292)
	μ=0.6, σ=1	2.27843 (0.02932)	1.44391 (0.03918)	6.2855 (0.03914)	2.89589 (0.04936)	3.25774 (0.04559)	3.54287 (0.04351)	–	4.65903 (0.0302)	7.94089 (0.01712)	9.15765 (0.04261)	11.1597 (0.01538)	0.3124 (0.02325)
	μ=1.5, σ=0.8	4.53112 (0.00709)	4.9352 (0.01193)	3.42031 (0.01859)	0.16537 (0.01265)	0.23204 (0.01079)	0.39639 (0.01113)	–	2.15653 (0.00699)	2.86826 (0.00937)	2.95094 (0.1426)	3.16488 (0.00736)	0.13859 (0.00641)

Table 20: MAD (QSR, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02394 (0.00349) (0.00223)	0.02719 (0.0042) (0.00356)	0.01752 (0.00273) (0.00196)	0.03668 (0.00371) (0.00233)	0.02708 (0.00365) (0.00284)	–	–	0.01553 (0.00211) (0.00177)	0.02711 (0.00359) (0.00208)	0.02416 (0.00243) (0.0008)	0.02441 (0.00423) (0.00226)	0.01885 (0.00116) (0.00068)
	α=1.5	0.23766 (0.02702) (0.01565)	0.30723 (0.03576) (0.026)	0.80742 (0.10573) (0.05881)	0.49999 (0.0688) (0.04975)	0.51269 (0.06804) (0.04281)	0.52718 (0.07987) (0.06061)	0.58715 (0.09503) (0.04403)	0.22388 (0.02577) (0.01321)	0.12842 (0.0128) (0.0087)	0.40824 (0.0589) (0.04026)	0.11653 (0.01459) (0.00798)	0.34345 (0.04291) (0.0179)
	α=1.1	0.67245 (0.10996) (0.09892)	0.81238 (0.08375) (0.0574)	7.55904 (0.84745) (0.31656)	3.72966 (0.41699) (0.31117)	3.50685 (0.38761) (0.37592)	3.21979 (0.46241) (0.18463)	2.94418 (0.3168) (0.05494)	0.51559 (0.06465) (0.04081)	0.77429 (0.11331) (0.07624)	2.31758 (0.3415) (0.14126)	0.42017 (0.04129) (0.02341)	3.67829 (0.67957) (0.3907)
Burr	c=1.2, k=2.5	0.05353 (0.00501) (0.0025)	0.064 (0.00507) (0.00411)	0.04113 (0.00272) (0.00185)	0.07511 (0.00774) (0.00449)	0.06254 (0.0061) (0.00467)	–	–	0.04268 (0.00377) (0.0028)	0.05368 (0.00586) (0.00418)	0.02287 (0.00187) (0.00148)	0.03595 (0.00286) (0.00116)	0.42141 (0.04331) (0.01628)
	c=3, k=1	0.02549 (0.00287) (0.00198)	0.03036 (0.00272) (0.00162)	0.01947 (0.00182) (0.00128)	0.03865 (0.00502) (0.00365)	0.02844 (0.00323) (0.0021)	–	–	0.01798 (0.00185) (0.00119)	0.02795 (0.0032) (0.00184)	0.02254 (0.0028) (0.00194)	0.02498 (0.00177) (0.00098)	0.05054 (0.0069) (0.00567)
	c=1.3, k=1.5	0.04522 (0.00657) (0.00324)	0.06816 (0.01176) (0.00688)	0.06176 (0.00934) (0.00623)	0.11689 (0.0132) (0.00919)	0.11256 (0.00957) (0.00833)	–	–	0.04881 (0.00601) (0.00398)	0.04632 (0.00647) (0.00498)	0.09666 (0.00968) (0.00314)	0.04666 (0.00584) (0.00323)	0.74238 (0.10774) (0.04455)
	c=3, k=1.1/3	0.04801 (0.00515) (0.00353)	0.17145 (0.01841) (0.01259)	5.93298 (0.94746) (0.70434)	14.4524 (1.66191) (1.26267)	13.7022 (1.33862) (0.58507)	–	–	0.12624 (0.0175) (0.00599)	0.08713 (0.01184) (0.01103)	3.68968 (0.51185) (0.26525)	0.11987 (0.01661) (0.01107)	5.79942 (1.20585) (0.46304)
Lognormal	μ=0.5, σ=0.5	0.02373 (0.00274) (0.0022)	0.03258 (0.00413) (0.00279)	0.01737 (0.00256) (0.00125)	0.03308 (0.00433) (0.00124)	0.03861 (0.00413) (0.00188)	0.02304 (0.00258) (0.00141)	–	0.02828 (0.00375) (0.00363)	0.01398 (0.00137) (0.0009)	0.0194 (0.00206) (0.00182)	0.01247 (0.0014) (0.00127)	0.02362 (0.00225) (0.00169)
	μ=0.6, σ=1	0.1187 (0.01159) (0.00837)	0.18857 (0.02532) (0.0141)	0.1953 (0.02474) (0.02347)	0.24676 (0.03072) (0.02679)	0.22346 (0.0275) (0.02343)	0.21933 (0.03141) (0.0225)	–	0.12198 (0.01043) (0.00648)	0.09076 (0.00801) (0.00661)	0.15126 (0.01792) (0.01456)	0.07828 (0.00564) (0.00405)	0.20386 (0.02101) (0.01409)
	μ=1.5, σ=0.8	0.08244 (0.01318) (0.01312)	0.12369 (0.01311) (0.00864)	0.15618 (0.01018) (0.00908)	0.10977 (0.00946) (0.00804)	0.1028 (0.009) (0.00619)	0.10678 (0.00817) (0.00474)	–	0.08303 (0.00884) (0.00507)	0.09798 (0.00905) (0.00614)	0.17252 (0.02255) (0.01083)	0.06425 (0.0052) (0.00296)	0.09576 (0.01062) (0.00824)

TABLES OBTAINED FOR THE QSR FOR n=10 000

Table A21: Mean – true QSR (QSR, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00261 (0.00417)	1.03908 (0.00386)	-0.13839 (0.00259)	0.52676 (0.00546)	0.90718 (0.00416)	–	–	0.80397 (0.00228)	1.00233 (0.00573)	0.00083 (0.0032)	-0.70465 (0.0035)	0.00178 (0.00365)
	α=1.5	-2.38855 (0.03775)	-2.07131 (0.04836)	5.97943 (0.14086)	-2.16546 (0.09216)	-1.93201 (0.10892)	-1.63166 (0.19453)	-0.89853 (0.26753)	-2.90525 (0.03568)	-1.82846 (0.01931)	-0.13751 (0.17909)	-5.76424 (0.01882)	-0.08662 (0.11921)
	α=1.1	-31.02607 (0.16086)	-30.25878 (0.18792)	-6.94722 (1.64031)	-25.40742 (1.05043)	-24.43578 (1.90392)	-24.54294 (1.12223)	-24.9226 (1.25802)	-33.77561 (0.13337)	-27.54336 (0.17295)	-18.04131 (1.09519)	-37.27262 (0.10522)	-7.75352 (15.1068)

Burr	c=1.2, k=2.5	-21.65481 (0.01132)	-21.56818 (0.01246)	-22.87844 (0.00828)	-22.08592 (0.01462)	-21.77102 (0.01255)	–	–	-21.92351 (0.00835)	-21.65703 (0.00823)	-23.38063 (0.00743)	-23.84129 (0.00904)	-0.00517 (0.10038)
	c=3, k=1	-1.92625 (0.00475)	-1.88545 (0.00698)	-3.08233 (0.00424)	-2.40095 (0.00815)	-2.03187 (0.00537)	–	–	-2.13882 (0.00354)	-1.92618 (0.00493)	-3.03991 (0.00505)	-3.71059 (0.00495)	-0.00027 (0.01315)
	c=1.3, k=1.5	-26.85472 (0.01034)	-26.45158 (0.01277)	-28.07286 (0.01042)	-26.70787 (0.01714)	-26.57532 (0.0148)	–	–	-27.16774 (0.01009)	-28.56704 (0.01018)	-26.85458 (0.0233)	-27.70517 (0.00836)	0.09255 (0.25959)
	c=3, k=1.1/3	-57.02267 (0.00677)	-50.29916 (0.02418)	-2.52555 (0.97813)	42.26736 (3.94189)	29.90385 (4.87817)	–	–	-56.04985 (0.01642)	-61.12365 (0.00989)	-43.45553 (0.57047)	-62.9402 (0.0144)	25.92978 (130.632)
Lognormal	μ=0.5, σ=0.5	1.22995 (0.00503)	1.2905 (0.00546)	-0.0693 (0.0035)	0.89704 (0.00621)	0.82935 (0.00752)	1.20048 (0.0054)	–	0.65177 (0.00664)	-0.24771 (0.00247)	-0.94998 (0.00329)	-1.92093 (0.00306)	0.00112 (0.00553)
	μ=0.6, σ=1	-2.26687 (0.0228)	-1.39639 (0.03388)	-6.25136 (0.03358)	-2.83992 (0.04243)	-3.21217 (0.03873)	-3.49941 (0.03762)	–	-4.64718 (0.02308)	-7.94075 (0.01633)	-9.15996 (0.06175)	-11.16078 (0.0104)	-0.00244 (0.03835)
	μ=1.5, σ=0.8	4.5272 (0.01172)	4.92333 (0.01652)	3.40251 (0.0218)	0.03051 (0.01432)	-0.18407 (0.01349)	-0.37414 (0.01395)	–	2.14927 (0.01028)	2.86369 (0.01532)	2.88221 (0.06577)	-3.16018 (0.01076)	0.00312 (0.01614)

Table A22: Median – true QSR (QSR, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00489 (0.00632)	1.03964 (0.00726)	-0.13843 (0.00486)	0.52485 (0.00912)	0.90436 (0.00723)	–	–	0.80383 (0.00388)	1.00427 (0.00799)	0.00062 (0.00462)	-0.70373 (0.00639)	0.00072 (0.00438)
	α=1.5	-2.39704 (0.04063)	-2.11585 (0.04848)	5.75467 (0.16062)	-2.34376 (0.08996)	-2.13756 (0.10735)	-2.02275 (0.12709)	-1.6533 (0.176)	-2.92734 (0.03775)	-1.83832 (0.02685)	-0.37427 (0.07437)	-5.7743 (0.02225)	-0.29768 (0.08079)
	α=1.1	-31.05097 (0.2259)	-30.31439 (0.29271)	-12.44898 (2.21984)	-29.05634 (1.01145)	-29.16104 (0.88708)	-29.37438 (0.90005)	-29.354 (2.02788)	-33.81742 (0.19903)	-27.5706 (0.18961)	-19.56879 (0.74485)	-37.29555 (0.14264)	-18.04897 (1.02889)
Burr	c=1.2, k=2.5	-21.65534 (0.01449)	-21.57798 (0.01846)	-22.8841 (0.01137)	-22.10016 (0.02076)	-21.78912 (0.01749)	–	–	-21.92951 (0.01205)	-21.65235 (0.01065)	-23.384 (0.00642)	-23.84388 (0.01003)	-0.02771 (0.1404)
	c=3, k=1	-1.92585 (0.00204)	-1.88763 (0.00802)	-3.08402 (0.00492)	-2.40733 (0.01037)	-2.03795 (0.00727)	–	–	-2.14054 (0.0031)	-1.92585 (0.00457)	-3.04228 (0.00534)	-3.71307 (0.0037)	-0.00193 (0.01397)
	c=1.3, k=1.5	-26.85772 (0.00819)	-26.45652 (0.01429)	-28.07829 (0.01345)	-26.72593 (0.02175)	-26.59585 (0.01776)	–	–	-27.17176 (0.0083)	-28.56953 (0.01086)	-26.86149 (0.01525)	-27.72789 (0.00964)	-0.09715 (0.12455)
	c=3, k=1.1/3	-57.02367 (0.00831)	-50.30273 (0.0297)	-4.37923 (0.8589)	33.93106 (2.55752)	19.07305 (2.41904)	–	–	-56.05192 (0.01822)	-61.11899 (0.01872)	-46.06318 (0.68461)	-62.94305 (0.02208)	-29.13981 (1.28706)
Lognormal	μ=0.5, σ=0.5	1.23009 (0.00456)	1.28502 (0.00619)	-0.0687 (0.00269)	0.89459 (0.00562)	0.82627 (0.00838)	1.19871 (0.00463)	–	0.65185 (0.0053)	-0.24865 (0.00275)	-0.95001 (0.00424)	-1.92084 (0.00356)	0.00269 (0.00481)
	μ=0.6, σ=1	-2.27335 (0.0287)	-1.38576 (0.04228)	-6.25327 (0.04083)	-2.84307 (0.0526)	-3.21081 (0.04684)	-3.49444 (0.04154)	–	-4.65027 (0.02722)	-7.9381 (0.01756)	-9.22341 (0.03835)	-11.1583 (0.00925)	-0.00976 (0.04407)
	μ=1.5, σ=0.8	4.52266 (0.01542)	4.91269 (0.02325)	3.37459 (0.0227)	0.02115 (0.01608)	-0.19096 (0.01477)	-0.38129 (0.01594)	–	2.14193 (0.01626)	2.86514 (0.02312)	2.79767 (0.04672)	-3.165 (0.01261)	0.00162 (0.02562)

Table A23: RMSE (QSR, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00363 (0.00414)	1.04024 (0.00383)	0.14206 (0.00259)	0.53033 (0.00523)	0.90832 (0.00409)	–	–	0.80445 (0.0023)	1.0035 (0.00574)	0.03969 (0.00237)	0.70606 (0.0035)	0.03631 (0.00347)
	α=1.5	2.42604 (0.03651)	2.13433 (0.04404)	6.1486 (0.16234)	2.3603 (0.06173)	2.2205 (0.07957)	2.39735 (0.50389)	2.88228 (0.63993)	2.93141 (0.0343)	1.84271 (0.01885)	1.45818 (1.10275)	5.76772 (0.01857)	1.09748 (0.50416)

	α=1.1	31.0543 (0.16084)	30.2944 (0.18774)	15.0163 (1.43675)	27.0526 (0.84634)	31.225 (14.033)	27.727 (2.4966)	26.0258 (1.06482)	33.7886 (0.13336)	27.5854 (0.1757)	19.1712 (0.6599)	37.28 (0.10531)	73.7836 (147.158)
Burr	c=1.2, k=2.5	21.6551 (0.01132)	21.5685 (0.01245)	22.8786 (0.00828)	22.0863 (0.01459)	21.7713 (0.01253)	–	–	21.9236 (0.00835)	21.6573 (0.00825)	23.3807 (0.00741)	23.8414 (0.00904)	0.72459 (0.05076)
	c=3, k=1	1.92687 (0.00475)	1.88621 (0.00695)	3.08252 (0.00424)	2.40186 (0.00813)	2.03247 (0.00533)	–	–	2.13905 (0.00354)	1.92687 (0.00493)	3.0402 (0.00502)	3.71087 (0.00495)	0.09395 (0.01122)
	c=1.3, k=1.5	26.8549 (0.01035)	26.452 (0.01278)	28.0732 (0.01042)	26.7089 (0.01715)	26.5763 (0.0148)	–	–	27.1679 (0.0101)	26.8548 (0.01019)	27.7061 (0.0228)	28.5672 (0.00837)	2.04298 (1.56114)
	c=3, k=1.1/3	57.0228 (0.00677)	50.3002 (0.02413)	12.3276 (1.6898)	55.1159 (8.77592)	51.8689 (14.3614)	–	–	56.0503 (0.01638)	61.1239 (0.00989)	44.5465 (0.43784)	62.9406 (0.01437)	480.724 (1298.18)
Lognormal	μ=0.5, σ=0.5	1.23072 (0.00506)	1.29197 (0.00557)	0.07656 (0.00317)	0.89885 (0.00633)	0.83185 (0.00772)	1.20102 (0.00546)	–	0.65401 (0.00671)	0.24922 (0.0024)	0.95071 (0.00326)	1.92109 (0.00305)	0.04105 (0.00191)
	μ=0.6, σ=1	2.27716 (0.02288)	1.44085 (0.03445)	6.2622 (0.03358)	2.87769 (0.04255)	3.23966 (0.03884)	3.52371 (0.03782)	–	4.65283 (0.02309)	7.94238 (0.01622)	9.17141 (0.05175)	11.1616 (0.01036)	0.38389 (0.02228)
	μ=1.5, σ=0.8	4.52973 (0.01178)	4.92823 (0.01671)	3.4136 (0.02275)	0.19252 (0.01731)	0.25714 (0.01272)	0.41682 (0.01292)	–	2.15461 (0.01038)	2.86952 (0.01559)	2.93913 (0.11338)	3.16274 (0.01084)	0.1685 (0.008)

Table A24: MAD (QSR, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.03054 (0.00437) (0.00318)	0.03312 (0.00396) (0.0021)	0.02159 (0.00203) (0.00133)	0.04334 (0.00449) (0.00253)	0.03128 (0.00345) (0.002)	–	–	0.01864 (0.00175) (0.00114)	0.03337 (0.00294) (0.00148)	0.02727 (0.00154) (0.00081)	0.0302 (0.00373) (0.00134)	0.02292 (0.0024) (0.00109)
	α=1.5	0.29449 (0.02551) (0.01748)	0.34548 (0.03272) (0.02291)	0.84011 (0.09257) (0.06087)	0.50639 (0.05287) (0.04161)	0.51943 (0.04635) (0.0369)	0.54911 (0.05513) (0.0383)	0.67963 (0.13811) (0.09319)	0.26543 (0.01938) (0.01503)	0.15732 (0.01546) (0.01319)	0.40544 (0.0503) (0.03684)	0.13471 (0.01503) (0.0138)	0.39248 (0.0525) (0.0409)
	α=1.1	0.99675 (0.09958) (0.0835)	1.06842 (0.09928) (0.0893)	5.61656 (2.14003) (1.29024)	2.84514 (1.17322) (0.61177)	2.63834 (1.20042) (0.56786)	2.32252 (1.23408) (0.40983)	1.63289 (1.66242) (0.22362)	0.67857 (0.07266) (0.06592)	1.14667 (0.10471) (0.0466)	2.46919 (0.37909) (0.23487)	0.52816 (0.07018) (0.05674)	3.57395 (0.64746) (0.51773)
Burr	c=1.2, k=2.5	0.06848 (0.00481) (0.00266)	0.07273 (0.00912) (0.00575)	0.04718 (0.00296) (0.00221)	0.08116 (0.00936) (0.00353)	0.06894 (0.00802) (0.00486)	–	–	0.05078 (0.00239) (0.00208)	0.06931 (0.00457) (0.00369)	0.02583 (0.00251) (0.00176)	0.04237 (0.00551) (0.00194)	0.46673 (0.05344) (0.02127)
	c=3, k=1	0.03435 (0.00362) (0.00225)	0.03893 (0.00374) (0.00302)	0.02494 (0.00196) (0.00141)	0.04675 (0.00409) (0.00274)	0.03565 (0.00287) (0.00203)	–	–	0.02197 (0.00163) (0.00083)	0.0332 (0.00296) (0.00102)	0.02823 (0.00174) (0.00163)	0.03158 (0.00365) (0.00109)	0.05876 (0.00572) (0.00416)
	c=1.3, k=1.5	0.05673 (0.00579) (0.00403)	0.08667 (0.01124) (0.00628)	0.07682 (0.01047) (0.00871)	0.1405 (0.01672) (0.00529)	0.13304 (0.01595) (0.01186)	–	–	0.05968 (0.00692) (0.00376)	0.05811 (0.00664) (0.00291)	0.11986 (0.01114) (0.00409)	0.06079 (0.00612) (0.00315)	0.94094 (0.08997) (0.06206)
	c=3, k=1.1/3	0.05725 (0.00627) (0.00505)	0.20447 (0.02238) (0.01804)	7.24885 (0.58077) (0.26068)	16.8263 (1.60638) (1.39771)	16.566 (2.63082) (1.94322)	–	–	0.14139 (0.01984) (0.0176)	0.10479 (0.01239) (0.008)	3.83124 (0.49299) (0.33131)	0.14541 (0.01394) (0.00862)	6.51487 (0.83039) (0.63029)
Lognormal	μ=0.5, σ=0.5	0.02849 (0.00345) (0.00261)	0.0391 (0.00313) (0.00172)	0.02088 (0.00271) (0.00068)	0.03718 (0.00292) (0.00177)	0.0423 (0.00361) (0.0026)	0.02558 (0.00251) (0.00192)	–	0.03645 (0.00373) (0.0034)	0.01898 (0.00126) (0.00074)	0.02417 (0.00239) (0.00052)	0.01625 (0.00157) (0.00079)	0.02719 (0.00281) (0.00089)
	μ=0.6, σ=1	0.14787 (0.02431) (0.01847)	0.23032 (0.05037) (0.0267)	0.23659 (0.04553) (0.02788)	0.29536 (0.05617) (0.03272)	0.27178 (0.05011) (0.02796)	0.26196 (0.04922) (0.03223)	–	0.15179 (0.02814) (0.01243)	0.10535 (0.01549) (0.00588)	0.16995 (0.02074) (0.00931)	0.09203 (0.00944) (0.00594)	0.2583 (0.02907) (0.02339)

	µ=1.5, σ=0.8	0.0999 (0.01249) (0.01064)	0.13993 (0.02164) (0.01609)	0.16995 (0.01846) (0.01176)	0.12022 (0.01053) (0.00741)	0.11315 (0.01004) (0.00704)	0.11523 (0.01014) (0.00599)	–	0.09906 (0.01189) (0.00591)	0.11659 (0.01367) (0.00853)	0.19706 (0.02459) (0.01838)	0.07822 (0.0091) (0.00605)	0.1139 (0.00952) (0.00377)

TABLES OBTAINED FOR THE QSR FOR n=5 000

Table A25: Mean – true QSR (QSR, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.99327 (0.00459)	1.04923 (0.00456)	-0.13501 (0.0031)	0.53635 (0.00636)	0.92762 (0.00536)	–	–	0.80624 (0.00236)	0.99409 (0.0055)	0.00575 (0.00608)	-0.70734 (0.00579)	0.00051 (0.00446)
	α=1.5	-2.45548 (0.06851)	-2.05621 (0.08711)	6.27898 (0.25909)	-1.9199 (0.19319)	-1.51621 (0.37734)	-0.73085 (0.72643)	-0.29724 (0.45755)	-2.9216 (0.06469)	-1.86088 (0.02411)	-0.06039 (0.17389)	-5.77677 (0.03342)	-0.0708 (0.06715)
	α=1.1	-31.46835 (0.12349)	-30.56434 (0.14134)	-2.76985 (1.18537)	-22.61364 (0.61364)	-21.7178 (0.95908)	-22.10939 (1.25947)	-18.91912 (0.75933)	-33.95335 (0.12802)	-28.01604 (0.19838)	-17.02524 (1.08143)	-37.36724 (0.11193)	-10.58184 (3.9289)
Burr	c=1.2, k=2.5	-21.67117 (0.01709)	-21.53983 (0.01276)	-22.86663 (0.00954)	-22.04752 (0.01419)	-21.7126 (0.01731)	–	–	-21.91886 (0.01132)	-21.67248 (0.01727)	-23.37499 (0.00616)	-23.84222 (0.00841)	-0.002 (0.11438)
	c=3, k=1	-1.93819 (0.00876)	-1.8737 (0.00795)	-3.07838 (0.00549)	-2.38697 (0.00873)	-2.00587 (0.00904)	–	–	-2.13764 (0.0055)	-1.93851 (0.00801)	-3.03249 (0.0054)	-3.71132 (0.00555)	0.00276 (0.01384)
	c=1.3, k=1.5	-26.85763 (0.01374)	-26.45896 (0.02486)	-28.07794 (0.02347)	-26.71344 (0.04439)	-26.57619 (0.04323)	–	–	-27.17152 (0.01621)	-26.8577 (0.01381)	-27.70904 (0.04142)	-28.57218 (0.01524)	0.01542 (0.31285)
	c=3, k=1.1/3	-57.02298 (0.01324)	-50.30025 (0.04728)	-1.80372 (1.69391)	48.54558 (6.13047)	46.17516 (9.27309)	–	–	-56.05644 (0.0293)	-61.14585 (0.02071)	-44.41772 (0.95678)	-62.96502 (0.02565)	-13.40327 (8.91628)
Lognormal	µ=0.5, σ=0.5	1.22704 (0.00538)	1.3291 (0.00827)	-0.06972 (0.00428)	0.91478 (0.00738)	0.84519 (0.0082)	1.22164 (0.00964)	–	0.6494 (0.00714)	-0.24968 (0.00348)	-0.93878 (0.00329)	-1.92319 (0.0037)	-0.00329 (0.00544)
	µ=0.6, σ=1	-2.28179 (0.05731)	-1.41354 (0.07595)	-6.25424 (0.0681)	-2.84719 (0.0871)	-3.22102 (0.08129)	-3.50614 (0.07596)	–	-4.66098 (0.05239)	-7.94921 (0.01905)	-9.17452 (0.04616)	-11.1659 (0.01722)	0.00269 (0.03719)
	µ=1.5, σ=0.8	4.52614 (0.02399)	4.92372 (0.03225)	3.41719 (0.04024)	0.03018 (0.02724)	-0.18627 (0.0257)	-0.37779 (0.02607)	–	2.14661 (0.02249)	2.85331 (0.02134)	2.86849 (0.06162)	-3.16975 (0.01531)	-0.01363 (0.02566)

Table A26: Median – true QSR (QSR, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.99089 (0.00701)	1.04444 (0.01005)	-0.13838 (0.00603)	0.53115 (0.0117)	0.9229 (0.00857)	–	–	0.80389 (0.00349)	0.99383 (0.00598)	0.0032 (0.00566)	-0.70987 (0.00751)	-0.00365 (0.0051)
	α=1.5	2.53031 (0.08777)	2.17474 (0.1126)	6.60123 (0.2805)	2.47367 (0.16314)	3.39241 (0.20043)	6.51258 (0.17207)	2.86372 (0.06595)	2.97383 (0.08444)	1.89002 (0.02852)	1.39393 (0.12788)	5.78383 (0.03399)	1.26035 (0.08519)
	α=1.1	-31.69905 (0.23625)	-30.76463 (0.32632)	-2.67409 (2.21667)	-24.29611 (1.3445)	-24.38193 (1.36414)	-24.06466 (0.737)	-21.70386 (0.40391)	-34.0376 (0.218)	-28.21703 (0.23814)	-19.81653 (0.57265)	-37.38853 (0.18986)	-19.17056 (0.68493)
Burr	c=1.2, k=2.5	-21.67852 (0.0292)	-21.54899 (0.02118)	-22.87333 (0.01385)	-22.06105 (0.0211)	-21.72631 (0.02336)	–	–	-21.92937 (0.01593)	-21.67722 (0.02811)	-23.37688 (0.00797)	-23.83934 (0.01241)	-0.06452 (0.14048)
	c=3, k=1	-1.94419 (0.01236)	-1.87974 (0.00933)	-3.08269 (0.00654)	-2.39182 (0.01167)	-2.01662 (0.00628)	–	–	-2.14256 (0.00778)	-1.94556 (0.01208)	-3.03494 (0.00591)	-3.7144 (0.00725)	-0.00455 (0.01927)
	c=1.3, k=1.5	-26.85555 (0.01372)	-26.46225 (0.02944)	-28.08918 (0.02707)	-26.74422 (0.05872)	-26.61949 (0.05484)	–	–	-27.17197 (0.01748)	-26.85795 (0.01546)	-27.76769 (0.0273)	-28.57258 (0.01975)	-0.3639 (0.25406)

	c=3, k=1.1/3	-57.02145 (0.0196)	-50.29479 (0.07)	-4.78792 (1.65717)	33.28013 (3.66476)	17.38831 (3.40893)	–	–	-56.05621 (0.03997)	-61.14562 (0.01947)	-47.88038 (0.9474)	-62.95959 (0.03681)	-32.18113 (1.2619)
Lognormal	μ=0.5, σ=0.5	1.22756 (0.00763)	1.31647 (0.01152)	-0.07095 (0.00672)	0.91041 (0.01114)	0.8417 (0.00955)	1.21623 (0.01085)	–	0.64836 (0.00963)	-0.24977 (0.00493)	-0.94053 (0.00636)	-1.92515 (0.00326)	-0.0038 (0.00801)
	μ=0.6, σ=1	-2.29003 (0.03218)	-1.43647 (0.05746)	-6.31273 (0.06611)	-2.90427 (0.08459)	-3.27604 (0.07667)	-3.55755 (0.07335)	–	-4.67254 (0.03357)	-7.94973 (0.0193)	-9.27328 (0.05607)	-11.1674 (0.02046)	-0.03717 (0.05276)
	μ=1.5, σ=0.8	4.53025 (0.03944)	4.91381 (0.05012)	3.38846 (0.05682)	0.02988 (0.04094)	-0.18414 (0.03774)	-0.37664 (0.03988)	–	2.14848 (0.03131)	2.85976 (0.03771)	2.75693 (0.03204)	-3.17004 (0.0234)	-0.01147 (0.02836)

Table A27: RMSE (QSR, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.99561 (0.00457)	1.05155 (0.00464)	0.14281 (0.00284)	0.54259 (0.00662)	0.92953 (0.00564)	–	–	0.80728 (0.00237)	0.9969 (0.00537)	0.05733 (0.0034)	0.71021 (0.00576)	0.04914 (0.00303)
	α=1.5	2.5295 (0.06465)	2.17332 (0.07931)	6.59501 (0.29435)	2.46533 (0.20752)	3.13777 (1.3536)	5.55083 (3.62211)	2.6737 (1.09736)	2.9732 (0.06183)	1.88987 (0.02221)	1.32487 (0.45438)	5.78374 (0.0329)	1.24792 (0.18136)
	α=1.1	31.5211 (0.12282)	30.6303 (0.1417)	16.0312 (1.22067)	25.6643 (1.02088)	27.3859 (3.11751)	25.9672 (1.35539)	20.1349 (0.52179)	33.9775 (0.1268)	28.0905 (0.19557)	20.1667 (0.80485)	37.3814 (0.11126)	45.0013 (20.9787)
Burr	c=1.2, k=2.5	21.6716 (0.01708)	21.5403 (0.01273)	22.8669 (0.00953)	22.0482 (0.01415)	21.7135 (0.01727)	–	–	21.9191 (0.01132)	21.673 (0.01724)	23.3751 (0.00615)	23.8424 (0.00841)	1.06365 (0.09712)
	c=3, k=1	1.9397 (0.00885)	1.87544 (0.00799)	3.07885 (0.00551)	2.3888 (0.00872)	2.00731 (0.00893)	–	–	2.13819 (0.00552)	1.94029 (0.00811)	3.03311 (0.00541)	3.71194 (0.00554)	0.13506 (0.01441)
	c=1.3, k=1.5	26.8579 (0.01372)	26.4596 (0.02481)	28.0785 (0.02343)	26.7154 (0.04422)	26.5781 (0.04307)	–	–	27.1719 (0.01618)	26.858 (0.01379)	27.7119 (0.03842)	28.5725 (0.01522)	2.69246 (1.66729)
	c=3, k=1.1/3	57.0231 (0.01324)	50.3022 (0.04728)	17.2087 (1.78303)	74.479 (13.3838)	108.527 (30.5245)	–	–	56.0573 (0.02928)	61.1463 (0.0207)	46.0177 (0.79833)	62.9658 (0.02562)	98.5723 (73.5842)
Lognormal	μ=0.5, σ=0.5	1.22865 (0.0055)	1.33287 (0.00838)	0.08451 (0.00386)	0.91831 (0.00751)	0.85011 (0.00834)	1.22292 (0.0099)	–	0.65385 (0.00743)	0.25236 (0.00349)	0.94003 (0.00328)	1.92348 (0.00369)	0.06036 (0.00529)
	μ=0.6, σ=1	2.32548 (0.08834)	1.546 (0.12065)	6.28307 (0.07182)	2.94737 (0.10051)	3.29601 (0.09398)	3.57033 (0.08382)	–	4.68015 (0.06137)	7.95214 (0.01899)	9.19249 (0.04294)	11.1675 (0.0172)	0.53473 (0.02783)
	μ=1.5, σ=0.8	4.53079 (0.02417)	4.9329 (0.03306)	3.43786 (0.0428)	0.26498 (0.02467)	0.31175 (0.01567)	0.4573 (0.0187)	–	2.15625 (0.02299)	2.86395 (0.02155)	2.93663 (0.09853)	3.17404 (0.01549)	0.22696 (0.01735)

Table A28: MAD (QSR, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.04753 (0.004) (0.00309)	0.04662 (0.00382) (0.00306)	0.03049 (0.00319) (0.00225)	0.05294 (0.00514) (0.00192)	0.03978 (0.00574) (0.00228)	–	–	0.02594 (0.00239) (0.00178)	0.05293 (0.00549) (0.00496)	0.03843 (0.00369) (0.00195)	0.04427 (0.00446) (0.00259)	0.03375 (0.00171) (0.00102)
	α=1.5	0.41793 (0.05773) (0.04095)	0.44456 (0.07418) (0.0525)	1.07787 (0.18703) (0.14096)	0.63479 (0.11384) (0.09673)	0.65491 (0.09489) (0.05739)	0.6262 (0.08789) (0.06423)	0.39821 (0.09194) (0.08313)	0.3589 (0.05657) (0.05538)	0.21809 (0.03163) (0.0226)	0.54435 (0.06743) (0.02984)	0.18131 (0.03095) (0.01906)	0.48381 (0.07833) (0.05863)
	α=1.1	1.34985 (0.11729) (0.07191)	1.60657 (0.09753) (0.05308)	11.6493 (0.78549) (0.68163)	6.51019 (0.54267) (0.38549)	6.45443 (0.37303) (0.28412)	6.33192 (0.50801) (0.28976)	2.90736 (0.45684) (0.20007)	1.00414 (0.0845) (0.06212)	1.53721 (0.18442) (0.14055)	3.19901 (0.5085) (0.19947)	0.75658 (0.05949) (0.04009)	4.07039 (0.43476) (0.24613)

Burr	c=1.2, k=2.5	0.09255 (0.00986) (0.00292)	0.09885 (0.01009) (0.00856)	0.06647 (0.0069) (0.00502)	0.10787 (0.01089) (0.00862)	0.09042 (0.01335) (0.00756)	–	–	0.06694 (0.0063) (0.00377)	0.09967 (0.00898) (0.00697)	0.03474 (0.00514) (0.00292)	0.05572 (0.00689) (0.00576)	0.71914 (0.09962) (0.0651)
	c=3, k=1	0.05289 (0.00483) (0.00293)	0.05443 (0.00922) (0.00607)	0.03542 (0.00607) (0.00429)	0.06001 (0.00866) (0.00608)	0.04913 (0.00769) (0.00373)	–	–	0.03208 (0.00371) (0.0023)	0.058 (0.0067) (0.00411)	0.03928 (0.00567) (0.00449)	0.04527 (0.00592) (0.00438)	0.08665 (0.01054) (0.00835)
	c=1.3, k=1.5	0.08086 (0.00818) (0.00421)	0.13095 (0.01647) (0.01287)	0.12033 (0.01545) (0.01198)	0.21507 (0.03296) (0.01817)	0.20491 (0.03043) (0.01487)	–	–	0.08789 (0.00964) (0.00476)	0.08641 (0.01133) (0.00971)	0.14916 (0.01649) (0.01285)	0.08452 (0.00733) (0.00603)	1.259 (0.19355) (0.14419)
	c=3, k=1.1/3	0.08384 (0.01226) (0.00743)	0.29943 (0.0438) (0.02655)	10.3269 (1.32151) (0.81487)	23.6649 (2.70463) (2.02245)	21.6484 (3.07369) (2.10492)	–	–	0.21036 (0.03011) (0.02183)	0.15415 (0.0191) (0.00547)	3.95921 (0.65834) (0.33994)	0.20893 (0.01832) (0.01295)	6.88262 (0.844) (0.70983)
Lognormal	μ=0.5, σ=0.5	0.04097 (0.0053) (0.00521)	0.058 (0.00891) (0.00604)	0.03115 (0.00467) (0.00398)	0.05278 (0.0097) (0.00581)	0.05804 (0.00894) (0.00709)	0.03873 (0.00913) (0.00561)	–	0.05138 (0.00697) (0.00562)	0.02536 (0.00364) (0.00205)	0.03045 (0.00244) (0.00116)	0.02387 (0.00344) (0.00191)	0.03965 (0.00483) (0.00371)
	μ=0.6, σ=1	0.223 (0.02355) (0.01557)	0.36633 (0.04004) (0.01799)	0.37251 (0.04217) (0.02563)	0.46943 (0.05206) (0.03942)	0.42841 (0.04703) (0.02642)	0.42109 (0.05104) (0.02231)	–	0.23669 (0.01617) (0.01185)	0.1477 (0.02413) (0.00755)	0.22231 (0.01322) (0.00734)	0.12723 (0.01824) (0.01535)	0.36047 (0.02175) (0.00737)
	μ=1.5, σ=0.8	0.13311 (0.01642) (0.01398)	0.20704 (0.02855) (0.02411)	0.25314 (0.0302) (0.01358)	0.17877 (0.02244) (0.01724)	0.16819 (0.02207) (0.01698)	0.17574 (0.02156) (0.0124)	–	0.13437 (0.01548) (0.00959)	0.15184 (0.01722) (0.00846)	0.248 (0.02727) (0.0194)	0.1114 (0.01533) (0.0103)	0.14969 (0.01737) (0.00658)

TABLES OBTAINED FOR THE QSR FOR n=1 000

Table A29: Mean – true QSR (QSR, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.99355 (0.01195)	1.11902 (0.01552)	-0.09544 (0.01039)	0.61172 (0.01829)	1.21169 (0.07433)	–	–	0.83032 (0.00901)	0.99569 (0.01812)	0.04105 (0.01189)	-0.6797 (0.01015)	-0.00104 (0.00936)
	α=1.5	-3.14265 (0.08615)	-2.45187 (0.11185)	6.49574 (0.37905)	-1.15211 (0.48149)	0.55546 (0.58949)	2.56858 (0.89664)	–	-3.35777 (0.07647)	-2.15896 (0.06117)	-0.04383 (0.24008)	-6.05745 (0.03629)	-0.0708 (0.06715)
	α=1.1	-34.585 (0.12365)	-33.61645 (0.14048)	-4.92998 (1.27724)	-9.97223 (27.0609)	-23.97812 (1.83477)	-5.32406 (4.97136)	–	-35.85608 (0.10525)	-31.29472 (0.13867)	-20.59926 (1.64014)	-38.86644 (0.08082)	-14.38018 (1.79271)
Burr	c=1.2, k=2.5	-21.65504 (0.0386)	-21.38272 (0.05617)	-22.77381 (0.038)	-21.81779 (0.07335)	-20.81695 (0.34716)	–	–	-21.85233 (0.03646)	-21.65779 (0.0441)	-23.33738 (0.01509)	-23.80932 (0.02123)	-0.04505 (0.26197)
	c=3, k=1	-1.93381 (0.01345)	-1.79359 (0.01843)	-3.03227 (0.01251)	-2.29983 (0.02215)	-1.68387 (0.05491)	–	–	-2.10722 (0.01053)	-1.93508 (0.01412)	-2.99905 (0.017)	-3.68414 (0.01344)	-0.01059 (0.03263)
	c=1.3, k=1.5	-26.8579 (0.02085)	-26.44683 (0.0301)	-28.04509 (0.02841)	-26.60155 (0.06115)	-26.36193 (0.10683)	–	–	-27.16377 (0.02123)	-26.85614 (0.02236)	-27.69961 (0.036)	-28.57207 (0.02363)	0.14898 (0.40665)
	c=3, k=1.1/3	-57.54063 (0.08025)	-51.32595 (0.1505)	-3.95409 (4.72884)	88.72651 (41.6201)	73.31515 (29.9192)	–	–	-56.74325 (0.10156)	-61.22882 (0.03646)	-46.73392 (1.93394)	-63.11659 (0.0647)	-26.44782 (9.33684)
Lognormal	μ=0.5, σ=0.5	1.20402 (0.01381)	1.61958 (0.0274)	-0.00022 (0.0142)	1.02324 (0.0201)	1.00215 (0.02427)	1.46811 (0.09657)	–	0.64262 (0.01709)	-0.25751 (0.00712)	-0.88558 (0.01173)	-1.92899 (0.00652)	-0.00382 (0.00582)
	μ=0.6, σ=1	-3.13657 (0.17948)	-2.02897 (0.19439)	-6.32211 (0.15744)	-3.06374 (0.195)	-3.56586 (0.17602)	-3.75559 (0.15588)	–	-5.21952 (0.14189)	-7.99646 (0.055)	-8.94905 (0.08362)	-11.15476 (0.04366)	0.04266 (0.09857)
	μ=1.5, σ=0.8	4.29946 (0.07916)	5.18678 (0.06691)	5.44705 (0.49502)	0.49694 (0.08892)	0.101 (0.0639)	-0.20525 (0.07111)	–	2.14292 (0.04964)	2.70482 (0.06709)	3.99009 (0.60068)	-3.16264 (0.027)	-0.01862 (0.02505)

Table A30: Median – true QSR (QSR, n=1000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.98249 (0.0185)	1.09895 (0.01952)	-0.11061 (0.01295)	0.5889 (0.0256)	1.12657 (0.01473)	–	–	0.8209 (0.01383)	0.98654 (0.02375)	0.03488 (0.01364)	-0.6908 (0.01251)	-0.00972 (0.01475)
	α=1.5	-3.3517 (0.07828)	-2.68608 (0.08232)	5.47515 (0.27422)	-2.58949 (0.11681)	-1.46687 (0.72087)	2.18076 (0.7642)	–	-3.565 (0.06872)	-2.22091 (0.02717)	-0.66027 (0.17733)	-6.14763 (0.03203)	-0.37117 (0.08519)
	α=1.1	-34.65871 (0.15215)	-33.85971 (0.16652)	-18.30419 (2.11812)	-32.76107 (0.52596)	-32.91944 (0.74591)	-9.432 (6.7815)	–	-36.04057 (0.13713)	-31.61809 (0.14251)	-24.42616 (0.33671)	-39.06163 (0.09906)	-22.29943 (1.15015)
Burr	c=1.2, k=2.5	-21.69426 (0.04719)	-21.43673 (0.06622)	-22.81833 (0.04532)	-21.93138 (0.09963)	-21.11767 (0.10089)	–	–	-21.8922 (0.04292)	-21.69406 (0.05419)	-23.35722 (0.01511)	-23.83899 (0.02652)	-0.13878 (0.28842)
	c=3, k=1	-1.95301 (0.0159)	-1.81546 (0.01922)	-3.04997 (0.01286)	-2.3335 (0.02542)	-1.7815 (0.03325)	–	–	-2.12132 (0.0126)	-1.94564 (0.01914)	-3.01779 (0.01471)	-3.70776 (0.0147)	-0.02971 (0.02696)
	c=1.3, k=1.5	-26.86718 (0.03483)	-26.46666 (0.03826)	-28.09968 (0.03591)	-26.75056 (0.05472)	-26.6073 (0.07046)	–	–	-27.1787 (0.03179)	-26.86608 (0.03801)	-27.78662 (0.02462)	-28.58341 (0.02703)	-0.53894 (0.35976)
	c=3, k=1.1/3	-57.10677 (0.03001)	-50.5995 (0.10718)	-12.12109 (3.52794)	16.63446 (7.79153)	3.15721 (5.50429)	–	–	-56.24521 (0.08462)	-61.22555 (0.02959)	-51.9218 (0.55891)	-63.11918 (0.07983)	-38.40074 (1.27408)
Lognormal	μ=0.5, σ=0.5	1.18498 (0.01844)	1.5598 (0.03553)	-0.02129 (0.01566)	1.0047 (0.0267)	0.97781 (0.02577)	1.42855 (0.05797)	–	0.63665 (0.02604)	-0.25804 (0.00847)	-0.88924 (0.01583)	-1.93418 (0.01206)	-0.00188 (0.01456)
	μ=0.6, σ=1	-2.43735 (0.10458)	-1.4661 (0.11244)	-6.21667 (0.15098)	-2.8243 (0.18672)	-3.24483 (0.18675)	-3.55695 (0.14465)	–	-4.77297 (0.11804)	-7.98264 (0.06881)	-9.19835 (0.06756)	-11.16399 (0.04647)	0.03294 (0.09209)
	μ=1.5, σ=0.8	4.42485 (0.09725)	5.16534 (0.07196)	3.8947 (0.14619)	0.31336 (0.07066)	0.03092 (0.05792)	-0.2616 (0.08818)	–	2.12914 (0.05248)	2.70235 (0.08644)	2.95918 (0.09011)	-3.18276 (0.03805)	-0.04182 (0.05354)

Table A31: RMSE (QSR, n=1000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	1.00137 (0.01244)	1.13059 (0.01625)	0.14506 (0.00699)	0.64159 (0.02051)	1.26652 (0.14034)	–	–	0.83551 (0.00927)	1.00528 (0.01904)	0.13336 (0.01082)	0.69093 (0.01013)	0.1124 (0.01)
	α=1.5	3.26466 (0.06797)	2.67651 (0.0773)	7.39533 (0.47934)	4.57527 (1.52064)	5.96053 (1.45449)	3.53533 (1.30493)	–	3.46343 (0.06258)	2.21699 (0.05153)	2.95764 (1.18244)	6.07198 (0.03442)	1.24792 (0.18136)
	α=1.1	34.6223 (0.11782)	33.6653 (0.13332)	36.2127 (9.40269)	134.513 (220.493)	30.9448 (2.78644)	12.0243 (2.71038)	–	35.8791 (0.10252)	31.3641 (0.12792)	26.1257 (3.49996)	38.8821 (0.07975)	36.0234 (9.11678)
Burr	c=1.2, k=2.5	21.6566 (0.03842)	21.3855 (0.05583)	22.7751 (0.03783)	21.8233 (0.07249)	20.8831 (0.27032)	–	–	21.8535 (0.03631)	21.6595 (0.04398)	23.3378 (0.01504)	23.8098 (0.02117)	2.22155 (0.16979)
	c=3, k=1	1.93878 (0.01332)	1.80232 (0.0179)	3.03465 (0.01235)	2.30966 (0.02136)	1.7125 (0.03333)	–	–	2.10984 (0.01041)	1.94084 (0.01417)	3.0021 (0.01637)	3.68661 (0.01317)	0.2963 (0.01925)
	c=1.3, k=1.5	26.8594 (0.02073)	26.4502 (0.02987)	28.0479 (0.02817)	26.6138 (0.05943)	26.3899 (0.08414)	–	–	27.1654 (0.02109)	26.8577 (0.02227)	27.7058 (0.03564)	28.5736 (0.02359)	4.78956 (1.14276)
	c=3, k=1.1/3	57.5576 (0.08258)	51.3906 (0.15756)	42.6559 (26.7327)	397.048 (350.043)	275.118 (117.912)	–	–	56.7688 (0.10408)	61.2306 (0.03636)	50.6096 (1.74418)	63.1194 (0.0646)	61.8498 (47.5083)
Lognormal	μ=0.5, σ=0.5	1.21264 (0.01401)	1.64677 (0.02999)	0.1509 (0.01188)	1.0432 (0.0201)	1.01775 (0.02447)	1.49064 (0.12335)	–	0.66554 (0.01624)	0.27062 (0.00607)	0.89333 (0.01116)	1.93052 (0.00651)	0.13156 (0.01579)
	μ=0.6, σ=1	3.79378 (0.29169)	3.1501 (0.36695)	6.66906 (0.19585)	4.01994 (0.30627)	4.28634 (0.2831)	4.3052 (0.2153)	–	5.5005 (0.1959)	8.01443 (0.05465)	9.03958 (0.06044)	11.1627 (0.04324)	1.12775 (0.08521)
	μ=1.5, σ=0.8	4.35473 (0.0773)	5.22848 (0.06784)	6.85806 (1.2037)	0.99589 (0.15649)	0.62521 (0.08437)	0.54881 (0.03334)	–	2.19463 (0.05069)	2.7828 (0.07254)	5.90799 (2.68845)	3.18688 (0.026)	0.52855 (0.03794)

Table A32: MAD (QSR, n=1000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.08086 (0.00764) (0.00417)	0.10431 (0.01257) (0.00893)	0.06885 (0.00681) (0.0033)	0.13162 (0.00841) (0.0061)	0.05966 (0.01131) (0.00727)	–	–	0.05928 (0.00617) (0.00451)	0.09489 (0.00806) (0.00578)	0.08124 (0.00598) (0.0044)	0.08707 (0.00813) (0.00577)	0.06928 (0.00975) (0.00614)
	α=1.5	0.53208 (0.05764) (0.05262)	0.65259 (0.08848) (0.04747)	2.03432 (0.3723) (0.26683)	1.22906 (0.21105) (0.1636)	1.48667 (0.48257) (0.37257)	1.46474 (0.53206) (0.32459)	–	0.50756 (0.07095) (0.0648)	0.31546 (0.03953) (0.01722)	0.91599 (0.07674) (0.04625)	0.26241 (0.04201) (0.01728)	0.48381 (0.07833) (0.05863)
	α=1.1	1.03937 (0.11247) (0.06713)	1.01952 (0.06082) (0.03914)	6.84568 (1.62701) (0.9932)	2.93416 (0.51596) (0.3192)	2.06589 (0.72722) (0.42958)	5.56927 (3.07879) (2.87321)	–	0.72152 (0.04534) (0.03403)	1.26379 (0.16804) (0.09992)	3.21364 (0.37553) (0.25377)	0.61529 (0.05561) (0.03938)	4.62625 (1.10441) (0.45288)
Burr	c=1.2, k=2.5	0.15054 (0.02001) (0.01348)	0.22559 (0.02676) (0.02058)	0.14828 (0.01779) (0.01314)	0.28935 (0.03749) (0.02807)	0.15864 (0.04303) (0.03034)	–	–	0.13726 (0.01884) (0.01628)	0.16842 (0.01687) (0.01225)	0.07564 (0.00911) (0.00782)	0.107 (0.01628) (0.00946)	1.49507 (0.08563) (0.04143)
	c=3, k=1	0.0878 (0.00845) (0.00608)	0.11834 (0.00889) (0.004)	0.0769 (0.00789) (0.00634)	0.14433 (0.0076) (0.00448)	0.08157 (0.02229) (0.0094)	–	–	0.0665 (0.00889) (0.00521)	0.09798 (0.00766) (0.00399)	0.08142 (0.00845) (0.00406)	0.08953 (0.00721) (0.00478)	0.20181 (0.01902) (0.00933)
	c=1.3, k=1.5	0.17772 (0.02129) (0.01589)	0.28484 (0.03205) (0.01894)	0.25102 (0.03055) (0.00991)	0.45507 (0.05462) (0.01615)	0.43855 (0.05173) (0.0239)	–	–	0.19989 (0.02012) (0.01477)	0.18454 (0.03017) (0.01719)	0.28699 (0.03655) (0.02181)	0.19159 (0.02812) (0.011)	2.46951 (0.2529) (0.20179)
	c=3, k=1.1/3	0.18704 (0.02337) (0.01505)	0.66801 (0.08345) (0.05374)	18.0424 (2.85878) (2.55498)	34.6911 (8.45049) (5.99457)	29.798 (4.04433) (2.1011)	–	–	0.45244 (0.06809) (0.03639)	0.30454 (0.03964) (0.03095)	4.09866 (0.54837) (0.46642)	0.41228 (0.06119) (0.03727)	7.01512 (0.69638) (0.35706)
Lognormal	μ=0.5, σ=0.5	0.10128 (0.01228) (0.00473)	0.17186 (0.01582) (0.01334)	0.09624 (0.01021) (0.00974)	0.13358 (0.01283) (0.00851)	0.12006 (0.00657) (0.00073)	0.07846 (0.02061) (0.01474)	–	0.12205 (0.00929) (0.00606)	0.05827 (0.00801) (0.00784)	0.07462 (0.01) (0.00766)	0.05393 (0.00626) (0.00285)	0.08809 (0.01519) (0.0081)
	μ=0.6, σ=1	0.67495 (0.12159) (0.0874)	0.86666 (0.14016) (0.12043)	0.93739 (0.14947) (0.09823)	1.15426 (0.17689) (0.08067)	1.01586 (0.15034) (0.08112)	0.96128 (0.16757) (0.1046)	–	0.60144 (0.08582) (0.04735)	0.37202 (0.03612) (0.01612)	0.42682 (0.05022) (0.03154)	0.28078 (0.04002) (0.027)	0.76375 (0.08616) (0.05823)
	μ=1.5, σ=0.8	0.46076 (0.07468) (0.05485)	0.45303 (0.06776) (0.03529)	0.80036 (0.11997) (0.05919)	0.47485 (0.06253) (0.01949)	0.39948 (0.05333) (0.02002)	0.33859 (0.03789) (0.02311)	–	0.31689 (0.03649) (0.0293)	0.46585 (0.04046) (0.02047)	0.64751 (0.08284) (0.0447)	0.26909 (0.02534) (0.02017)	0.36169 (0.03628) (0.02246)

TABLES OBTAINED FOR THE THEIL MEASURE FOR n=15 000

Table A33: Mean – true Theil (Theil, n=15000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07121 (0.00033)	0.07908 (0.00051)	-0.01758 (0.0003)	0.03478 (0.00061)	0.05872 (0.00057)	–	–	0.0436 (0.00021)	0.07737 (0.00033)	0.00037 (0.0005)	-0.0566 (0.00019)	0.00002 (0.00037)
	α=1.5	-0.69824 (0.00216)	-0.66293 (0.00333)	-0.18573 (0.01226)	-0.6303 (0.01056)	-0.61939 (0.01496)	-0.59747 (0.02232)	-0.51035 (0.06131)	-0.73087 (0.00208)	-0.59875 (0.0019)	-0.05984 (0.02372)	-0.78896 (0.00172)	-0.06026 (0.03915)
	α=1.1	-6.42824 (0.01502)	-6.28831 (0.01652)	-4.35593 (0.07247)	-5.28102 (0.0905)	-5.36298 (0.09826)	-5.45274 (0.10153)	-5.50068 (0.07876)	-6.71201 (0.01033)	-6.35308 (0.01636)	-5.29732 (0.06651)	-6.91868 (0.00988)	-4.915 (0.10467)

Burr	c=1.2, k=2.5	-0.44499 (0.00049)	-0.43627 (0.0007)	-0.50252 (0.00038)	-0.46436 (0.00067)	-0.45331 (0.36842)	–	–	-0.46579 (0.00042)	-0.44187 (0.00065)	-0.48457 (0.00054)	-0.52408 (0.0003)	-0.00007 (0.00095)
	c=3, k=1	-0.04555 (0.0006)	-0.03738 (0.00059)	-0.13103 (0.00039)	-0.07999 (0.00066)	-0.05816 (0.00053)	–	–	-0.07309 (0.00035)	-0.03982 (0.00064)	-0.11321 (0.00051)	-0.16824 (0.00028)	-9.76E-0.7 (0.00045)
	c=1.3, k=1.5	-0.45545 (0.00067)	-0.36685 (0.00146)	-0.50827 (0.00185)	-0.35017 (0.00339)	-0.36465 (0.0035)	–	–	-0.49227 (0.00086)	-0.44756 (0.00069)	-0.46483 (0.00661)	-0.6437 (0.00069)	-0.0058 (0.0093)
	c=3, k=1.1/3	-7.22022 (0.00036)	-6.99627 (0.00065)	-5.86244 (0.00802)	-5.69098 (0.01083)	-5.80792 (0.01444)	–	–	-7.1725 (0.00068)	-7.20873 (0.00035)	-5.18981 (0.08252)	-7.27451 (0.00114)	-5.06337 (0.11338)
Lognormal	μ=0.5, σ=0.5	0.05904 (0.00009)	0.06099 (0.00018)	-0.02669 (0.00011)	0.02032 (0.00024)	0.05176 (0.00012)	0.05821 (0.00011)	–	0.05157 (0.00008)	0.06951 (0.0001)	-0.01156 (0.00018)	-0.07385 (0.00007)	0.00005 (0.00016)
	μ=0.6, σ=1	-0.01481 (0.00106)	0.02602 (0.00161)	-0.06051 (0.00243)	0.00898 (0.00228)	-0.02296 (0.00198)	-0.02962 (0.00172)	–	-0.07642 (0.00114)	0.00026 (0.00109)	-0.0165 (0.01429)	-0.19251 (0.00129)	-0.00001 (0.00148)
	μ=1.5, σ=0.8	0.03809 (0.0003)	0.05865 (0.00059)	0.06325 (0.00131)	0.00349 (0.00077)	-0.00802 (0.0007)	-0.01503 (0.00062)	–	-0.01623 (0.00034)	0.05654 (0.00031)	0.09994 (0.01077)	-0.0661 (0.00039)	0.00006 (0.00034)

Table A34: Median – true Theil (Theil, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07149 (0.00043)	0.07875 (0.00091)	-0.01778 (0.00056)	0.0344 (0.00107)	0.05818 (0.00092)	–	–	0.04362 (0.0003)	0.07747 (0.00052)	-0.0004 (0.00073)	-0.05663 (0.00035)	-0.00066 (0.00053)
	α=1.5	-0.69988 (0.00444)	-0.6679 (0.00376)	-0.21238 (0.01218)	-0.65554 (0.0086)	-0.6544 (0.0088)	-0.65307 (0.01081)	-0.64768 (0.01856)	-0.73385 (0.00318)	-0.59977 (0.00365)	-0.14415 (0.01094)	-0.79103 (0.00258)	-0.14339 (0.0285)
	α=1.1	-6.42039 (0.01531)	-6.28478 (0.01829)	-4.41023 (0.04116)	-5.45375 (0.04433)	-5.59105 (0.04344)	-5.72041 (0.05019)	-5.83976 (0.03962)	-6.71116 (0.01056)	-6.34442 (0.02109)	-5.51142 (0.03516)	-6.9186 (0.01341)	-5.3061 (0.08495)
Burr	c=1.2, k=2.5	-0.44481 (0.00072)	-0.437 (0.00126)	-0.50316 (0.00066)	-0.46522 (0.00105)	-0.45441 (4.41504)	–	–	-0.46609 (0.00066)	-0.44183 (0.00091)	-0.4856 (0.00056)	-0.52429 (0.0004)	-0.00121 (0.00172)
	c=3, k=1	-0.04549 (0.00059)	-0.03792 (0.00076)	-0.13138 (0.00062)	-0.0807 (0.00117)	-0.05909 (0.00067)	–	–	-0.07327 (0.00023)	-0.03977 (0.00058)	-0.11382 (0.00075)	-0.16845 (0.00046)	-0.0012 (0.00074)
	c=1.3, k=1.5	-0.45535 (0.00098)	-0.36698 (0.00142)	-0.50866 (0.0017)	-0.35182 (0.00319)	-0.36751 (0.00206)	–	–	-0.49252 (0.00127)	-0.44749 (0.00076)	-0.48391 (0.00537)	-0.644 (0.00082)	-0.02611 (0.00491)
	c=3, k=1.1/3	-7.22037 (0.00039)	-6.99647 (0.00084)	-5.86232 (0.01196)	-5.69967 (0.01667)	-5.82406 (0.02151)	–	–	-7.17285 (0.00093)	-7.20894 (0.0003)	-5.40303 (0.09688)	-7.27488 (0.00138)	-5.36768 (0.08979)
Lognormal	μ=0.5, σ=0.5	0.05911 (0.00011)	0.06086 (0.00024)	-0.02664 (0.00012)	0.02025 (0.00033)	0.05173 (0.00018)	0.05821 (0.00017)	–	0.0516 (0.0001)	0.06951 (0.00018)	-0.01164 (0.0002)	-0.07382 (0.00013)	0.00005 (0.00027)
	μ=0.6, σ=1	-0.01495 (0.00129)	0.02549 (0.00221)	-0.06198 (0.00292)	0.00785 (0.0029)	-0.02371 (0.00265)	-0.03023 (0.00199)	–	-0.07663 (0.00148)	0.00021 (0.00131)	-0.04161 (0.00729)	-0.19262 (0.00152)	-0.00045 (0.00186)
	μ=1.5, σ=0.8	0.03782 (0.00055)	0.05832 (0.00062)	0.06151 (0.00127)	0.00292 (0.00084)	-0.00839 (0.00067)	-0.01542 (0.00054)	–	-0.01648 (0.0004)	0.05627 (0.00053)	0.08053 (0.00289)	-0.06627 (0.00066)	-0.00025 (0.00052)

Table A35: RMSE (Theil, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07142 (0.00032)	0.07944 (0.00049)	0.01817 (0.00034)	0.03566 (0.00057)	0.05914 (0.00056)	–	–	0.04376 (0.0002)	0.07759 (0.00032)	0.00651 (0.00092)	0.05668 (0.0002)	0.0056 (0.00039)
	α=1.5	0.69879 (0.00213)	0.6642 (0.00322)	0.22955 (0.00989)	0.63822 (0.00745)	0.63177 (0.00749)	0.62239 (0.00846)	0.63985 (0.03608)	0.73135 (0.00204)	0.59937 (0.00187)	0.33439 (0.10738)	0.78929 (0.00168)	0.34049 (0.13778)
	α=1.1	6.42969 (0.01513)	6.29042 (0.01659)	4.43921 (0.06401)	5.36936 (0.07598)	5.46007 (0.0809)	5.54859 (0.08384)	5.59365 (0.06576)	6.71306 (0.01037)	6.35461 (0.01647)	5.33919 (0.05876)	6.91957 (0.0099)	5.06597 (0.08926)
Burr	c=1.2, k=2.5	0.44504 (0.00049)	0.43637 (0.0007)	0.50254 (0.00038)	0.46445 (0.00066)	6.24437 (0.26085)	–	–	0.46583 (0.00042)	0.44193 (0.00065)	0.48462 (0.00053)	0.52409 (0.0003)	0.01672 (0.00155)
	c=3, k=1	0.04586 (0.0006)	0.03811 (0.00055)	0.13107 (0.00038)	0.08034 (0.00065)	0.05856 (0.00051)	–	–	0.07316 (0.00034)	0.04024 (0.00062)	0.11333 (0.00049)	0.16823 (0.00028)	0.00845 (0.00131)
	c=1.3, k=1.5	0.45551 (0.00068)	0.36723 (0.00146)	0.50868 (0.00183)	0.35193 (0.00325)	0.36631 (0.00332)	–	–	0.49237 (0.00086)	0.44763 (0.00069)	0.47147 (0.00553)	0.64375 (0.00068)	0.09997 (0.02639)
	c=3, k=1.1/3	7.22023 (0.00036)	6.99628 (0.00065)	5.86308 (0.00802)	5.69227 (0.01087)	5.81007 (0.01441)	–	–	7.17251 (0.00068)	7.20874 (0.00035)	5.2514 (0.07656)	7.27452 (0.00114)	5.17863 (0.08913)
Lognormal	μ=0.5, σ=0.5	0.05906 (0.00009)	0.06103 (0.00018)	0.02674 (0.00011)	0.02052 (0.00023)	0.05179 (0.00011)	0.05823 (0.00011)	–	0.05159 (0.00008)	0.06953 (0.0001)	0.01173 (0.00017)	0.07386 (0.00007)	0.00172 (0.00013)
	μ=0.6, σ=1	0.01632 (0.00103)	0.02851 (0.0015)	0.06341 (0.00234)	0.01963 (0.00155)	0.02737 (0.00173)	0.03235 (0.00156)	–	0.07681 (0.00115)	0.0073 (0.00043)	0.12711 (0.07546)	0.19268 (0.0013)	0.01043 (0.00087)
	μ=1.5, σ=0.8	0.03838 (0.00032)	0.05915 (0.0006)	0.06491 (0.00136)	0.00995 (0.00076)	0.01173 (0.00064)	0.01682 (0.00058)	–	0.01702 (0.00034)	0.05675 (0.00031)	0.13819 (0.06114)	0.06631 (0.00038)	0.00565 (0.00045)

Table A36: MAD (Theil, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00359 (0.00042) (0.00029)	0.00514 (0.00075) (0.00048)	0.00314 (0.00053) (0.00034)	0.00542 (0.00075) (0.00067)	0.00471 (0.00054) (0.00041)	–	–	0.0026 (0.0003) (0.00018)	0.00376 (0.00049) (0.00029)	0.00376 (0.00051) (0.00054)	0.00193 (0.00026) (0.00012)	0.00334 (0.00037) (0.00016)
	α=1.5	0.01862 (0.00246) (0.00176)	0.02812 (0.0033) (0.0029)	0.08477 (0.01235) (0.01004)	0.05263 (0.00664) (0.00458)	0.05388 (0.00641) (0.00435)	0.05747 (0.00566) (0.00394)	0.06571 (0.0112) (0.00484)	0.01823 (0.00242) (0.00193)	0.01851 (0.00241) (0.00211)	0.08068 (0.01371) (0.0049)	0.01476 (0.0014) (0.00097)	0.08689 (0.01246) (0.00922)
	α=1.1	0.08355 (0.01336) (0.00855)	0.11412 (0.00603) (0.00469)	0.75921 (0.09619) (0.04552)	0.68928 (0.07418) (0.06101)	0.67175 (0.07311) (0.04008)	0.6254 (0.06867) (0.03177)	0.56949 (0.04714) (0.02923)	0.08461 (0.0066) (0.00649)	0.08658 (0.01139) (0.00827)	0.25926 (0.04324) (0.03053)	0.07718 (0.00537) (0.00338)	0.45732 (0.07102) (0.04118)
Burr	c=1.2, k=2.5	0.00437 (0.00052) (0.00026)	0.00654 (0.00066) (0.00037)	0.00359 (0.00033) (0.00027)	0.00658 (0.00057) (0.00027)	0.77575 (0.66363) (0.35217)	–	–	0.00364 (0.00045) (0.00033)	0.00468 (0.00049) (0.00012)	0.00398 (0.0003) (0.00026)	0.00275 (0.00021) (0.00015)	0.01087 (0.00109) (0.00069)
	c=3, k=1	0.00355 (0.00032) (0.00017)	0.00483 (0.0004) (0.00033)	0.00292 (0.00024) (0.00014)	0.00504 (0.00048) (0.00038)	0.0046 (0.00059) (0.00032)	–	–	0.00266 (0.00034) (0.00018)	0.00403 (0.0003) (0.00012)	0.00353 (0.00042) (0.00021)	0.00188 (0.00013) (0.0001)	0.00455 (0.00056) (0.00045)
	c=1.3, k=1.5	0.00582 (0.00077) (0.00061)	0.01105 (0.00111) (0.00071)	0.01339 (0.00113) (0.00071)	0.0231 (0.00197) (0.00129)	0.02318 (0.00188) (0.00147)	–	–	0.00691 (0.00087) (0.00078)	0.006 (0.00067) (0.00037)	0.0272 (0.00274) (0.00177)	0.00558 (0.00074) (0.00038)	0.03436 (0.004) (0.00278)

	c=3, k=1.1/3	0.00265 (0.00015) (0.00006)	0.00516 (0.00055) (0.00053)	0.05632 (0.00912) (0.00525)	0.08504 (0.00973) (0.00671)	0.10299 (0.01094) (0.00848)	–	–	0.00513 (0.00055) (0.0003)	0.00275 (0.00022) (0.00013)	0.44221 (0.04652) (0.03037)	0.00713 (0.00086) (0.00068)	0.42944 (0.05764) (0.02971)
Lognormal	μ=0.5, σ=0.5	0.00104 (0.00011) (0.00009)	0.0013 (0.00016) (0.00012)	0.00105 (0.00013) (0.00011)	0.00186 (0.00026) (0.00016)	0.00107 (0.00013) (0.00011)	0.00095 (0.00011) (0.00007)	–	0.00084 (0.0001) (0.00007)	0.00106 (0.00009) (0.00004)	0.0012 (0.0001) (0.00008)	0.00082 (0.00012) (0.00012)	0.00117 (0.00016) (0.00015)
	μ=0.6, σ=1	0.00439 (0.00054) (0.00041)	0.00762 (0.0007) (0.00014)	0.012 (0.0011) (0.00078)	0.0111 (0.00081) (0.00039)	0.00947 (0.00069) (0.00036)	0.00836 (0.00057) (0.00029)	–	0.00517 (0.00061) (0.00026)	0.00483 (0.00063) (0.00032)	0.02503 (0.00425) (0.00295)	0.00546 (0.00086) (0.00073)	0.0067 (0.00081) (0.00044)
	μ=1.5, σ=0.8	0.0031 (0.00046) (0.00026)	0.00528 (0.0005) (0.00049)	0.00965 (0.00083) (0.00062)	0.00602 (0.0005) (0.00033)	0.00547 (0.00048) (0.00024)	0.00498 (0.00048) (0.00031)	–	0.00339 (0.00033) (0.00024)	0.00325 (0.00052) (0.00027)	0.01788 (0.00267) (0.00188)	0.00349 (0.00038) (0.00029)	0.00386 (0.00036) (0.00022)

TABLES OBTAINED FOR THE THEIL MEASURE FOR n=10 000

Table A37: Mean – true Theil (Theil, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07064 (0.00062)	0.07968 (0.00069)	-0.01737 (0.00044)	0.03523 (0.00075)	0.05971 (0.00074)	–	–	0.04359 (0.00038)	0.07673 (0.00073)	0.00057 (0.00046)	-0.05665 (0.00029)	0.00066 (0.00101)
	α=1.5	-0.70244 (0.00298)	-0.66726 (0.00397)	-0.19598 (0.01207)	-0.63738 (0.00894)	-0.62789 (0.01089)	-0.60344 (0.01904)	-0.5213 (0.02902)	-0.73399 (0.00271)	-0.60238 (0.00299)	-0.09143 (0.03168)	-0.79158 (0.00249)	-0.08215 (0.03242)
	α=1.1	-6.47558 (0.02052)	-6.33427 (0.02498)	-4.41463 (0.11932)	-5.34023 (0.13493)	-5.42789 (0.14329)	-5.5063 (0.14141)	-5.47256 (0.16745)	-6.74225 (0.02057)	-6.39847 (0.0199)	-5.36864 (0.12087)	-6.94357 (0.01883)	-5.0678 (0.12189)
Burr	c=1.2, k=2.5	-0.446 (0.00135)	-0.43548 (0.00163)	-0.50231 (0.00093)	-0.46342 (0.00162)	-0.45166 (0.43613)	–	–	-0.46594 (0.00105)	-0.44305 (0.00121)	-0.48383 (0.00237)	-0.52416 (0.0007)	-0.00083 (0.00264)
	c=3, k=1	-0.04593 (0.00078)	-0.03665 (0.00113)	-0.13071 (0.00065)	-0.07939 (0.0011)	-0.0572 (0.00089)	–	–	-0.073 (0.00056)	-0.04023 (0.00078)	-0.11261 (0.00112)	-0.16818 (0.00036)	0.00025 (0.00115)
	c=1.3, k=1.5	-0.45526 (0.00116)	-0.36616 (0.00149)	-0.50707 (0.00162)	-0.34744 (0.00261)	-0.36156 (0.00222)	–	–	-0.49186 (0.00118)	-0.44732 (0.00114)	-0.4652 (0.0121)	-0.64337 (0.00097)	-0.00266 (0.01551)
	c=3, k=1.1/3	-7.22035 (0.00055)	-6.99635 (0.00084)	-5.85855 (0.00703)	-5.68627 (0.01229)	-5.79877 (0.01582)	–	–	-7.17252 (0.00087)	-7.20882 (0.00058)	-5.24877 (0.05902)	-7.27388 (0.00127)	-5.00147 (0.10131)
Lognormal	μ=0.5, σ=0.5	0.05904 (0.00027)	0.06146 (0.00028)	-0.02664 (0.00025)	0.02083 (0.0004)	0.05193 (0.00028)	0.05843 (0.00029)	–	0.05158 (0.00021)	0.06951 (0.00023)	-0.01136 (0.00023)	-0.07381 (0.00021)	0.00005 (0.0003)
	μ=0.6, σ=1	-0.01476 (0.00066)	0.02671 (0.00114)	-0.05887 (0.002)	0.01042 (0.00176)	-0.02177 (0.00149)	-0.02851 (0.00128)	–	-0.07618 (0.00073)	0.00037 (0.00082)	-0.02259 (0.01657)	-0.19229 (0.00078)	-0.0005 (0.00111)
	μ=1.5, σ=0.8	0.03804 (0.00035)	0.05827 (0.00064)	0.06255 (0.00134)	0.00306 (0.0008)	-0.00843 (0.00072)	-0.01542 (0.00064)	–	-0.01636 (0.00039)	0.05656 (0.00058)	0.09767 (0.01075)	-0.06609 (0.00053)	-0.00013 (0.00035)

Table A38: Median – true Theil (Theil, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07109 (0.00084)	0.07928 (0.00136)	-0.01773 (0.0009)	0.03472 (0.0014)	0.05901 (0.00121)	–	–	0.04359 (0.0006)	0.0771 (0.00084)	-0.00021 (0.00086)	-0.05663 (0.0006)	-0.0006 (0.00086)
	α=1.5	-0.70276 (0.00456)	-0.67021 (0.0038)	-0.21948 (0.01703)	-0.66106 (0.01219)	-0.6602 (0.01334)	-0.65885 (0.018)	-0.6089 (0.02488)	-0.7356 (0.00379)	-0.60295 (0.00408)	-0.16266 (0.01745)	-0.79265 (0.00293)	-0.16535 (0.0178)
	α=1.1	-6.45291 (0.02962)	-6.33064 (0.0411)	-4.73811 (0.13516)	-5.76293 (0.12768)	-5.9 (0.11703)	-6.0272 (0.11767)	-6.08009 (0.30252)	-6.74271 (0.03782)	-6.38713 (0.02556)	-5.56869 (0.0728)	-6.94268 (0.03396)	-5.40086 (0.09938)
Burr	c=1.2, k=2.5	-0.4459 (0.00144)	-0.43679 (0.00178)	-0.50307 (0.00082)	-0.46503 (0.00154)	-0.45332 (2.69063)	–	–	-0.46648 (0.00117)	-0.4428 (0.00141)	-0.48562 (0.00122)	-0.52455 (0.00069)	-0.00361 (0.00301)
	c=3, k=1	-0.04568 (0.00068)	-0.03736 (0.00148)	-0.13121 (0.00079)	-0.08052 (0.00149)	-0.05852 (0.00105)	–	–	-0.07337 (0.00059)	-0.04016 (0.00106)	-0.11387 (0.00088)	-0.1685 (0.00039)	-0.00108 (0.001)
	c=1.3, k=1.5	-0.45544 (0.0014)	-0.36654 (0.00159)	-0.50816 (0.00176)	-0.34935 (0.00289)	-0.36392 (0.00271)	–	–	-0.49232 (0.00138)	-0.44743 (0.00141)	-0.482 (0.00675)	-0.64385 (0.00131)	-0.02718 (0.00722)
	c=3, k=1.1/3	-7.22045 (0.00068)	-6.99638 (0.00096)	-5.85815 (0.01034)	-5.69578 (0.01523)	-5.81459 (0.02037)	–	–	-7.17271 (0.00104)	-7.20894 (0.00053)	-5.50482 (0.05382)	-7.27411 (0.00128)	-5.38838 (0.07846)
Lognormal	μ=0.5, σ=0.5	0.05901 (0.00029)	0.06133 (0.00045)	-0.02667 (0.00032)	0.02073 (0.00048)	0.05185 (0.00028)	0.0584 (0.00038)	–	0.05154 (0.00025)	0.06946 (0.00035)	-0.01155 (0.00035)	-0.07386 (0.00024)	0.00006 (0.00034)
	μ=0.6, σ=1	-0.01437 (0.00036)	0.02723 (0.00116)	-0.05909 (0.00219)	0.01021 (0.00195)	-0.02172 (0.00158)	-0.02839 (0.0015)	–	-0.07581 (0.00058)	0.00064 (0.00087)	-0.0451 (0.0061)	-0.19207 (0.00083)	-0.00107 (0.00137)
	μ=1.5, σ=0.8	0.03791 (0.00067)	0.05766 (0.00073)	0.0609 (0.00143)	0.00239 (0.00086)	-0.00893 (0.00072)	-0.01594 (0.00063)	–	-0.01661 (0.00075)	0.05646 (0.00091)	0.07558 (0.00303)	-0.06658 (0.00071)	-0.00051 (0.00039)

Table A39: RMSE (Theil, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07099 (0.0006)	0.08014 (0.00067)	0.0182 (0.00043)	0.03637 (0.00071)	0.06027 (0.00072)	–	–	0.04382 (0.00039)	0.07708 (0.00071)	0.00696 (0.00054)	0.05676 (0.00029)	0.00953 (0.00433)
	α=1.5	0.70323 (0.00294)	0.66876 (0.00387)	0.23936 (0.00883)	0.64501 (0.00754)	0.63917 (0.008)	0.6347 (0.01228)	0.60893 (0.0222)	0.73462 (0.00267)	0.60329 (0.00295)	0.33422 (0.15428)	0.79198 (0.00245)	0.31988 (0.0909)
	α=1.1	6.47802 (0.02056)	6.33743 (0.025)	4.49817 (0.11276)	5.42812 (0.12603)	5.52406 (0.13361)	5.60677 (0.13188)	5.55328 (0.15348)	6.74382 (0.02059)	6.40104 (0.01996)	5.41294 (0.1062)	6.94483 (0.01883)	5.18966 (0.09475)
Burr	c=1.2, k=2.5	0.44609 (0.00135)	0.43562 (0.00162)	0.50235 (0.00092)	0.46355 (0.0016)	5.90682 (0.30082)	–	–	0.46599 (0.00105)	0.44316 (0.00121)	0.48425 (0.00149)	0.52418 (0.00069)	0.02046 (0.00241)
	c=3, k=1	0.04644 (0.00078)	0.03771 (0.00107)	0.13079 (0.00065)	0.0799 (0.00108)	0.05783 (0.00085)	–	–	0.07312 (0.00055)	0.04088 (0.00077)	0.11301 (0.00088)	0.16818 (0.00036)	0.0111 (0.00593)
	c=1.3, k=1.5	0.45538 (0.00116)	0.36676 (0.00151)	0.50773 (0.00163)	0.35034 (0.00264)	0.36433 (0.00226)	–	–	0.49203 (0.00119)	0.44744 (0.00115)	0.47416 (0.00558)	0.64345 (0.00098)	0.12745 (0.10242)
	c=3, k=1.1/3	7.22036 (0.00055)	6.99636 (0.00084)	5.85949 (0.00694)	5.68819 (0.01209)	5.80191 (0.01542)	–	–	7.17253 (0.00087)	7.20883 (0.00058)	5.32096 (0.05352)	7.2739 (0.00126)	5.14376 (0.07882)
Lognormal	μ=0.5, σ=0.5	0.05908 (0.00027)	0.06151 (0.00028)	0.02673 (0.00025)	0.02111 (0.00041)	0.05197 (0.00028)	0.05846 (0.00029)	–	0.05161 (0.00021)	0.06955 (0.00023)	0.01178 (0.00039)	0.07383 (0.00021)	0.00208 (0.00011)
	μ=0.6, σ=1	0.01684 (0.00061)	0.03034 (0.00112)	0.06345 (0.00192)	0.02404 (0.00175)	0.02859 (0.00147)	0.03284 (0.00131)	–	0.07676 (0.00072)	0.0087 (0.00065)	0.11139 (0.0735)	0.19253 (0.00077)	0.01281 (0.00085)
	μ=1.5, σ=0.8	0.03844 (0.00034)	0.05896 (0.00065)	0.06476 (0.00151)	0.01122 (0.00102)	0.01306 (0.00073)	0.01774 (0.0006)	–	0.01744 (0.00042)	0.05687 (0.00058)	0.13661 (0.04725)	0.06641 (0.00053)	0.00686 (0.00053)

Table A40: MAD (Theil, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00445 (0.00072) (0.00037)	0.00599 (0.00066) (0.00036)	0.00367 (0.0004) (0.00028)	0.00626 (0.00055) (0.00041)	0.00534 (0.00073) (0.00043)	–	–	0.00309 (0.0004) (0.00022)	0.00489 (0.00067) (0.00055)	0.0041 (0.00036) (0.00023)	0.00237 (0.00024) (0.00007)	0.00418 (0.00061) (0.0004)
	α=1.5	0.0234 (0.00179) (0.00131)	0.02995 (0.00208) (0.00119)	0.08302 (0.00699) (0.00483)	0.04963 (0.00455) (0.00362)	0.05132 (0.00485) (0.00382)	0.05555 (0.00699) (0.00473)	0.07649 (0.01673) (0.00863)	0.02101 (0.00168) (0.00135)	0.02276 (0.00214) (0.00208)	0.07559 (0.01207) (0.0074)	0.01717 (0.00152) (0.00085)	0.09485 (0.01374) (0.01313)
	α=1.1	0.13496 (0.01499) (0.00768)	0.15251 (0.00969) (0.00647)	0.46715 (0.17481) (0.0901)	0.44759 (0.19957) (0.08667)	0.42126 (0.22394) (0.08459)	0.38616 (0.22622) (0.09236)	0.27163 (0.25593) (0.05042)	0.10969 (0.00872) (0.00649)	0.14423 (0.01672) (0.01329)	0.27596 (0.06685) (0.0519)	0.09574 (0.00919) (0.00594)	0.42419 (0.0784) (0.06955)
Burr	c=1.2, k=2.5	0.00593 (0.00088) (0.00038)	0.00692 (0.00083) (0.00061)	0.00394 (0.00052) (0.00039)	0.00714 (0.00093) (0.00055)	0.1575 (0.41898) (0.00585)	–	–	0.00456 (0.00055) (0.00028)	0.00629 (0.00077) (0.00033)	0.00447 (0.00047) (0.00027)	0.00324 (0.00047) (0.00021)	0.01196 (0.00163) (0.0011)
	c=3, k=1	0.00475 (0.00041) (0.00027)	0.0062 (0.00066) (0.00042)	0.00376 (0.0003) (0.00014)	0.00635 (0.00075) (0.0003)	0.00579 (0.00065) (0.0004)	–	–	0.00333 (0.00039) (0.00016)	0.00499 (0.00035) (0.00016)	0.00419 (0.0004) (0.00019)	0.00256 (0.00023) (0.00011)	0.00575 (0.00064) (0.00047)
	c=1.3, k=1.5	0.00742 (0.00057) (0.00018)	0.01383 (0.00174) (0.00054)	0.0166 (0.00258) (0.00201)	0.02761 (0.004) (0.00226)	0.02707 (0.00391) (0.00367)	–	–	0.00864 (0.00098) (0.00066)	0.00747 (0.00091) (0.00065)	0.03422 (0.00459) (0.00267)	0.00718 (0.00087) (0.00045)	0.04084 (0.00625) (0.00308)
	c=3, k=1.1/3	0.0032 (0.00038) (0.00011)	0.00594 (0.00081) (0.00056)	0.0695 (0.00714) (0.00447)	0.09775 (0.01109) (0.00715)	0.12266 (0.0147) (0.00793)	–	–	0.00618 (0.00079) (0.00049)	0.00339 (0.00037) (0.0002)	0.44246 (0.07398) (0.06575)	0.00851 (0.00091) (0.00058)	0.47481 (0.06128) (0.04719)
Lognormal	μ=0.5, σ=0.5	0.00131 (0.00017) (0.00013)	0.00168 (0.00019) (0.00013)	0.0014 (0.00012) (0.00008)	0.0023 (0.00023) (0.00016)	0.00134 (0.00016) (0.00012)	0.00129 (0.00021) (0.00016)	–	0.00106 (0.00014) (0.00007)	0.00139 (0.00015) (0.0001)	0.00168 (0.00014) (0.00007)	0.00103 (0.00009) (0.00002)	0.00146 (0.00013) (0.00006)
	μ=0.6, σ=1	0.00547 (0.00078) (0.00052)	0.00924 (0.00178) (0.00094)	0.0154 (0.00305) (0.00253)	0.01408 (0.00288) (0.00211)	0.01198 (0.00235) (0.00141)	0.01047 (0.00216) (0.0015)	–	0.00619 (0.00101) (0.00071)	0.00595 (0.00087) (0.00065)	0.02749 (0.00332) (0.002)	0.00648 (0.00081) (0.00046)	0.00902 (0.00076) (0.00038)
	μ=1.5, σ=0.8	0.00375 (0.00036) (0.00026)	0.00573 (0.00068) (0.00048)	0.01075 (0.00163) (0.00124)	0.00686 (0.00094) (0.00041)	0.00631 (0.00082) (0.00036)	0.00558 (0.00074) (0.00042)	–	0.00407 (0.00046) (0.0004)	0.00403 (0.00033) (0.00022)	0.01702 (0.00232) (0.00127)	0.00434 (0.00042) (0.00026)	0.00436 (0.00029) (0.00023)

TABLES OBTAINED FOR THE THEIL MEASURE FOR n=5 000

Table A41: Mean – true Theil (Theil, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.06882 (0.00102)	0.0816 (0.00089)	-0.01676 (0.00065)	0.0365 (0.00108)	0.06367 (0.00102)	–	–	0.04363 (0.00059)	0.07498 (0.0011)	0.00141 (0.00136)	-0.05683 (0.00046)	-0.0002 (0.00096)
	α=1.5	-0.70949 (0.00495)	-0.66822 (0.00683)	-0.1763 (0.02136)	-0.6158 (0.01881)	-0.59312 (0.06038)	-0.53662 (0.04813)	-0.46077 (0.04688)	-0.73657 (0.00462)	-0.60964 (0.00478)	-0.08163 (0.04045)	-0.79354 (0.00422)	-0.08975 (0.01731)
	α=1.1	-6.55791 (0.01567)	-6.4014 (0.01817)	-4.26414 (0.08526)	-5.12267 (0.08192)	-5.20499 (0.08004)	-5.26372 (0.09835)	-4.65229 (0.08361)	-6.78543 (0.01788)	-6.48315 (0.0182)	-5.37903 (0.07921)	-6.97669 (0.01759)	-5.22034 (0.08156)

Burr	c=1.2, k=2.5	-0.44764 (0.00131)	-0.43265 (0.00122)	-0.50121 (0.00078)	-0.46007 (0.00133)	-0.44526 (0.44832)	–	–	-0.46568 (0.00088)	-0.44473 (0.00124)	-0.48295 (0.00133)	-0.52408 (0.00064)	-0.00056 (0.00314)
	c=3, k=1	-0.04773 (0.00105)	-0.03413 (0.00101)	-0.12975 (0.0007)	-0.07732 (0.00112)	-0.0524 (0.00152)	–	–	-0.07279 (0.00058)	-0.04214 (0.00099)	-0.11144 (0.00145)	-0.16814 (0.00048)	-0.00033 (0.00196)
	c=1.3, k=1.5	-0.45582 (0.00168)	-0.3678 (0.00381)	-0.50846 (0.00475)	-0.34895 (0.00839)	-0.3619 (0.00821)	–	–	-0.49263 (0.00229)	-0.44793 (0.00171)	-0.4684 (0.01846)	-0.64419 (0.00187)	-0.00899 (0.01723)
	c=3, k=1.1/3	-7.22073 (0.00051)	-6.99696 (0.0008)	-5.86541 (0.0127)	-5.69085 (0.01585)	-5.79229 (0.01814)	–	–	-7.17321 (0.00091)	-7.20939 (0.00048)	-5.46572 (0.07694)	-7.27537 (0.00168)	-5.22617 (0.09935)
Lognormal	μ=0.5, σ=0.5	0.05889 (0.00032)	0.06277 (0.00038)	-0.02656 (0.00032)	0.02188 (0.00049)	0.05222 (0.00033)	0.05909 (0.00049)	–	0.05155 (0.00026)	0.06939 (0.00031)	-0.01114 (0.00031)	-0.07393 (0.00025)	-0.00009 (0.00032)
	μ=0.6, σ=1	-0.01495 (0.00103)	0.02633 (0.00172)	-0.05879 (0.00288)	0.01051 (0.00262)	-0.02171 (0.00224)	-0.02829 (0.00192)	–	-0.07641 (0.00116)	0.00035 (0.00105)	-0.02591 (0.01041)	-0.19262 (0.00127)	0.00018 (0.00126)
	μ=1.5, σ=0.8	0.0376 (0.00085)	0.05785 (0.00132)	0.0628 (0.00234)	0.00287 (0.00148)	-0.00877 (0.00137)	-0.01573 (0.00126)	–	-0.01677 (0.00088)	0.0561 (0.00098)	0.09616 (0.01063)	-0.06654 (0.00079)	-0.00042 (0.00103)

Table A42: Median – true Theil (Theil, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.06875 (0.00148)	0.0803 (0.00108)	-0.01743 (0.00081)	0.03528 (0.00131)	0.06152 (0.00104)	–	–	0.04313 (0.00066)	0.07508 (0.0015)	-0.00023 (0.0008)	-0.05727 (0.00069)	-0.0018 (0.00095)
	α=1.5	-0.71407 (0.00699)	-0.6709 (0.00896)	-0.18194 (0.02735)	-0.63357 (0.01754)	-0.6243 (0.01869)	-0.60626 (0.01507)	-0.57626 (0.0107)	-0.74129 (0.00627)	-0.61369 (0.0072)	-0.16998 (0.02382)	-0.79801 (0.00442)	-0.18588 (0.02351)
	α=1.1	-6.61164 (0.02568)	-6.44992 (0.031)	-4.14255 (0.15239)	-5.10846 (0.20438)	-5.22276 (0.18683)	-5.21525 (0.09538)	-4.86042 (0.04248)	-6.81702 (0.02979)	-6.52942 (0.03193)	-5.62969 (0.07526)	-6.99734 (0.0318)	-5.56237 (0.05919)
Burr	c=1.2, k=2.5	-0.44782 (0.00187)	-0.4339 (0.00172)	-0.50221 (0.00097)	-0.46198 (0.00158)	-0.44825 (0.00548)	–	–	-0.46626 (0.0011)	-0.44498 (0.00219)	-0.4849 (0.00156)	-0.52439 (0.00104)	-0.00414 (0.00399)
	c=3, k=1	-0.04823 (0.00149)	-0.03528 (0.00134)	-0.1306 (0.00083)	-0.0788 (0.00135)	-0.05422 (0.0012)	–	–	-0.07325 (0.00068)	-0.04267 (0.00103)	-0.11362 (0.00141)	-0.16864 (0.00068)	-0.00175 (0.00135)
	c=1.3, k=1.5	-0.45554 (0.00218)	-0.36896 (0.00431)	-0.511 (0.0052)	-0.35556 (0.0103)	-0.37005 (0.00882)	–	–	-0.49293 (0.00233)	-0.44801 (0.00174)	-0.4978 (0.00884)	-0.645 (0.00218)	-0.04177 (0.01003)
	c=3, k=1.1/3	-7.22065 (0.00051)	-6.99667 (0.00107)	-5.868 (0.01647)	-5.70307 (0.02193)	-5.83266 (0.0246)	–	–	-7.17283 (0.00125)	-7.20932 (0.00077)	-5.72245 (0.08489)	-7.27524 (0.00204)	-5.63954 (0.07566)
Lognormal	μ=0.5, σ=0.5	0.05877 (0.00027)	0.06241 (0.0004)	-0.02675 (0.00023)	0.02172 (0.00049)	0.05211 (0.00025)	0.05906 (0.00051)	–	0.05144 (0.00027)	0.06932 (0.00045)	-0.01137 (0.00033)	-0.07408 (0.00024)	-0.00024 (0.00037)
	μ=0.6, σ=1	-0.01491 (0.0013)	0.02516 (0.00253)	-0.06305 (0.00349)	0.00692 (0.00331)	-0.02451 (0.00284)	-0.03084 (0.0024)	–	-0.07675 (0.00135)	0.00013 (0.00176)	-0.05722 (0.00703)	-0.19305 (0.00175)	-0.00107 (0.0019)
	μ=1.5, σ=0.8	0.03805 (0.00123)	0.05796 (0.00172)	0.06102 (0.00304)	0.00246 (0.00215)	-0.00886 (0.00211)	-0.0157 (0.00163)	–	-0.01642 (0.00116)	0.0564 (0.00117)	0.07053 (0.00411)	-0.06619 (0.00132)	-0.00051 (0.00136)

Table A43: RMSE (Theil, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.06962 (0.001)	0.08241 (0.00092)	0.01841 (0.00055)	0.03848 (0.00117)	0.06459 (0.00115)	–	–	0.0441 (0.00058)	0.07581 (0.00106)	0.01148 (0.00299)	0.05705 (0.00045)	0.00962 (0.00126)
	α=1.5	0.71104 (0.00485)	0.67085 (0.00663)	0.25071 (0.01595)	0.63482 (0.01307)	0.76539 (0.19607)	0.67516 (0.07592)	0.54602 (0.02282)	0.73778 (0.00452)	0.61143 (0.00462)	0.32758 (0.08467)	0.79434 (0.00413)	0.35241 (0.04264)
	α=1.1	6.56245 (0.01566)	6.40733 (0.01828)	4.39027 (0.09211)	5.25678 (0.09004)	5.34801 (0.08693)	5.39072 (0.09225)	4.71894 (0.07507)	6.78835 (0.0178)	6.48769 (0.01821)	5.45076 (0.06563)	6.97914 (0.01751)	5.34093 (0.06844)
Burr	c=1.2, k=2.5	0.4478 (0.00131)	0.43288 (0.00121)	0.50128 (0.00078)	0.46032 (0.00131)	4.93484 (0.38104)	–	–	0.46576 (0.00087)	0.44491 (0.00124)	0.48311 (0.0013)	0.52412 (0.00064)	0.02816 (0.00422)
	c=3, k=1	0.04896 (0.00106)	0.03643 (0.00091)	0.12998 (0.00069)	0.07842 (0.00105)	0.05387 (0.00128)	–	–	0.07311 (0.00057)	0.04369 (0.00103)	0.11215 (0.00121)	0.16819 (0.00047)	0.01311 (0.00311)
	c=1.3, k=1.5	0.45605 (0.00167)	0.36893 (0.00377)	0.50969 (0.00468)	0.35435 (0.00794)	0.36722 (0.00777)	–	–	0.49295 (0.00226)	0.44818 (0.0017)	0.49111 (0.02492)	0.64434 (0.00186)	0.15333 (0.09152)
	c=3, k=1.1/3	7.22074 (0.00051)	6.99697 (0.0008)	5.86714 (0.01264)	5.69437 (0.01565)	5.79851 (0.01781)	–	–	7.17322 (0.00091)	7.2094 (0.00048)	5.5395 (0.07317)	7.2754 (0.00168)	5.36274 (0.08097)
Lognormal	μ=0.5, σ=0.5	0.05896 (0.00033)	0.06289 (0.00039)	0.02673 (0.00031)	0.02235 (0.00052)	0.0523 (0.00033)	0.05914 (0.0005)	–	0.05159 (0.00026)	0.06945 (0.00032)	0.01176 (0.00023)	0.07397 (0.00024)	0.00294 (0.0003)
	μ=0.6, σ=1	0.01917 (0.00079)	0.03377 (0.00205)	0.06827 (0.00235)	0.0336 (0.00355)	0.03502 (0.00201)	0.03729 (0.00161)	–	0.07766 (0.00111)	0.01259 (0.00074)	0.12374 (0.05946)	0.19312 (0.00124)	0.01897 (0.00143)
	μ=1.5, σ=0.8	0.03839 (0.00084)	0.05915 (0.00136)	0.06691 (0.00265)	0.01513 (0.0013)	0.01633 (0.0009)	0.01992 (0.00093)	–	0.01871 (0.00079)	0.05673 (0.00097)	0.13391 (0.03565)	0.06709 (0.0008)	0.00913 (0.00057)

Table A44: MAD (Theil, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00764 (0.00049) (0.00024)	0.00771 (0.00086) (0.00031)	0.0052 (0.00053) (0.00032)	0.00809 (0.00082) (0.00051)	0.00671 (0.00081) (0.0004)	–	–	0.00425 (0.00044) (0.00024)	0.00813 (0.00057) (0.00043)	0.00585 (0.00065) (0.00057)	0.00344 (0.00032) (0.00016)	0.00573 (0.00061) (0.00041)
	α=1.5	0.03252 (0.00609) (0.00192)	0.03942 (0.00607) (0.0028)	0.09398 (0.01646) (0.0145)	0.06053 (0.00867) (0.0065)	0.06228 (0.00809) (0.00603)	0.06579 (0.00877) (0.00522)	0.03628 (0.00967) (0.00711)	0.02861 (0.00449) (0.00179)	0.03184 (0.00576) (0.00339)	0.09234 (0.01541) (0.00737)	0.02334 (0.00367) (0.00181)	0.11025 (0.0159) (0.00693)
	α=1.1	0.18136 (0.02189) (0.01301)	0.23734 (0.02014) (0.01034)	0.892 (0.09058) (0.04339)	0.96541 (0.05421) (0.0401)	0.9742 (0.05604) (0.03747)	0.97015 (0.07547) (0.05276)	0.48921 (0.03752) (0.01188)	0.16731 (0.01288) (0.00541)	0.17671 (0.01454) (0.00377)	0.35476 (0.05395) (0.00655)	0.14564 (0.01136) (0.00779)	0.45318 (0.03899) (0.02236)
Burr	c=1.2, k=2.5	0.00832 (0.00101) (0.00061)	0.00921 (0.00142) (0.00079)	0.00544 (0.00081) (0.00031)	0.00938 (0.00162) (0.00095)	0.01728 (0.00523) (0.00446)	–	–	0.00606 (0.00091) (0.00057)	0.00929 (0.00094) (0.00071)	0.00604 (0.00076) (0.00045)	0.00418 (0.00055) (0.00026)	0.01582 (0.00208) (0.00145)
	c=3, k=1	0.00765 (0.00085) (0.00041)	0.00824 (0.00082) (0.00053)	0.00514 (0.00039) (0.00034)	0.00809 (0.00064) (0.00042)	0.0076 (0.00093) (0.00038)	–	–	0.00478 (0.00043) (0.0003)	0.00837 (0.00065) (0.00046)	0.00604 (0.00087) (0.00055)	0.00352 (0.00037) (0.00023)	0.00727 (0.00051) (0.00039)
	c=1.3, k=1.5	0.01031 (0.00097) (0.00072)	0.02045 (0.00272) (0.0025)	0.0243 (0.00294) (0.00254)	0.04088 (0.00528) (0.00237)	0.03963 (0.00478) (0.00197)	–	–	0.01288 (0.00163) (0.00128)	0.01041 (0.00102) (0.00087)	0.03783 (0.00636) (0.00611)	0.01015 (0.00133) (0.00086)	0.05217 (0.00657) (0.00412)

	c=3, k=1.1/3	0.00443 (0.00047) (0.0001)	0.00809 (0.00117) (0.00059)	0.10279 (0.01156) (0.00638)	0.13739 (0.01717) (0.01184)	0.17213 (0.02744) (0.02693)	–	–	0.0083 (0.00111) (0.00076)	0.00445 (0.0004) (0.00016)	0.42711 (0.06304) (0.03481)	0.0119 (0.0017) (0.00101)	0.44731 (0.05477) (0.03782)
Lognormal	μ=0.5, σ=0.5	0.00186 (0.0003) (0.00013)	0.00232 (0.00029) (0.0002)	0.00199 (0.00031) (0.00021)	0.00294 (0.00041) (0.00015)	0.00183 (0.00026) (0.00019)	0.00183 (0.00032) (0.00017)	–	0.00148 (0.00031) (0.0001)	0.00198 (0.00027) (0.00018)	0.00224 (0.00012) (0.00005)	0.00151 (0.00022) (0.00012)	0.00198 (0.00017) (0.00012)
	μ=0.6, σ=1	0.00829 (0.00076) (0.00064)	0.01552 (0.0013) (0.00114)	0.02418 (0.0023) (0.00148)	0.02231 (0.00194) (0.00093)	0.01933 (0.00153) (0.00066)	0.01691 (0.00143) (0.00064)	–	0.00986 (0.00113) (0.00069)	0.00877 (0.00129) (0.00094)	0.03083 (0.00359) (0.00171)	0.00998 (0.0011) (0.00084)	0.01247 (0.00098) (0.00071)
	μ=1.5, σ=0.8	0.0053 (0.00063) (0.00028)	0.00867 (0.00114) (0.00087)	0.01577 (0.00215) (0.00116)	0.01034 (0.00129) (0.00085)	0.00958 (0.00115) (0.00084)	0.00847 (0.00116) (0.00093)	–	0.00567 (0.00083) (0.00054)	0.00564 (0.00051) (0.00032)	0.02005 (0.00256) (0.00188)	0.00583 (0.00088) (0.00047)	0.00596 (0.00056) (0.00058)

TABLES OBTAINED FOR THE THEIL MEASURE FOR n=1 000

Table A45: Mean – true Theil (Theil, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.06777 (0.0016)	0.09424 (0.00235)	-0.00939 (0.00154)	0.04917 (0.0026)	0.1274 (0.01869)	–	–	0.04696 (0.00121)	0.07385 (0.00198)	0.00879 (0.00244)	-0.05431 (0.00086)	-0.00019 (0.00185)
	α=1.5	-0.76524 (0.00559)	-0.71238 (0.00786)	-0.21497 (0.02737)	-0.5864 (0.27683)	-0.19812 (0.17511)	-0.24722 (0.3086)	–	-0.77442 (0.00524)	-0.6643 (0.00568)	-0.13829 (0.03985)	-0.82567 (0.00433)	-0.08975 (0.01731)
	α=1.1	-7.00316 (0.01475)	-6.86389 (0.01749)	-5.17961 (0.06746)	-5.98191 (0.08317)	-5.87997 (0.06572)	-4.09758 (0.30663)	–	-7.10464 (0.01409)	-6.92084 (0.01504)	-5.936 (0.06626)	-7.25935 (0.01093)	-5.70393 (0.0909)
Burr	c=1.2, k=2.5	-0.44697 (0.00275)	-0.41813 (0.00472)	-0.49312 (0.00298)	-0.43806 (0.00664)	-0.33627 (0.0379)	–	–	-0.46047 (0.00266)	-0.44418 (0.00302)	-0.47261 (0.00361)	-0.52066 (0.00189)	-0.0025 (0.0058)
	c=3, k=1	-0.04851 (0.00222)	-0.0213 (0.00313)	-0.12227 (0.00215)	-0.0639 (0.00357)	0.01486 (0.01487)	–	–	-0.06922 (0.00176)	-0.04321 (0.00224)	-0.10399 (0.00416)	-0.16552 (0.00111)	-0.00066 (0.00293)
	c=1.3, k=1.5	-0.4569 (0.00294)	-0.36828 (0.00448)	-0.50292 (0.00548)	-0.33014 (0.0106)	-0.32473 (0.01548)	–	–	-0.4924 (0.003)	-0.44877 (0.00317)	-0.4759 (0.01223)	-0.64354 (0.00286)	-0.02254 (0.01869)
	c=3, k=1.1/3	-7.22453 (0.0015)	-7.00537 (0.00248)	-5.9327 (0.02769)	-5.7773 (0.0381)	-5.89754 (0.03932)	–	–	-7.18089 (0.00264)	-7.21297 (0.00147)	-5.94884 (0.07668)	-7.28505 (0.00403)	-5.86191 (0.12045)
Lognormal	μ=0.5, σ=0.5	0.05803 (0.00063)	0.07148 (0.00106)	-0.02381 (0.00077)	0.02672 (0.0011)	0.05613 (0.00072)	0.06801 (0.00361)	–	0.05169 (0.00047)	0.06836 (0.0007)	-0.00694 (0.00145)	-0.07416 (0.0004)	-0.00036 (0.00043)
	μ=0.6, σ=1	-0.01995 (0.00289)	0.03287 (0.00375)	-0.02857 (0.00654)	0.03396 (0.00595)	-0.00558 (0.00505)	-0.01348 (0.00572)	–	-0.07615 (0.00285)	-0.00551 (0.0029)	-0.00027 (0.01597)	-0.1922 (0.00288)	-0.00004 (0.00278)
	μ=1.5, σ=0.8	0.03069 (0.00292)	0.06747 (0.00237)	0.15904 (0.01864)	0.0272 (0.00418)	0.00597 (0.00304)	-0.0066 (0.00319)	–	-0.01655 (0.00208)	0.04858 (0.00274)	0.19879 (0.04342)	-0.06614 (0.00176)	-0.00099 (0.0018)

Table A46: Median – true Theil (Theil, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.06545 (0.00172)	0.09222 (0.00332)	-0.01206 (0.00195)	0.04398 (0.0036)	0.10511 (0.0012)	–	–	0.04555 (0.0014)	0.07218 (0.00194)	0.00293 (0.00179)	-0.05643 (0.001)	-0.00366 (0.0016)
	α=1.5	-0.77726 (0.00595)	-0.73356 (0.00871)	-0.30002 (0.02895)	-0.71369 (0.06479)	-0.57485 (0.0838)	-0.33694 (0.1178)	–	-0.78946 (0.00545)	-0.67704 (0.0059)	-0.26638 (0.01586)	-0.8393 (0.0034)	-0.18588 (0.02351)
	α=1.1	-7.00308 (0.01171)	-6.89073 (0.02196)	-5.64023 (0.10316)	-6.59933 (0.05329)	-6.65429 (0.15912)	-4.2794 (0.38059)	–	-7.12914 (0.01771)	-6.91971 (0.01084)	-6.20953 (0.05419)	-7.28572 (0.0125)	-6.00267 (0.11153)
Burr	c=1.2, k=2.5	-0.45094 (0.00248)	-0.42394 (0.00527)	-0.49885 (0.00308)	-0.45136 (0.00651)	-0.37466 (0.00544)	–	–	-0.46413 (0.00257)	-0.44777 (0.00278)	-0.48221 (0.00302)	-0.52452 (0.0022)	-0.01204 (0.00481)
	c=3, k=1	-0.0517 (0.00288)	-0.0258 (0.00287)	-0.12672 (0.0018)	-0.07176 (0.00359)	-0.00753 (0.00347)	–	–	-0.07182 (0.00216)	-0.04598 (0.00269)	-0.11136 (0.00228)	-0.16861 (0.00088)	-0.00547 (0.00261)
	c=1.3, k=1.5	-0.45652 (0.00342)	-0.36999 (0.00619)	-0.51114 (0.00842)	-0.35894 (0.01563)	-0.37596 (0.01373)	–	–	-0.49311 (0.00351)	-0.44752 (0.00423)	-0.51721 (0.00929)	-0.64603 (0.00242)	-0.07215 (0.00797)
	c=3, k=1.1/3	-7.22388 (0.00193)	-7.00312 (0.0028)	-5.93616 (0.03575)	-5.79238 (0.0501)	-5.91899 (0.03641)	–	–	-7.17955 (0.00324)	-7.21246 (0.00187)	-6.2201 (0.07081)	-7.28552 (0.00492)	-6.13406 (0.08242)
Lognormal	μ=0.5, σ=0.5	0.05782 (0.00097)	0.07003 (0.00158)	-0.02407 (0.00131)	0.02645 (0.00187)	0.05571 (0.00109)	0.06639 (0.00119)	–	0.05151 (0.00067)	0.06812 (0.0011)	-0.0086 (0.00127)	-0.07441 (0.00078)	-0.00046 (0.00068)
	μ=0.6, σ=1	-0.01699 (0.00337)	0.02853 (0.00569)	-0.04337 (0.0094)	0.02279 (0.0085)	-0.01467 (0.0065)	-0.02297 (0.00626)	–	-0.07675 (0.00338)	-0.00312 (0.00372)	-0.05956 (0.00818)	-0.19229 (0.00333)	-0.00316 (0.00459)
	μ=1.5, σ=0.8	0.03405 (0.00343)	0.06636 (0.00214)	0.08992 (0.00822)	0.0183 (0.00237)	0.00236 (0.00238)	-0.00922 (0.0037)	–	-0.01741 (0.0024)	0.05125 (0.00336)	0.08151 (0.01124)	-0.06701 (0.00207)	-0.00213 (0.00202)

Table A47: RMSE (Theil, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.07022 (0.00167)	0.09806 (0.00243)	0.0209 (0.00127)	0.05803 (0.00314)	0.15263 (0.04791)	–	–	0.04905 (0.0012)	0.07633 (0.00217)	0.0273 (0.00464)	0.05531 (0.00081)	0.02043 (0.00405)
	α=1.5	0.76732 (0.00526)	0.71635 (0.00733)	0.33226 (0.01283)	2.45701 (0.37894)	1.23513 (0.30285)	0.54423 (0.67395)	–	0.77631 (0.00494)	0.66654 (0.00531)	0.46686 (0.08199)	0.82696 (0.00405)	0.35241 (0.04264)
	α=1.1	7.00546 (0.01446)	6.86713 (0.01712)	5.26965 (0.06495)	6.09817 (0.07375)	6.02146 (0.06209)	4.11752 (0.31297)	–	7.10646 (0.01391)	6.92315 (0.01472)	5.98982 (0.0508)	7.2609 (0.01081)	5.79006 (0.08535)
Burr	c=1.2, k=2.5	0.44755 (0.00267)	0.41944 (0.0045)	0.49357 (0.00289)	0.44059 (0.00603)	0.37267 (0.02639)	–	–	0.46091 (0.00259)	0.44482 (0.00295)	0.47419 (0.00317)	0.52083 (0.00186)	0.05611 (0.00746)
	c=3, k=1	0.05233 (0.00166)	0.0358 (0.00123)	0.12379 (0.0019)	0.07189 (0.00218)	0.06087 (0.043)	–	–	0.0709 (0.00148)	0.04783 (0.00171)	0.1082 (0.00205)	0.16588 (0.00107)	0.02918 (0.0077)
	c=1.3, k=1.5	0.45828 (0.00299)	0.37382 (0.00451)	0.50926 (0.00524)	0.36221 (0.00843)	0.37908 (0.01709)	–	–	0.49415 (0.00298)	0.45024 (0.00321)	0.50627 (0.0121)	0.64445 (0.00281)	0.20371 (0.0489)
	c=3, k=1.1/3	7.22455 (0.0015)	7.00542 (0.00248)	5.93863 (0.02743)	5.78778 (0.0377)	5.91276 (0.0386)	–	–	7.18094 (0.00264)	7.21298 (0.00147)	6.01516 (0.06229)	7.28514 (0.00403)	5.93391 (0.09989)
Lognormal	μ=0.5, σ=0.5	0.05835 (0.00064)	0.07223 (0.00104)	0.02506 (0.00078)	0.02869 (0.00102)	0.05645 (0.00072)	0.06878 (0.00446)	–	0.05191 (0.00048)	0.06866 (0.00071)	0.01459 (0.00209)	0.07432 (0.0004)	0.00655 (0.00048)
	μ=0.6, σ=1	0.0382 (0.00227)	0.05221 (0.0041)	0.08476 (0.00476)	0.07772 (0.00609)	0.05849 (0.0041)	0.05541 (0.0048)	–	0.08193 (0.00239)	0.03454 (0.00241)	0.21367 (0.06031)	0.19459 (0.00269)	0.03842 (0.00258)
	μ=1.5, σ=0.8	0.03929 (0.0025)	0.07225 (0.00239)	0.23573 (0.0378)	0.05248 (0.00728)	0.03343 (0.0041)	0.02709 (0.00155)	–	0.02573 (0.00158)	0.05497 (0.00265)	0.41557 (0.13907)	0.06924 (0.0015)	0.02162 (0.00173)

Table A48: MAD (Theil, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.01119 (0.00093) (0.00027)	0.01906 (0.00158) (0.00105)	0.01232 (0.00125) (0.00064)	0.02093 (0.0017) (0.00091)	0.00746 (0.00193) (0.00103)	–	–	0.00909 (0.00089) (0.00063)	0.01277 (0.00103) (0.00073)	0.01285 (0.0008) (0.00078)	0.00661 (0.00067) (0.00032)	0.01047 (0.00103) (0.00105)
	α=1.5	0.03353 (0.00403) (0.00295)	0.04521 (0.0052) (0.00309)	0.14791 (0.03446) (0.0205)	0.12636 (0.05434) (0.03849)	0.13586 (0.07313) (0.05334)	0.12769 (0.06538) (0.02882)	–	0.03065 (0.00477) (0.00295)	0.03201 (0.00428) (0.00374)	0.1242 (0.01387) (0.01475)	0.0242 (0.0038) (0.00241)	0.11025 (0.0159) (0.00693)
	α=1.1	0.11413 (0.01501) (0.01342)	0.12108 (0.01104) (0.00774)	0.40122 (0.09972) (0.04227)	0.36513 (0.10396) (0.07571)	0.27695 (0.15478) (0.10434)	0.35641 (0.34548) (0.21983)	–	0.08689 (0.00868) (0.00758)	0.11663 (0.01353) (0.00622)	0.27164 (0.04394) (0.01093)	0.08172 (0.00836) (0.00733)	0.43861 (0.08839) (0.04866)
Burr	c=1.2, k=2.5	0.01366 (0.00136) (0.00082)	0.02211 (0.00273) (0.00227)	0.01294 (0.00193) (0.00119)	0.02654 (0.00416) (0.00257)	0.01253 (0.00528) (0.00425)	–	–	0.0122 (0.0011) (0.00092)	0.0141 (0.00123) (0.00072)	0.01296 (0.00162) (0.00117)	0.00766 (0.00138) (0.00116)	0.03044 (0.00307) (0.00193)
	c=3, k=1	0.01172 (0.00159) (0.00121)	0.01943 (0.00169) (0.001)	0.01222 (0.00133) (0.00094)	0.02071 (0.0019) (0.00164)	0.00922 (0.00226) (0.00162)	–	–	0.0093 (0.00119) (0.00066)	0.01204 (0.00117) (0.00058)	0.01229 (0.00136) (0.00085)	0.00672 (0.00066) (0.00041)	0.01426 (0.00091) (0.00071)
	c=1.3, k=1.5	0.02296 (0.00282) (0.00207)	0.04483 (0.00648) (0.00273)	0.05342 (0.00719) (0.00197)	0.08951 (0.01145) (0.00431)	0.08525 (0.01083) (0.00486)	–	–	0.02807 (0.00428) (0.00143)	0.02396 (0.00262) (0.00168)	0.05729 (0.00637) (0.00237)	0.02258 (0.00315) (0.00055)	0.07554 (0.01002) (0.0043)
	c=3, k=1.1/3	0.00919 (0.00138) (0.00108)	0.016 (0.00263) (0.00207)	0.19784 (0.02212) (0.00912)	0.25285 (0.02887) (0.02207)	0.29355 (0.05077) (0.04473)	–	–	0.01649 (0.00239) (0.0014)	0.009 (0.00178) (0.00088)	0.39029 (0.06071) (0.03552)	0.02305 (0.00296) (0.00146)	0.39175 (0.04009) (0.03305)
Lognormal	μ=0.5, σ=0.5	0.00411 (0.00042) (0.00032)	0.00641 (0.00096) (0.00066)	0.00491 (0.00045) (0.00035)	0.00652 (0.00063) (0.00054)	0.00369 (0.0005) (0.00029)	0.00375 (0.00127) (0.00062)	–	0.00316 (0.00027) (0.00016)	0.00432 (0.00051) (0.00036)	0.00651 (0.00095) (0.00078)	0.0034 (0.00046) (0.00029)	0.00438 (0.00043) (0.00013)
	μ=0.6, σ=1	0.01996 (0.00306) (0.00201)	0.02798 (0.00293) (0.00087)	0.05014 (0.00545) (0.00376)	0.04545 (0.00505) (0.0031)	0.03854 (0.00508) (0.00323)	0.03342 (0.00498) (0.00226)	–	0.02044 (0.00321) (0.00158)	0.02253 (0.00279) (0.00275)	0.0474 (0.00555) (0.00443)	0.02004 (0.00216) (0.00192)	0.02378 (0.00271) (0.0013)
	μ=1.5, σ=0.8	0.01764 (0.0023) (0.00129)	0.01807 (0.00205) (0.00154)	0.04911 (0.0067) (0.00276)	0.0267 (0.00231) (0.0012)	0.02183 (0.00208) (0.00181)	0.017 (0.00149) (0.00115)	–	0.01356 (0.00173) (0.00147)	0.01864 (0.002) (0.00117)	0.04069 (0.00923) (0.00529)	0.01419 (0.00114) (0.00068)	0.01419 (0.00129) (0.00055)

TABLES OBTAINED FOR THE ATKINSON MEASURE FOR n=15 000

Table A49: Mean – true Atkinson (Atkinson, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02812 (0.00012)	0.03015 (0.00015)	-0.00828 (0.00009)	0.01289 (0.00019)	0.02351 (0.01661)	–	–	0.01857 (0.00008)	0.02994 (0.00014)	0.0001 (0.00014)	-0.02318 (0.00008)	0.00002 (0.00009)
	α=1.5	-0.1859 (0.00058)	-0.17891 (0.00077)	-0.06234 (0.00196)	-0.17619 (0.00191)	-0.17338 (0.00255)	-0.16891 (0.38839)	-0.15504 (0.3634)	-0.19509 (0.00055)	-0.13198 (0.0005)	-0.00532 (0.0027)	-0.2083 (0.00053)	-0.00548 (0.00488)
	α=1.1	-0.38898 (0.00264)	-0.36954 (0.0028)	-0.10991 (0.00736)	-0.27437 (0.01061)	-0.28054 (0.01135)	-0.28555 (0.01168)	-0.28643 (0.00887)	-0.45567 (0.00201)	-0.35213 (0.0027)	-0.17322 (0.00563)	-0.50169 (0.00216)	-0.13977 (0.00918)
Burr	c=1.2, k=2.5	-0.20765 (0.00017)	-0.20543 (0.00023)	-0.22953 (0.00013)	-0.21546 (0.00022)	-0.21078 (0.02105)	–	–	-0.21452 (0.00015)	-0.20682 (0.00023)	-0.22103 (0.00016)	-0.23654 (0.00012)	1.73E-06 (0.00028)

	c=3, k=1	-0.0281 (0.0002)	-0.02598 (0.00018)	-0.06269 (0.00013)	-0.04235 (0.00022)	-0.03274 (0.01724)	–	–	-0.03755 (0.00012)	-0.02642 (0.00022)	-0.05433 (0.00015)	-0.07667 (0.00011)	-0.00001 (0.00017)
	c=1.3, k=1.5	-0.17493 (0.00024)	-0.15148 (0.00042)	-0.20533 (0.00053)	-0.15294 (0.00089)	-0.15421 (0.00091)	–	–	-0.18825 (0.00028)	-0.17279 (0.00025)	-0.19505 (0.00126)	-0.24552 (0.00024)	-0.00074 (0.00139)
	c=3, k=1.1/3	-0.48376 (0.00016)	-0.41192 (0.00026)	-0.1117 (0.00245)	-0.05888 (0.00314)	-0.089 (0.00403)	–	–	-0.47074 (0.00027)	-0.4798 (0.00015)	-0.16654 (0.00783)	-0.52127 (0.00039)	-0.14127 (0.00971)
Lognormal	μ=0.5, σ=0.5	0.02625 (0.00004)	0.02684 (0.00007)	-0.0154 (0.00005)	0.00649 (0.00011)	0.0233 (0.00005)	0.02594 (0.00529)	–	0.0236 (0.00004)	0.03008 (0.00004)	-0.00819 (0.00007)	-0.03644 (0.00003)	0.00003 (0.00007)
	μ=0.6, σ=1	-0.04714 (0.00036)	-0.01235 (0.00049)	-0.02039 (0.00069)	-0.01876 (0.00066)	-0.0108 (0.0006)	-0.00054 (0.00052)	–	-0.09015 (0.00039)	-0.00597 (0.00037)	-0.03757 (0.00201)	-0.03479 (0.00046)	-8.55E-07 (0.0004)
	μ=1.5, σ=0.8	0.02008 (0.00012)	0.02567 (0.00018)	0.02095 (0.00034)	-0.00543 (0.00023)	-0.00951 (0.00022)	-0.00868 (0.0002)	–	-0.00284 (0.00013)	0.02788 (0.00013)	0.03167 (0.00148)	-0.03206 (0.00016)	0.00002 (0.00012)

Table A50: Median – true Atkinson (Atkinson, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.0282 (0.00016)	0.03011 (0.00019)	-0.0083 (0.00013)	0.01284 (0.00029)	0.02338 (0.02384)	–	–	0.01858 (0.00013)	0.02997 (0.00026)	-0.00001 (0.00018)	-0.02319 (0.00014)	-0.00011 (0.00017)
	α=1.5	-0.18626 (0.00128)	-0.17987 (0.00111)	-0.06626 (0.00182)	-0.18047 (0.00164)	-0.17917 (0.00181)	-0.17785 (0.18722)	-0.17562 (0.21099)	-0.1958 (0.00115)	-0.13233 (0.00094)	-0.01436 (0.00178)	-0.20898 (0.00078)	-0.01405 (0.0042)
	α=1.1	-0.38751 (0.00284)	-0.36897 (0.00298)	-0.11551 (0.00577)	-0.29313 (0.00618)	-0.30443 (0.00516)	-0.31271 (0.00596)	-0.32103 (0.00479)	-0.45529 (0.00237)	-0.35101 (0.00368)	-0.18858 (0.00336)	-0.50103 (0.0028)	-0.16634 (0.00953)
Burr	c=1.2, k=2.5	-0.20759 (0.00024)	-0.20561 (0.00032)	-0.22968 (0.00022)	-0.21571 (0.00042)	-0.21097 (0.0376)	–	–	-0.21457 (0.0002)	-0.20683 (0.00036)	-0.22126 (0.00019)	-0.23663 (0.00016)	0.00004 (0.00051)
	c=3, k=1	-0.02809 (0.00022)	-0.02607 (0.00013)	-0.06277 (0.00015)	-0.0425 (0.00033)	-0.03296 (0.01738)	–	–	-0.03758 (0.00012)	-0.02643 (0.00025)	-0.05441 (0.00026)	-0.07675 (0.00018)	-0.00023 (0.00027)
	c=1.3, k=1.5	-0.17484 (0.00031)	-0.15155 (0.00056)	-0.20552 (0.0005)	-0.15316 (0.00083)	-0.15469 (0.00069)	–	–	-0.18828 (0.00038)	-0.17279 (0.00028)	-0.19771 (0.00117)	-0.24563 (0.00032)	-0.00292 (0.00105)
	c=3, k=1.1/3	-0.48374 (0.00022)	-0.41197 (0.00041)	-0.11196 (0.00358)	-0.06136 (0.0048)	-0.09342 (0.00582)	–	–	-0.47076 (0.00046)	-0.47982 (0.00022)	-0.17902 (0.01057)	-0.5213 (0.00048)	-0.15948 (0.01066)
Lognormal	μ=0.5, σ=0.5	0.02629 (0.00005)	0.02681 (0.00006)	-0.01537 (0.00007)	0.00648 (0.00014)	0.02329 (0.00005)	0.02595 (0.00613)	–	0.02361 (0.00004)	0.03008 (0.00005)	-0.00819 (0.00008)	-0.03643 (0.00006)	0.00003 (0.00013)
	μ=0.6, σ=1	-0.04741 (0.00048)	-0.01266 (0.00068)	-0.02065 (0.00095)	-0.01899 (0.00087)	-0.01086 (0.00079)	-0.00063 (0.00066)	–	-0.09018 (0.00052)	-0.00603 (0.00044)	-0.04083 (0.0015)	-0.03486 (0.00055)	-0.00013 (0.00051)
	μ=1.5, σ=0.8	0.01997 (0.00024)	0.02557 (0.00021)	0.02062 (0.00039)	-0.00557 (0.00027)	-0.00963 (0.00026)	-0.00878 (0.00022)	–	-0.00292 (0.00021)	0.02776 (0.00022)	0.02908 (0.00066)	-0.03216 (0.00026)	-0.00008 (0.00019)

Table A51: RMSE (Atkinson, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02818 (0.00012)	0.03025 (0.00014)	0.00842 (0.0001)	0.01315 (0.00018)	6.31471 (0.0164)	–	–	0.01862 (0.00008)	0.03001 (0.00013)	0.00197 (0.00023)	0.02321 (0.00008)	0.00171 (0.00011)
	α=1.5	0.18604 (0.00058)	0.17916 (0.00075)	0.0665 (0.00158)	0.17728 (0.00155)	0.17492 (0.00174)	6.27053 (0.26387)	6.0855 (0.26759)	0.19521 (0.00054)	0.13216 (0.00049)	0.03952 (0.00939)	0.20841 (0.00052)	0.04067 (0.01232)
	α=1.1	0.38984 (0.0027)	0.37066 (0.00283)	0.14028 (0.00403)	0.29747 (0.00708)	0.30504 (0.00725)	0.30939 (0.00751)	0.30842 (0.00599)	0.45627 (0.00203)	0.35303 (0.00276)	0.18369 (0.00432)	0.50226 (0.00217)	0.17171 (0.00698)

Burr	c=1.2, k=2.5	0.20766 (0.00017)	0.20545 (0.00023)	0.22953 (0.00013)	0.21548 (0.00022)	3.65509 (0.02146)	–	–	0.21453 (0.00015)	0.20683 (0.00023)	0.22104 (0.00016)	0.23655 (0.00012)	0.00419 (0.00022)
	c=3, k=1	0.02817 (0.0002)	0.0261 (0.00017)	0.0627 (0.00013)	0.04243 (0.00022)	5.94745 (0.01752)	–	–	0.03758 (0.00012)	0.02651 (0.00021)	0.05436 (0.00014)	0.07668 (0.00011)	0.00238 (0.00023)
	c=1.3, k=1.5	0.17501 (0.00024)	0.15162 (0.00042)	0.20547 (0.00052)	0.15329 (0.00087)	0.15455 (0.00088)	–	–	0.18834 (0.00028)	0.17287 (0.00025)	0.19557 (0.00118)	0.24559 (0.00024)	0.01388 (0.00237)
	c=3, k=1.1/3	0.48371 (0.00016)	0.41188 (0.00027)	0.11469 (0.00246)	0.06842 (0.00317)	0.09933 (0.00373)	–	–	0.4707 (0.00027)	0.47975 (0.00015)	0.1832 (0.00645)	0.52124 (0.00039)	0.16582 (0.00545)
Lognormal	μ=0.5, σ=0.5	0.02626 (0.00004)	0.02685 (0.00007)	0.01542 (0.00005)	0.00661 (0.0001)	0.0233 (0.00005)	8.59265 (0.00529)	–	0.0236 (0.00004)	0.03009 (0.00004)	0.00822 (0.00007)	0.03645 (0.00003)	0.00075 (0.00006)
	μ=0.6, σ=1	0.01109 (0.00037)	0.00361 (0.00024)	0.04744 (0.00069)	0.01331 (0.00064)	0.02087 (0.00059)	0.01916 (0.00051)	–	0.0349 (0.00039)	0.0065 (0.00037)	0.04181 (0.00304)	0.0902 (0.00046)	0.00317 (0.00027)
	μ=1.5, σ=0.8	0.02017 (0.00012)	0.02579 (0.00018)	0.02133 (0.00034)	0.00622 (0.00023)	0.00996 (0.00022)	0.00906 (0.0002)	–	0.00347 (0.00014)	0.02794 (0.00012)	0.03441 (0.00438)	0.03213 (0.00015)	0.00204 (0.00016)

Table A52: MAD (Atkinson, n=15 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00129 (0.00013) (0.00013)	0.0016 (0.00021) (0.00006)	0.00105 (0.00015) (0.00006)	0.00183 (0.00027) (0.0002)	0.15088 (0.02159) (0.01187)	–	–	0.00096 (0.00006) (0.00003)	0.00136 (0.00017) (0.00011)	0.00122 (0.00018) (0.00014)	0.00077 (0.0001) (0.00006)	0.00107 (0.00011) (0.00007)
	α=1.5	0.00486 (0.00055) (0.0004)	0.00667 (0.00105) (0.00082)	0.01454 (0.00238) (0.00211)	0.0108 (0.00148) (0.00096)	0.01117 (0.00141) (0.00122)	1.1039 (0.12161) (0.04211)	1.16922 (0.27188) (0.167)	0.00461 (0.00069) (0.00061)	0.00467 (0.00065) (0.00037)	0.01434 (0.00228) (0.00123)	0.00467 (0.00053) (0.00019)	0.01562 (0.00197) (0.00109)
	α=1.1	0.01641 (0.0027) (0.00141)	0.0193 (0.00121) (0.00089)	0.07385 (0.00942) (0.00402)	0.0839 (0.00979) (0.00549)	0.08224 (0.00949) (0.00579)	0.07825 (0.0104) (0.0057)	0.07168 (0.00701) (0.00235)	0.01597 (0.00119) (0.00114)	0.0161 (0.00268) (0.00159)	0.03078 (0.00454) (0.00196)	0.01642 (0.00202) (0.00068)	0.05103 (0.00786) (0.00536)
Burr	c=1.2, k=2.5	0.00153 (0.00018) (0.0001)	0.00197 (0.0002) (0.00013)	0.00117 (0.00015) (0.00009)	0.00209 (0.00021) (0.00012)	0.19658 (0.01517) (0.01067)	–	–	0.00127 (0.00016) (0.00011)	0.00162 (0.00016) (0.00004)	0.00131 (0.0001) (0.00005)	0.00109 (0.00008) (0.00005)	0.00285 (0.0002) (0.00011)
	c=3, k=1	0.00133 (0.00016) (0.00011)	0.00159 (0.00015) (0.00012)	0.001 (0.00008) (0.00006)	0.0017 (0.0002) (0.00013)	0.14902 (0.01787) (0.0142)	–	–	0.00098 (0.00012) (0.00009)	0.00144 (0.00013) (0.00004)	0.00115 (0.00017) (0.0001)	0.00078 (0.00005) (0.00004)	0.00145 (0.00013) (0.00008)
	c=1.3, k=1.5	0.00203 (0.00029) (0.00008)	0.0033 (0.00046) (0.00036)	0.004 (0.00039) (0.00027)	0.00632 (0.00059) (0.00048)	0.00618 (0.00058) (0.00053)	–	–	0.00239 (0.00035) (0.00018)	0.00214 (0.00027) (0.00015)	0.00634 (0.00062) (0.00029)	0.00195 (0.00023) (0.00012)	0.00668 (0.00088) (0.00053)
	c=3, k=1.1/3	0.00114 (0.00011) (0.00006)	0.00206 (0.00023) (0.00011)	0.01736 (0.00278) (0.00181)	0.02399 (0.00261) (0.0018)	0.02918 (0.00333) (0.0025)	–	–	0.00209 (0.00027) (0.00006)	0.00119 (0.00012) (0.00008)	0.04851 (0.00549) (0.00231)	0.00253 (0.00034) (0.00013)	0.04639 (0.00673) (0.00294)
Lognormal	μ=0.5, σ=0.5	0.00042 (0.00005) (0.00004)	0.00048 (0.00005) (0.00006)	0.00044 (0.00007) (0.00006)	0.00081 (0.00008) (0.00007)	0.00043 (0.00005) (0.00003)	0.03965 (0.00557) (0.00333)	–	0.00034 (0.00004) (0.00003)	0.00046 (0.00004) (0.00002)	0.00048 (0.00004) (0.00003)	0.00036 (0.00004) (0.00004)	0.00052 (0.00006) (0.00004)
	μ=0.6, σ=1	0.0017 (0.00019) (0.00015)	0.00233 (0.00027) (0.00013)	0.00345 (0.00039) (0.00021)	0.00329 (0.00039) (0.00015)	0.00293 (0.00034) (0.00012)	0.00253 (0.00023) (0.00015)	–	0.00187 (0.00021) (0.00012)	0.0017 (0.00022) (0.00017)	0.00535 (0.00074) (0.00047)	0.00202 (0.00027) (0.00019)	0.00203 (0.00022) (0.00017)
	μ=1.5, σ=0.8	0.00123 (0.00018)	0.00173 (0.00016)	0.00266 (0.0002)	0.002 (0.00014)	0.00195 (0.00014)	0.00175 (0.00015)	–	0.00133 (0.00016)	0.00129 (0.00017)	0.00402 (0.00054)	0.00136 (0.00012)	0.00139 (0.00015)

		(0.00012)	(0.00009)	(0.00013)	(0.00006)	(0.00009)	(0.00009)		(0.00013)	(0.00011)	(0.00029)	(0.00009)	(0.00007)

TABLES OBTAINED FOR THE ATKINSON MEASURE FOR n=10 000

Table A53: Mean – true Atkinson (Atkinson, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02793 (0.00021)	0.03031 (0.00022)	-0.00822 (0.00015)	0.01302 (0.00025)	0.02379 (0.02502)	–	–	0.01856 (0.00014)	0.02974 (0.00025)	0.00016 (0.00015)	-0.02321 (0.00011)	0.00014 (0.00025)
	α=1.5	-0.18685 (0.00077)	-0.1798 (0.00094)	-0.06388 (0.00208)	-0.17749 (0.00176)	-0.17492 (0.00202)	-0.17013 (0.21015)	-0.15618 (0.2225)	-0.1958 (0.0007)	-0.13275 (0.00074)	-0.00941 (0.00358)	-0.20907 (0.00077)	-0.00795 (0.00418)
	α=1.1	-0.39697 (0.00397)	-0.37674 (0.00445)	-0.11376 (0.01229)	-0.27989 (0.01627)	-0.28693 (0.01714)	-0.29079 (0.01689)	-0.282 (0.01942)	-0.46094 (0.00404)	-0.35939 (0.00373)	-0.18023 (0.01182)	-0.5064 (0.00411)	-0.15236 (0.01052)
Burr	c=1.2, k=2.5	-0.20811 (0.0004)	-0.20534 (0.00043)	-0.22955 (0.00026)	-0.21531 (0.00043)	-0.2104 (0.04486)	–	–	-0.21467 (0.00031)	-0.20729 (0.00036)	-0.22101 (0.00051)	-0.23666 (0.00024)	-0.00017 (0.00072)
	c=3, k=1	-0.02821 (0.00029)	-0.02577 (0.00037)	-0.06259 (0.00023)	-0.04216 (0.00038)	-0.03246 (0.02937)	–	–	-0.03752 (0.00021)	-0.02655 (0.00029)	-0.0542 (0.00031)	-0.07665 (0.00015)	0.00004 (0.00033)
	c=1.3, k=1.5	-0.17491 (0.00043)	-0.15132 (0.00051)	-0.20502 (0.00052)	-0.15228 (0.00075)	-0.15346 (0.00063)	–	–	-0.18816 (0.00043)	-0.17275 (0.00043)	-0.19489 (0.00189)	-0.24543 (0.00036)	-0.00015 (0.00195)
	c=3, k=1.1/3	-0.48379 (0.0002)	-0.41191 (0.0003)	-0.11042 (0.00214)	-0.05734 (0.0035)	-0.08632 (0.00442)	–	–	-0.47073 (0.00032)	-0.4798 (0.00022)	-0.17173 (0.00562)	-0.52107 (0.00041)	-0.13446 (0.00859)
Lognormal	μ=0.5, σ=0.5	0.02626 (0.0001)	0.02699 (0.0001)	-0.01539 (0.0001)	0.00673 (0.00016)	0.02336 (0.0001)	0.02603 (0.01116)	–	0.0236 (0.00008)	0.03009 (0.00009)	-0.00812 (0.0001)	-0.03643 (0.00009)	0.00002 (0.00013)
	μ=0.6, σ=1	-0.01082 (0.00023)	-0.0004 (0.00034)	-0.04676 (0.00056)	-0.01202 (0.0005)	-0.02012 (0.00044)	-0.0185 (0.00038)	–	-0.03476 (0.00025)	-0.00595 (0.00028)	-0.03829 (0.00244)	-0.09011 (0.00028)	-0.00013 (0.00037)
	μ=1.5, σ=0.8	0.02007 (0.00014)	0.02557 (0.0002)	0.02078 (0.00036)	-0.00555 (0.00025)	-0.00965 (0.00024)	-0.00881 (0.00021)	–	-0.00288 (0.00016)	0.02788 (0.00021)	0.03132 (0.00152)	-0.03205 (0.0002)	-0.00002 (0.00015)

Table A54: Median – true Atkinson (Atkinson, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02804 (0.00025)	0.03028 (0.00035)	-0.0083 (0.00025)	0.01286 (0.00041)	0.02364 (0.03651)	–	–	0.01856 (0.00018)	0.0298 (0.0003)	-0.00002 (0.00028)	-0.0232 (0.00023)	-0.00007 (0.00023)
	α=1.5	-0.18709 (0.0013)	-0.1804 (0.00101)	-0.06668 (0.00205)	-0.18111 (0.00166)	-0.17946 (0.00192)	-0.17782 (0.21406)	-0.16867 (0.14958)	-0.19632 (0.00106)	-0.13279 (0.00093)	-0.01721 (0.00318)	-0.20937 (0.00094)	-0.01703 (0.00355)
	α=1.1	-0.39326 (0.00535)	-0.3752 (0.00786)	-0.14232 (0.01884)	-0.32489 (0.02109)	-0.33888 (0.01844)	-0.34879 (0.01891)	-0.35191 (0.04042)	-0.46026 (0.00692)	-0.35794 (0.00457)	-0.19244 (0.00973)	-0.50613 (0.00602)	-0.1749 (0.01277)
Burr	c=1.2, k=2.5	-0.20801 (0.00041)	-0.20567 (0.00043)	-0.22984 (0.00022)	-0.21572 (0.0004)	-0.21089 (0.02742)	–	–	-0.21485 (0.00032)	-0.2072 (0.00033)	-0.22142 (0.00027)	-0.23689 (0.00022)	-0.0005 (0.00075)
	c=3, k=1	-0.02815 (0.00029)	-0.02595 (0.00039)	-0.06275 (0.00031)	-0.04238 (0.00051)	-0.03278 (0.03419)	–	–	-0.03758 (0.00022)	-0.02653 (0.00035)	-0.05444 (0.00032)	-0.07675 (0.00014)	-0.00015 (0.00035)
	c=1.3, k=1.5	-0.17511 (0.00048)	-0.1515 (0.00059)	-0.2054 (0.00055)	-0.15302 (0.00079)	-0.15396 (0.00089)	–	–	-0.18835 (0.00052)	-0.17291 (0.00055)	-0.19719 (0.00137)	-0.24555 (0.00044)	-0.00211 (0.00141)
	c=3, k=1.1/3	-0.48379 (0.00023)	-0.41194 (0.00043)	-0.11048 (0.0031)	-0.06019 (0.00421)	-0.09197 (0.00526)	–	–	-0.47074 (0.00046)	-0.47978 (0.00026)	-0.18609 (0.00851)	-0.52104 (0.00045)	-0.1579 (0.00928)

Lognormal	μ=0.5, σ=0.5	0.02625 (0.00012)	0.02697 (0.00013)	-0.0154 (0.00012)	0.0067 (0.0002)	0.02339 (0.00011)	0.02601 (0.01598)	–	0.0236 (0.0001)	0.03006 (0.00012)	-0.00815 (0.00015)	-0.03643 (0.0001)	0.00005 (0.00014)
	μ=0.6, σ=1	-0.01069 (0.00023)	-0.00014 (0.00036)	-0.0466 (0.00066)	-0.01181 (0.00055)	-0.01987 (0.00045)	-0.01832 (0.00039)	–	-0.03467 (0.00027)	-0.00582 (0.00033)	-0.0413 (0.00149)	-0.09 (0.00036)	-0.00024 (0.00042)
	μ=1.5, σ=0.8	0.01996 (0.00024)	0.02546 (0.00032)	0.02045 (0.00047)	-0.0057 (0.00027)	-0.00974 (0.00029)	-0.00895 (0.00025)	–	-0.00296 (0.00029)	0.02778 (0.00036)	0.02826 (0.00079)	-0.03215 (0.00023)	-0.0001 (0.00024)

Table A55: RMSE (Atkinson, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02804 (0.0002)	0.03044 (0.00022)	0.00842 (0.00014)	0.01336 (0.00024)	6.34458 (0.02486)	–	–	0.01863 (0.00014)	0.02985 (0.00024)	0.00215 (0.00014)	0.02325 (0.00011)	0.00247 (0.0006)
	α=1.5	0.18705 (0.00076)	0.18011 (0.00092)	0.06835 (0.00169)	0.17858 (0.00159)	0.1764 (0.0017)	6.29337 (0.16022)	6.30212 (0.13708)	0.19597 (0.00069)	0.13301 (0.00073)	0.0411 (0.01367)	0.20923 (0.00076)	0.04199 (0.00891)
	α=1.1	0.39838 (0.00401)	0.37839 (0.00449)	0.14471 (0.00928)	0.30361 (0.01403)	0.31207 (0.01469)	0.31685 (0.01441)	0.30197 (0.01579)	0.46183 (0.00406)	0.36086 (0.00379)	0.1919 (0.0086)	0.50722 (0.00412)	0.18135 (0.00718)
Burr	c=1.2, k=2.5	0.20813 (0.0004)	0.20537 (0.00043)	0.22956 (0.00026)	0.21533 (0.00043)	3.69832 (0.0465)	–	–	0.21469 (0.00031)	0.20731 (0.00036)	0.22104 (0.00046)	0.23667 (0.00024)	0.00517 (0.00045)
	c=3, k=1	0.02832 (0.00028)	0.02593 (0.00036)	0.06262 (0.00022)	0.04227 (0.00037)	5.97767 (0.02967)	–	–	0.03756 (0.00021)	0.02668 (0.00028)	0.05426 (0.00029)	0.07666 (0.00015)	0.00296 (0.00083)
	c=1.3, k=1.5	0.175 (0.00044)	0.15151 (0.00051)	0.20521 (0.00052)	0.15283 (0.00076)	0.154 (0.00064)	–	–	0.18827 (0.00043)	0.17285 (0.00043)	0.19558 (0.00149)	0.24551 (0.00036)	0.01739 (0.00853)
	c=3, k=1.1/3	0.48374 (0.0002)	0.41188 (0.0003)	0.11495 (0.00182)	0.07152 (0.00276)	0.10159 (0.00297)	–	–	0.47069 (0.00032)	0.47976 (0.00022)	0.19138 (0.00479)	0.52104 (0.00041)	0.16678 (0.00701)
Lognormal	μ=0.5, σ=0.5	0.02627 (0.0001)	0.02701 (0.0001)	0.01542 (0.0001)	0.00688 (0.00016)	0.02338 (0.0001)	8.60108 (0.01115)	–	0.02361 (0.00008)	0.0301 (0.00009)	0.00819 (0.0001)	0.03643 (0.00009)	0.0009 (0.00004)
	μ=0.6, σ=1	0.01121 (0.00021)	0.00435 (0.00031)	0.04722 (0.00055)	0.01351 (0.00048)	0.02085 (0.00043)	0.01912 (0.00038)	–	0.03491 (0.00024)	0.0067 (0.00025)	0.04212 (0.00253)	0.09018 (0.00027)	0.00394 (0.00023)
	μ=1.5, σ=0.8	0.02019 (0.00014)	0.02574 (0.0002)	0.0213 (0.00038)	0.00658 (0.00024)	0.01024 (0.00023)	0.00931 (0.00021)	–	0.00371 (0.00017)	0.02797 (0.00022)	0.03441 (0.00395)	0.03215 (0.0002)	0.00247 (0.00017)

Table A56: MAD (Atkinson, n=10 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00157 (0.00022) (0.00017)	0.00187 (0.00017) (0.00003)	0.00122 (0.00012) (0.00008)	0.00211 (0.00026) (0.00018)	0.17916 (0.02213) (0.01085)	–	–	0.00112 (0.00011) (0.00007)	0.00172 (0.00018) (0.0001)	0.00135 (0.00009) (0.00006)	0.00097 (0.0001) (0.00003)	0.00136 (0.00016) (0.00009)
	α=1.5	0.00633 (0.00082) (0.00071)	0.0073 (0.00055) (0.0002)	0.01485 (0.00168) (0.0011)	0.0107 (0.00117) (0.00082)	0.01099 (0.00101) (0.00069)	1.15863 (0.13854) (0.08953)	1.48468 (0.3668) (0.1624)	0.00563 (0.00056) (0.00051)	0.00568 (0.00067) (0.00049)	0.01449 (0.0021) (0.00109)	0.00558 (0.00057) (0.00041)	0.01745 (0.0022) (0.00153)
	α=1.1	0.02505 (0.00285) (0.00188)	0.02611 (0.00198) (0.00146)	0.0566 (0.01525) (0.00861)	0.06542 (0.02197) (0.01016)	0.06297 (0.02634) (0.01131)	0.05351 (0.0263) (0.0134)	0.03511 (0.03168) (0.00533)	0.02104 (0.00211) (0.00165)	0.0255 (0.00312) (0.00243)	0.03374 (0.00648) (0.0038)	0.02087 (0.00234) (0.0022)	0.04908 (0.00744) (0.00539)
Burr	c=1.2, k=2.5	0.00206 (0.00025) (0.00016)	0.00223 (0.00026) (0.00017)	0.00138 (0.00018) (0.00013)	0.00231 (0.00027) (0.0002)	0.22324 (0.03627) (0.02364)	–	–	0.00159 (0.00021) (0.0001)	0.00215 (0.00027) (0.00018)	0.00153 (0.00013) (0.00011)	0.00136 (0.00023) (0.00015)	0.00332 (0.00036) (0.0003)

	c=3, k=1	0.0017 (0.00015) (0.00011)	0.00202 (0.00021) (0.00011)	0.00128 (0.00013) (0.00007)	0.0021 (0.00024) (0.00016)	0.18965 (0.02258) (0.00931)	–	–	0.00121 (0.00011) (0.00003)	0.00175 (0.00014) (0.00007)	0.00141 (0.00012) (0.00007)	0.00102 (0.00009) (0.00008)	0.00175 (0.00016) (0.00011)
	c=1.3, k=1.5	0.00277 (0.00019) (0.00009)	0.0042 (0.0005) (0.00027)	0.00507 (0.00067) (0.00054)	0.00768 (0.00116) (0.00087)	0.0075 (0.00106) (0.00082)	–	–	0.00305 (0.00023) (0.00012)	0.00272 (0.00027) (0.00023)	0.00804 (0.00089) (0.00037)	0.0026 (0.00031) (0.0002)	0.00801 (0.00101) (0.00036)
	c=3, k=1.1/3	0.00136 (0.00015) (0.00012)	0.00234 (0.00029) (0.00018)	0.02146 (0.00209) (0.00146)	0.02801 (0.00319) (0.00237)	0.03427 (0.00469) (0.00437)	–	–	0.00238 (0.00027) (0.00018)	0.00139 (0.00017) (0.00006)	0.0511 (0.00731) (0.0067)	0.00301 (0.00037) (0.00031)	0.05162 (0.00534) (0.00428)
Lognormal	μ=0.5, σ=0.5	0.00056 (0.00005) (0.00003)	0.00066 (0.00009) (0.00004)	0.00059 (0.00004) (0.00002)	0.00098 (0.0001) (0.00006)	0.00056 (0.00007) (0.00005)	0.05249 (0.00564) (0.00457)	–	0.00046 (0.00005) (0.00003)	0.00061 (0.00007) (0.00005)	0.00068 (0.00007) (0.00005)	0.00045 (0.00004) (0.00002)	0.00064 (0.00005) (0.00004)
	μ=0.6, σ=1	0.00195 (0.00024) (0.00015)	0.0028 (0.00047) (0.00026)	0.00417 (0.00064) (0.00037)	0.0039 (0.00075) (0.00049)	0.00348 (0.00068) (0.00039)	0.00307 (0.00059) (0.00042)	–	0.00216 (0.00027) (0.00018)	0.00212 (0.00022) (0.00017)	0.00632 (0.00062) (0.00034)	0.0024 (0.00029) (0.00021)	0.00273 (0.0002) (0.00014)
	μ=1.5, σ=0.8	0.00146 (0.00015) (0.00005)	0.00194 (0.00026) (0.00017)	0.00303 (0.0004) (0.00017)	0.00224 (0.00029) (0.00023)	0.00219 (0.00028) (0.00019)	0.00195 (0.00024) (0.00011)	–	0.00156 (0.00016) (0.00009)	0.00148 (0.00017) (0.00007)	0.00418 (0.00052) (0.00028)	0.00171 (0.00017) (0.00013)	0.00163 (0.00014) (0.00004)

TABLES OBTAINED FOR THE ATKINSON MEASURE FOR n=5 000

Table A57: Mean – true Atkinson (Atkinson, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02739 (0.00035)	0.0309 (0.00027)	-0.00802 (0.0002)	0.01345 (0.00035)	0.025 (0.03151)	–	–	0.01861 (0.00021)	0.02923 (0.00038)	0.00039 (0.00037)	-0.02328 (0.00018)	-0.00004 (0.00028)
	α=1.5	-0.18834 (0.00121)	-0.17966 (0.00155)	-0.0596 (0.00361)	-0.17288 (0.00361)	-0.16833 (0.24468)	-0.15828 (0.32224)	-0.14354 (0.61895)	-0.19625 (0.00115)	-0.13429 (0.0011)	-0.00693 (0.00559)	-0.20963 (0.00132)	-0.0083 (0.00293)
	α=1.1	-0.41059 (0.00345)	-0.38653 (0.00375)	-0.09192 (0.01037)	-0.24939 (0.01183)	-0.25643 (0.01134)	-0.25796 (0.01277)	-0.18118 (0.00795)	-0.46797 (0.00384)	-0.37282 (0.00382)	-0.18124 (0.00804)	-0.5126 (0.00412)	-0.16543 (0.0091)
Burr	c=1.2, k=2.5	-0.20846 (0.00131)	-0.20441 (0.00122)	-0.22913 (0.00078)	-0.21419 (0.00133)	-0.20852 (0.44832)	–	–	-0.21448 (0.00088)	-0.20767 (0.00124)	-0.22057 (0.00133)	-0.23656 (0.00064)	-0.00011 (0.00314)
	c=3, k=1	-0.02874 (0.00034)	-0.02503 (0.00031)	-0.06229 (0.00023)	-0.04148 (0.00035)	-0.03105 (0.04658)	–	–	-0.03742 (0.00019)	-0.02711 (0.00031)	-0.05384 (0.00039)	-0.07664 (0.00019)	-0.00005 (0.0005)
	c=1.3, k=1.5	-0.17507 (0.00059)	-0.15176 (0.00113)	-0.20541 (0.00139)	-0.15272 (0.00227)	-0.15363 (0.00221)	–	–	-0.18839 (0.00075)	-0.17292 (0.00062)	-0.1958 (0.00292)	-0.2457 (0.00066)	-0.00122 (0.00249)
	c=3, k=1.1/3	-0.48399 (0.00024)	-0.41221 (0.00037)	-0.11236 (0.00394)	-0.05848 (0.00475)	-0.08444 (0.00535)	–	–	-0.47104 (0.00039)	-0.48009 (0.00023)	-0.19198 (0.00954)	-0.5216 (0.00057)	-0.15379 (0.00943)
Lognormal	μ=0.5, σ=0.5	0.02619 (0.00013)	0.02739 (0.00014)	-0.01539 (0.00013)	0.0072 (0.0002)	0.02346 (0.01246)	0.02624 (0.02019)	–	0.02358 (0.0001)	0.03003 (0.00013)	-0.00804 (0.00011)	-0.03648 (0.00011)	-0.00005 (0.00013)
	μ=0.6, σ=1	-0.01091 (0.00035)	-0.00053 (0.00052)	-0.0468 (0.0008)	-0.01207 (0.00074)	-0.02017 (0.00066)	-0.0185 (0.00057)	–	-0.03485 (0.00039)	-0.006 (0.00036)	-0.03882 (0.00174)	-0.09024 (0.00046)	0.00003 (0.00037)
	μ=1.5, σ=0.8	0.01988 (0.00031)	0.02538 (0.00043)	0.02078 (0.00064)	-0.00568 (0.00046)	-0.00983 (0.00045)	-0.00897 (0.00042)	–	-0.00306 (0.00033)	0.0277 (0.00035)	0.03112 (0.00164)	-0.03224 (0.00031)	-0.00018 (0.00038)

Table A58: Median – true Atkinson (Atkinson, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02735 (0.00052)	0.03066 (0.00039)	-0.00816 (0.0003)	0.01313 (0.00037)	0.02453 (0.04792)	–	–	0.01845 (0.00022)	0.02915 (0.00057)	0.00004 (0.0003)	-0.02344 (0.00028)	-0.00037 (0.0003)
	α=1.5	-0.18956 (0.00161)	-0.18103 (0.0016)	-0.06303 (0.00384)	-0.17664 (0.00375)	-0.17401 (0.30927)	-0.16926 (0.31627)	-0.16136 (0.1093)	-0.19767 (0.00127)	-0.13537 (0.00137)	-0.0173 (0.00444)	-0.21088 (0.00139)	-0.01961 (0.00314)
	α=1.1	-0.41423 (0.00639)	-0.38889 (0.00777)	-0.05898 (0.01756)	-0.22475 (0.02371)	-0.23739 (0.02338)	-0.2386 (0.01309)	-0.20369 (0.00565)	-0.46942 (0.00698)	-0.37655 (0.00635)	-0.20055 (0.00957)	-0.51273 (0.00725)	-0.19059 (0.0079)
Burr	c=1.2, k=2.5	-0.20846 (0.00187)	-0.20466 (0.00172)	-0.22935 (0.00097)	-0.21463 (0.00158)	-0.20923 (0.00548)	–	–	-0.21469 (0.0011)	-0.20772 (0.00219)	-0.22096 (0.00156)	-0.23662 (0.00104)	-0.00078 (0.00399)
	c=3, k=1	-0.02886 (0.00046)	-0.02528 (0.00026)	-0.06248 (0.00022)	-0.04184 (0.00035)	-0.03155 (0.03373)	–	–	-0.03754 (0.00026)	-0.02718 (0.00035)	-0.05435 (0.00039)	-0.0768 (0.00026)	-0.00034 (0.0004)
	c=1.3, k=1.5	-0.17512 (0.0006)	-0.15183 (0.00118)	-0.20613 (0.00145)	-0.15442 (0.00245)	-0.15538 (0.00251)	–	–	-0.18842 (0.00078)	-0.17292 (0.00059)	-0.2007 (0.00216)	-0.24595 (0.00091)	-0.00564 (0.00281)
	c=3, k=1.1/3	-0.48393 (0.00041)	-0.41205 (0.00055)	-0.11346 (0.00487)	-0.06226 (0.00629)	-0.09542 (0.00722)	–	–	-0.47089 (0.00062)	-0.48005 (0.00039)	-0.21124 (0.01219)	-0.52134 (0.00076)	-0.18364 (0.00866)
Lognormal	μ=0.5, σ=0.5	0.02614 (0.00012)	0.02729 (0.00016)	-0.01547 (0.00012)	0.00715 (0.00023)	0.02341 (0.01011)	0.02622 (0.02396)	–	0.02354 (0.00011)	0.02998 (0.00018)	-0.00814 (0.00014)	-0.03654 (0.0001)	-0.00007 (0.00019)
	μ=0.6, σ=1	-0.01091 (0.00052)	-0.00083 (0.00082)	-0.04763 (0.00114)	-0.0127 (0.00098)	-0.02063 (0.00088)	-0.01907 (0.00066)	–	-0.03488 (0.00051)	-0.00607 (0.00056)	-0.04324 (0.00193)	-0.09035 (0.00079)	-0.00027 (0.00052)
	μ=1.5, σ=0.8	0.02002 (0.00035)	0.02543 (0.00056)	0.02049 (0.00093)	-0.00568 (0.00072)	-0.00973 (0.00068)	-0.00891 (0.00061)	–	-0.00289 (0.00037)	0.02786 (0.00044)	0.02729 (0.00098)	-0.03209 (0.00055)	-0.00023 (0.00045)

Table A59: RMSE (Atkinson, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02763 (0.00035)	0.03112 (0.00027)	0.00842 (0.00019)	0.01406 (0.00036)	6.46838 (0.03205)	–	–	0.01875 (0.00021)	0.02948 (0.00037)	0.00337 (0.00051)	0.02336 (0.00018)	0.00297 (0.00028)
	α=1.5	0.18875 (0.00119)	0.18024 (0.0015)	0.06803 (0.00281)	0.17548 (0.0029)	6.44503 (0.19027)	6.59217 (0.20722)	6.99752 (0.37503)	0.19657 (0.00112)	0.13481 (0.00106)	0.04602 (0.00969)	0.20994 (0.00129)	0.04982 (0.00455)
	α=1.1	0.41324 (0.00345)	0.38963 (0.0038)	0.14269 (0.01073)	0.28809 (0.01326)	0.29693 (0.01265)	0.29484 (0.0109)	0.19688 (0.00554)	0.46964 (0.00379)	0.37546 (0.00383)	0.20157 (0.00599)	0.51418 (0.00408)	0.19877 (0.0073)
Burr	c=1.2, k=2.5	0.2085 (0.00043)	0.20446 (0.00038)	0.22915 (0.00025)	0.21424 (0.0004)	3.8981 (0.05292)	–	–	0.2145 (0.0003)	0.20772 (0.00041)	0.2206 (0.00038)	0.23657 (0.00025)	0.00713 (0.00071)
	c=3, k=1	0.029 (0.00034)	0.02537 (0.0003)	0.06235 (0.00023)	0.04172 (0.00034)	6.12408 (0.04774)	–	–	0.03751 (0.00019)	0.02741 (0.00032)	0.05396 (0.00037)	0.07667 (0.00019)	0.00396 (0.00061)
	c=1.3, k=1.5	0.1752 (0.00058)	0.15207 (0.00112)	0.20573 (0.00138)	0.1537 (0.0022)	0.1546 (0.00215)	–	–	0.18854 (0.00075)	0.17306 (0.00061)	0.19735 (0.00204)	0.24581 (0.00066)	0.0223 (0.00774)
	c=3, k=1.1/3	0.48395 (0.00024)	0.41219 (0.00037)	0.12054 (0.00347)	0.08253 (0.00367)	0.11341 (0.00424)	–	–	0.47101 (0.00039)	0.48005 (0.00023)	0.21449 (0.00946)	0.52159 (0.00057)	0.18959 (0.00668)
Lognormal	μ=0.5, σ=0.5	0.02621 (0.00013)	0.02743 (0.00014)	0.01544 (0.00012)	0.00745 (0.00022)	8.34449 (0.01254)	8.62284 (0.02025)	–	0.0236 (0.0001)	0.03005 (0.00013)	0.00817 (0.0001)	0.0365 (0.00011)	0.0013 (0.00013)
	μ=0.6, σ=1	0.01171 (0.00033)	0.00635 (0.00059)	0.04777 (0.00074)	0.01508 (0.00068)	0.02173 (0.0006)	0.01984 (0.00051)	–	0.03518 (0.00039)	0.00744 (0.00032)	0.04408 (0.00357)	0.09038 (0.00046)	0.00568 (0.0004)
	μ=1.5, σ=0.8	0.02011 (0.00031)	0.02571 (0.00043)	0.02173 (0.0007)	0.00745 (0.00035)	0.01089 (0.00039)	0.00989 (0.00036)	–	0.00444 (0.00024)	0.02789 (0.00035)	0.03486 (0.00362)	0.03242 (0.00032)	0.0033 (0.0002)

Table A60: MAD (Atkinson, n=5 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00257 (0.00024) (0.00014)	0.0026 (0.00025) (0.00021)	0.00174 (0.00018) (0.00009)	0.00275 (0.00032) (0.00018)	0.23703 (0.02272) (0.01491)	–	–	0.00154 (0.00016) (0.00013)	0.00273 (0.0002) (0.00014)	0.00199 (0.00028) (0.00026)	0.00138 (0.00012) (0.00005)	0.0019 (0.00018) (0.00007)
	α=1.5	0.00856 (0.00137) (0.00051)	0.00951 (0.00132) (0.00055)	0.0183 (0.00284) (0.00143)	0.01269 (0.00174) (0.00157)	1.46324 (0.22484) (0.12093)	1.52474 (0.19589) (0.13635)	0.63541 (0.11046) (0.06129)	0.00766 (0.0011) (0.00033)	0.00817 (0.00117) (0.00059)	0.0192 (0.00285) (0.00237)	0.00744 (0.00132) (0.00071)	0.02157 (0.00305) (0.00183)
	α=1.1	0.03589 (0.00329) (0.00225)	0.04076 (0.00277) (0.00248)	0.08053 (0.01224) (0.01035)	0.11388 (0.00846) (0.00485)	0.11849 (0.00795) (0.00305)	0.11763 (0.00754) (0.00519)	0.04628 (0.00428) (0.00292)	0.03242 (0.00306) (0.00188)	0.03472 (0.00228) (0.00116)	0.04577 (0.00696) (0.00333)	0.03143 (0.00212) (0.00163)	0.0584 (0.00503) (0.00411)
Burr	c=1.2, k=2.5	0.00279 (0.00045) (0.00026)	0.00303 (0.00043) (0.00023)	0.0018 (0.00023) (0.00008)	0.00295 (0.00049) (0.00019)	0.27568 (0.04781) (0.02765)	–	–	0.00209 (0.00032) (0.00014)	0.00306 (0.00032) (0.00028)	0.00201 (0.00022) (0.00016)	0.00167 (0.0002) (0.00012)	0.00454 (0.00069) (0.00021)
	c=3, k=1	0.00261 (0.00026) (0.00015)	0.00276 (0.00023) (0.00009)	0.00177 (0.0002) (0.00011)	0.00277 (0.00022) (0.00013)	0.24869 (0.03198) (0.02066)	–	–	0.00174 (0.00018) (0.00013)	0.0029 (0.00027) (0.0002)	0.00206 (0.00026) (0.00017)	0.00142 (0.00018) (0.00016)	0.00247 (0.00021) (0.00012)
	c=1.3, k=1.5	0.00364 (0.00044) (0.00031)	0.00626 (0.00085) (0.00074)	0.00737 (0.00104) (0.00108)	0.01136 (0.00137) (0.00104)	0.01094 (0.0013) (0.00082)	–	–	0.00433 (0.00052) (0.00025)	0.00373 (0.00044) (0.00032)	0.00972 (0.00116) (0.00096)	0.0036 (0.00044) (0.00036)	0.01067 (0.00112) (0.00062)
	c=3, k=1.1/3	0.00189 (0.0002) (0.00013)	0.00331 (0.00042) (0.00033)	0.03149 (0.00353) (0.00124)	0.04023 (0.00514) (0.0042)	0.0477 (0.00794) (0.00659)	–	–	0.00333 (0.00044) (0.0003)	0.00196 (0.00022) (0.00014)	0.05631 (0.00823) (0.00449)	0.00439 (0.00058) (0.00035)	0.05793 (0.0054) (0.00355)
Lognormal	μ=0.5, σ=0.5	0.00077 (0.00011) (0.00006)	0.00089 (0.00012) (0.00005)	0.00082 (0.00013) (0.00006)	0.00122 (0.00015) (0.00008)	0.07303 (0.00978) (0.00733)	0.07307 (0.01245) (0.00687)	–	0.0006 (0.0001) (0.00005)	0.00082 (0.00012) (0.00009)	0.00091 (0.00006) (0.00003)	0.00068 (0.0001) (0.00005)	0.00085 (0.00007) (0.00003)
	μ=0.6, σ=1	0.00289 (0.00038) (0.00033)	0.00448 (0.00062) (0.00058)	0.00672 (0.00068) (0.00036)	0.00635 (0.00072) (0.00044)	0.00568 (0.00057) (0.0004)	0.00508 (0.00046) (0.00028)	–	0.00328 (0.0004) (0.00025)	0.00304 (0.0004) (0.00026)	0.00764 (0.00091) (0.00051)	0.00354 (0.00034) (0.0003)	0.00376 (0.00032) (0.0002)
	μ=1.5, σ=0.8	0.00201 (0.00015) (0.00011)	0.00287 (0.00038) (0.00032)	0.00438 (0.00057) (0.0004)	0.00333 (0.00043) (0.00022)	0.00322 (0.00039) (0.00015)	0.00291 (0.00033) (0.00021)	–	0.00224 (0.00028) (0.00026)	0.00216 (0.00015) (0.00007)	0.00513 (0.00069) (0.0004)	0.00231 (0.00036) (0.00021)	0.00218 (0.00027) (0.00021)

TABLES OBTAINED FOR THE ATKINSON MEASURE FOR n=1 000

Table A61: Mean – true Atkinson (Atkinson, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02715 (0.00062)	0.03467 (0.0008)	-0.00568 (0.00053)	0.01735 (0.00086)	0.04175 (0.49156)	–	–	0.01974 (0.00049)	0.029 (0.00074)	0.00267 (0.00072)	-0.02228 (0.00035)	-0.00004 (0.00056)
	α=1.5	-0.20244 (0.00173)	-0.18894 (0.00226)	-0.06171 (0.00612)	-0.16434 (0.29912)	-0.09768 (0.40838)	-0.09308 (0.1218)	–	-0.20562 (0.00163)	-0.14757 (0.00172)	-0.01546 (0.00625)	-0.2201 (0.00144)	-0.0083 (0.00293)
	α=1.1	-0.50087 (0.00359)	-0.4711 (0.00389)	-0.16172 (0.01066)	-0.33651 (0.01427)	-0.3216 (0.01035)	-0.06139 (0.03516)	–	-0.53264 (0.00351)	-0.45768 (0.00343)	-0.24975 (0.00781)	-0.57754 (0.00306)	-0.21397 (0.01385)

Burr	c=1.2, k=2.5	-0.20809 (0.00093)	-0.20006 (0.00146)	-0.22656 (0.00097)	-0.20784 (0.00196)	-0.18115 (0.31029)	–	–	-0.21273 (0.0009)	-0.23526 (0.00103)	-0.20734 (0.00105)	-0.21758 (0.00074)	-0.00059 (0.00154)
	c=3, k=1	-0.02889 (0.00082)	-0.02119 (0.00102)	-0.05991 (0.00072)	-0.03738 (0.00114)	-0.01311 (0.43359)	–	–	-0.03621 (0.00066)	-0.02733 (0.0008)	-0.05154 (0.00114)	-0.07561 (0.00044)	-0.00026 (0.00092)
	c=1.3, k=1.5	-0.17536 (0.00109)	-0.15178 (0.00137)	-0.20385 (0.00161)	-0.14818 (0.00279)	-0.1449 (0.0453)	–	–	-0.18829 (0.00105)	-0.17315 (0.00115)	-0.19666 (0.00249)	-0.24557 (0.001)	-0.00283 (0.00327)
	c=3, k=1.1/3	-0.48553 (0.00064)	-0.41543 (0.00101)	-0.13246 (0.00839)	-0.08266 (0.01104)	-0.11373 (0.01125)	–	–	-0.47393 (0.00104)	-0.48155 (0.00061)	-0.25003 (0.01067)	-0.52515 (0.00145)	-0.21859 (0.0158)
Lognormal	μ=0.5, σ=0.5	0.02586 (0.00025)	0.03027 (0.00038)	-0.01448 (0.0003)	0.00925 (0.00045)	0.02491 (0.02906)	0.02922 (0.28854)	–	0.02367 (0.00018)	0.02964 (0.00028)	-0.00665 (0.00046)	-0.03658 (0.00018)	-0.00014 (0.00017)
	μ=0.6, σ=1	-0.01248 (0.00107)	0.0014 (0.00118)	-0.03841 (0.00183)	-0.0056 (0.00169)	-0.01564 (0.00146)	-0.01453 (0.00156)	–	-0.03478 (0.00105)	-0.00781 (0.00103)	-0.03249 (0.00292)	-0.09009 (0.00108)	0.00002 (0.00085)
	μ=1.5, σ=0.8	0.01758 (0.00105)	0.02849 (0.00077)	0.04716 (0.00506)	0.00201 (0.0013)	-0.00492 (0.00097)	-0.00608 (0.00101)	–	-0.00294 (0.00078)	0.02523 (0.00101)	0.05153 (0.0071)	-0.03209 (0.00065)	-0.00034 (0.00058)

Table A62: Median – true Atkinson (Atkinson, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02653 (0.00064)	0.0344 (0.00073)	-0.00625 (0.00059)	0.01645 (0.00108)	0.03683 (0.09207)	–	–	0.01949 (0.00055)	0.02849 (0.00083)	0.00129 (0.00064)	-0.02291 (0.0004)	-0.00089 (0.00059)
	α=1.5	-0.20558 (0.00185)	-0.19368 (0.00255)	-0.07587 (0.00967)	-0.19036 (0.51699)	-0.15376 (2.01286)	-0.10618 (0.01916)	–	-0.20951 (0.00167)	-0.1513 (0.00149)	-0.03307 (0.00496)	-0.22428 (0.0013)	-0.01961 (0.00314)
	α=1.1	-0.50224 (0.00395)	-0.47683 (0.00446)	-0.21774 (0.02338)	-0.4328 (0.01537)	-0.44664 (0.03048)	-0.0811 (0.04606)	–	-0.53855 (0.00436)	-0.45893 (0.00288)	-0.27841 (0.00741)	-0.58314 (0.00383)	-0.24484 (0.02003)
Burr	c=1.2, k=2.5	-0.20919 (0.00101)	-0.20144 (0.00145)	-0.22807 (0.00105)	-0.21078 (0.00225)	-0.18929 (0.10223)	–	–	-0.21358 (0.00081)	-0.23662 (0.00109)	-0.20792 (0.00108)	-0.22002 (0.00084)	-0.00187 (0.00145)
	c=3, k=1	-0.02965 (0.00098)	-0.02228 (0.00096)	-0.06096 (0.00071)	-0.03923 (0.00105)	-0.01797 (0.10552)	–	–	-0.03678 (0.00078)	-0.028 (0.001)	-0.0535 (0.00077)	-0.07665 (0.00037)	-0.00147 (0.00071)
	c=1.3, k=1.5	-0.17522 (0.00109)	-0.15211 (0.00171)	-0.20612 (0.00232)	-0.15434 (0.00398)	-0.15651 (0.00436)	–	–	-0.18839 (0.00127)	-0.17302 (0.00124)	-0.20428 (0.00235)	-0.24614 (0.00086)	-0.00949 (0.00217)
	c=3, k=1.1/3	-0.48546 (0.00062)	-0.4147 (0.00134)	-0.13292 (0.01074)	-0.08793 (0.01317)	-0.12062 (0.00993)	–	–	-0.47349 (0.00124)	-0.48144 (0.00069)	-0.27531 (0.01181)	-0.52522 (0.00196)	-0.24781 (0.01147)
Lognormal	μ=0.5, σ=0.5	0.02582 (0.00034)	0.02985 (0.00052)	-0.01454 (0.00049)	0.00919 (0.00076)	0.02479 (0.05269)	0.02877 (0.06677)	–	0.02355 (0.00027)	0.02961 (0.00045)	-0.00706 (0.00054)	-0.0367 (0.00035)	-0.00015 (0.00024)
	μ=0.6, σ=1	-0.01167 (0.00136)	0.0006 (0.00187)	-0.04229 (0.00257)	-0.00808 (0.00272)	-0.018 (0.00238)	-0.01678 (0.00211)	–	-0.03496 (0.0013)	-0.00689 (0.00165)	-0.04265 (0.00218)	-0.08994 (0.00107)	-0.00036 (0.00096)
	μ=1.5, σ=0.8	0.01815 (0.00132)	0.02818 (0.00091)	0.02876 (0.00169)	-0.00051 (0.0006)	-0.0059 (0.00069)	-0.00666 (0.00106)	–	-0.00315 (0.00095)	0.0258 (0.0014)	0.03126 (0.00318)	-0.03235 (0.00078)	-0.00058 (0.00105)

Table A63: RMSE (Atkinson, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.02794 (0.00063)	0.03573 (0.00081)	0.00833 (0.00039)	0.01992 (0.00095)	7.22175 (0.28693)	–	–	0.02041 (0.00048)	0.02982 (0.00079)	0.0081 (0.00106)	0.02267 (0.00033)	0.0065 (0.00095)
	α=1.5	0.20315 (0.00163)	0.19008 (0.00212)	0.08206 (0.00361)	4.3977 (0.21755)	4.42725 (0.36608)	0.18914 (0.28023)	–	0.20625 (0.00153)	0.14843 (0.00159)	0.07499 (0.01087)	0.22067 (0.00134)	0.04982 (0.00455)
	α=1.1	0.50306 (0.00339)	0.47375 (0.0037)	0.21294 (0.01)	0.3854 (0.01135)	0.37778 (0.01141)	0.09408 (0.03088)	–	0.53424 (0.00342)	0.45987 (0.00324)	0.27197 (0.0044)	0.57906 (0.003)	0.25144 (0.01038)
Burr	c=1.2, k=2.5	0.20824 (0.00091)	0.20032 (0.00142)	0.22666 (0.00095)	0.20828 (0.00186)	5.70637 (0.18703)	–	–	0.21285 (0.00088)	0.2075 (0.00102)	0.21782 (0.001)	0.23532 (0.00073)	0.01481 (0.00151)
	c=3, k=1	0.02973 (0.0007)	0.02307 (0.00071)	0.06024 (0.00066)	0.03879 (0.00091)	7.02455 (0.2679)	–	–	0.03664 (0.00059)	0.02829 (0.0007)	0.05225 (0.00087)	0.07575 (0.00043)	0.00872 (0.00125)
	c=1.3, k=1.5	0.17587 (0.0011)	0.15312 (0.00138)	0.20529 (0.00156)	0.15357 (0.00241)	0.39571 (0.39206)	–	–	0.18888 (0.00104)	0.17368 (0.00117)	0.20006 (0.00223)	0.24593 (0.00099)	0.03708 (0.00587)
	c=3, k=1.1/3	0.48551 (0.00064)	0.41548 (0.00101)	0.15525 (0.00656)	0.13118 (0.00677)	0.16525 (0.00763)	–	–	0.47397 (0.00105)	0.48154 (0.00061)	0.27921 (0.00666)	0.52524 (0.00146)	0.24979 (0.00966)
Lognormal	μ=0.5, σ=0.5	0.02598 (0.00025)	0.03051 (0.00038)	0.01481 (0.0003)	0.0102 (0.0004)	8.49154 (0.02902)	8.8595 (0.14254)	–	0.02376 (0.00019)	0.02977 (0.00028)	0.008 (0.00026)	0.03665 (0.00018)	0.00287 (0.00024)
	μ=0.6, σ=1	0.01672 (0.0009)	0.01252 (0.00097)	0.04452 (0.00167)	0.02072 (0.00124)	0.02315 (0.00102)	0.02116 (0.00098)	–	0.03633 (0.00095)	0.01395 (0.0009)	0.05178 (0.00547)	0.09077 (0.00104)	0.01161 (0.0006)
	μ=1.5, σ=0.8	0.01979 (0.001)	0.02992 (0.00076)	0.06753 (0.01023)	0.01502 (0.00224)	0.0125 (0.00101)	0.01083 (0.00042)	–	0.00827 (0.00052)	0.02693 (0.00101)	0.08126 (0.01737)	0.0331 (0.00058)	0.00788 (0.00057)

Table A64: MAD (Atkinson, n=1 000)

		midpoint1	midpoint2	Par_0	Par_1	Par_2	Par_3	Par_4	logN_midp	rand_midp	Rand_pareto	rand_ln	raw_data
Pareto	α=3	0.00423 (0.00046) (0.00019)	0.00587 (0.00055) (0.00033)	0.00402 (0.00032) (0.00025)	0.00682 (0.00064) (0.00048)	0.28764 (0.08875) (0.06524)	–	–	0.00333 (0.00035) (0.00021)	0.00469 (0.0004) (0.00032)	0.00447 (0.00028) (0.0001)	0.00275 (0.00025) (0.00013)	0.0038 (0.00035) (0.00021)
	α=1.5	0.01075 (0.00132) (0.00096)	0.0135 (0.00195) (0.00169)	0.03578 (0.00649) (0.00507)	1.98788 (0.5863) (0.3286)	2.34962 (1.6578) (1.81641)	0.02437 (0.01033) (0.00755)	–	0.00998 (0.0014) (0.00085)	0.00982 (0.00123) (0.00107)	0.03141 (0.00302) (0.0024)	0.00915 (0.0014) (0.00061)	0.02157 (0.00305) (0.00183)
	α=1.1	0.03151 (0.0036) (0.00233)	0.02977 (0.00219) (0.00189)	0.0801 (0.0154) (0.00756)	0.07872 (0.01926) (0.01236)	0.04898 (0.02734) (0.00605)	0.04034 (0.04223) (0.02663)	–	0.02374 (0.00136) (0.00087)	0.03007 (0.00362) (0.00087)	0.05122 (0.00765) (0.0035)	0.02428 (0.00204) (0.00163)	0.07625 (0.01465) (0.00714)
Burr	c=1.2, k=2.5	0.0048 (0.00041) (0.00025)	0.00674 (0.00073) (0.00035)	0.00422 (0.00051) (0.00041)	0.00813 (0.00092) (0.00067)	0.30779 (0.1049) (0.05644)	–	–	0.00436 (0.00029) (0.00014)	0.00504 (0.00039) (0.00023)	0.00444 (0.00058) (0.00037)	0.00313 (0.00054) (0.00042)	0.00943 (0.00122) (0.00054)
	c=3, k=1	0.00435 (0.00045) (0.00032)	0.00623 (0.00059) (0.00043)	0.00421 (0.00037) (0.00035)	0.00686 (0.00063) (0.00034)	0.35015 (0.10205) (0.04664)	–	–	0.00361 (0.00044) (0.00041)	0.00459 (0.0005) (0.00034)	0.00412 (0.00037) (0.00027)	0.00271 (0.00024) (0.00013)	0.00512 (0.00046) (0.00039)
	c=1.3, k=1.5	0.0081 (0.00092) (0.00047)	0.01343 (0.00203) (0.00129)	0.01571 (0.00207) (0.00065)	0.0248 (0.00333) (0.0008)	0.02367 (0.00367) (0.00198)	–	–	0.00947 (0.00152) (0.00047)	0.00837 (0.0008) (0.00053)	0.01645 (0.00175) (0.00105)	0.00824 (0.00115) (0.00036)	0.01889 (0.00235) (0.00092)

	c=3, k=1.1/3	0.00395 (0.00054) (0.00046)	0.00643 (0.00101) (0.0004)	0.05963 (0.00718) (0.00341)	0.07318 (0.00887) (0.00678)	0.08208 (0.01166) (0.00792)	–	–	0.00662 (0.00084) (0.00063)	0.00413 (0.00068) (0.00063)	0.06896 (0.01038) (0.00623)	0.00835 (0.00128) (0.00115)	0.06674 (0.00682) (0.00692)
Lognormal	μ=0.5, σ=0.5	0.0017 (0.00017) (0.00013)	0.00237 (0.00033) (0.00025)	0.00203 (0.0002) (0.00013)	0.00271 (0.0002) (0.00017)	0.14655 (0.02254) (0.00903)	0.16074 (0.04591) (0.03371)	–	0.00135 (0.00005) (0.00003)	0.00182 (0.00019) (0.00013)	0.00257 (0.0003) (0.00024)	0.00151 (0.0002) (0.00007)	0.00192 (0.00024) (0.00014)
	μ=0.6, σ=1	0.00715 (0.00084) (0.00074)	0.00816 (0.0012) (0.00111)	0.01441 (0.00134) (0.00051)	0.01293 (0.00135) (0.00088)	0.01124 (0.00141) (0.00075)	0.01007 (0.00139) (0.00094)	–	0.00704 (0.0008) (0.00059)	0.00771 (0.00097) (0.00089)	0.01316 (0.00169) (0.00138)	0.00732 (0.00079) (0.00065)	0.00768 (0.0007) (0.00047)
	μ=1.5, σ=0.8	0.00647 (0.00071) (0.00028)	0.00623 (0.00075) (0.00062)	0.01396 (0.00167) (0.00055)	0.00885 (0.00081) (0.00041)	0.00754 (0.00064) (0.00036)	0.00589 (0.00053) (0.00033)	–	0.00533 (0.0006) (0.00043)	0.00689 (0.00076) (0.00052)	0.01231 (0.00245) (0.00142)	0.00563 (0.00041) (0.00031)	0.00521 (0.00054) (0.00035)

Appendix B:
Programming code

Please note that the programming code for only the Burr distribution with parameters c = 1.2 and k = 2.5 and n = 15 000 is displayed. For the other distributions, parameters and n values the programs were slightly adjusted. The programming was done in statistical programming packages R and SAS.

Program 1:

In Program 1 observations are simulated for a specific distribution and parameters. The sample size (n) and number of replicates are also specified. This was coded in R.

```
fix(sim_Burr)

function (n,c,k,N)
{
density_burr=function(x, c, k) {c*k*x^(c-1)*(1+x^c)^(-k-1)}
distribution_burr=function(x, c, k) {1-(1+x^c)^(-k)}
quantile_burr=function(u, c, k) {((1-u)^(-1/k)-1)^(1/c)}
random_burr=function(n, c, k) quantile_burr(u=runif(n),c,k)

simulated_data<<-matrix(random_burr(n*N,c,k),nrow=n*N,ncol=1)
repl<-rep(1:N,each=n)
dat_mat<<-matrix(c(repl,simulated_data),ncol=2,byrow=FALSE)
hh_mat<-matrix(0,ncol=5,nrow=N)

for(i in 1:N)
{
dat<-dat_mat[dat_mat[,1]==i,2]
hh_mat[i,]<-hist(dat,breaks=c(0,2,4,8,16,max(dat)))$counts
}
print(hh_mat)
}

sim_Burr(n=15000,c=1.2,k=2.5,N=1300)
write.csv(simulated_data,file =
"C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/Sim_dat.csv")
```

Program 2:

In Program 2, the simulated data is categorised into interval data and the Pareto means method is performed on the interval data. The interval data with the results for the Pareto means method are exported. This was programmed in R.

```
fix(ParMidp)

function ()
{
lowerbound<-c(1,2,4,8,16)
upperbound<-c(2,4,8,16,Inf)
n<-length(lowerbound)
N<-max(dat_mat[,1])
pp<-1
ff<-1

ParMat_f<-matrix(0,ncol=9,nrow=5*N)
Volmat_f<-matrix(0,ncol=8,nrow=15000*N)
Volmat<-NULL

for(f in 1:N)
{
simulated_data<-dat_mat[dat_mat[,1]==f,2]

interval<-cut(simulated_data,
breaks=c(0,2,4,8,16,max(simulated_data)+1),right=FALSE)
freq<-table(interval)

Y<-lowerbound
N<-rep(0,n)
for(i in 1:n)
{
aa<-i:n
N[i]<-sum(freq[aa])
}

newN<-N[N!=0]
newY<-Y[N!=0]

alpha<-rep(0,length(newN)-2)
for(i in 1:(length(newN)-2))
{
bb<-i:length(newN)
trimN<-newN[bb]
trimY<-newY[bb]
mod<-lm(log(trimN)~log(trimY))
alpha[i]<-mod$coefficients[2]
}
results_mat<-matrix(0,ncol=5,nrow=n)
results_mat[,1]<-lowerbound
results_mat[,2]<-upperbound
results_mat[,3]<-freq
results_mat[,4]<-1:length(lowerbound)
results_mat[,5]<-rep(length(newN),length(lowerbound))

new_alpha<-(-alpha)
midp_mat<-matrix(0,ncol=length(newN)-2,nrow=length(newN))

for(i in 1:(length(newN)-2))
{
midp<-rep(0,length(newN))
```

```
zz<-0:(i-1)

        for(j in 1:(length(newN)-1))
        {
        midp[j]<-(new_alpha[i]/(1-new_alpha[i]))*((upperbound[j]^(-
new_alpha[i]+1)-lowerbound[j]^(-new_alpha[i]+1))/(lowerbound[j]^(-
new_alpha[i])-
        upperbound[j]^(-new_alpha[i])))
        }

        if(i>1)
        {
                for(k in 1:length(zz))
                {
                midp[zz][k-1]<-trunc(mean(c(lowerbound[k-1],upperbound[k-1])))
                }
        }
midp[length(newN)]<-(new_alpha[i]/(new_alpha[i]-1))*lowerbound[length(newN)]
midp_mat[,i]<-midp
}
repl_freq<-(rep(f,length(newN)))
ParMat<-cbind(results_mat[1:length(newN),],midp_mat)
ParMat_f[pp:(pp+(length(newN)-1)),1:(length(newN)+4)]<-
cbind(repl_freq,ParMat)

Volmat2<-matrix(0,nrow=sum(freq),ncol=length(newN)+3)
num<-c(1,2,4:ncol(ParMat))

for(i in num)
{
sp<-NULL
        for(j in 1:nrow(ParMat))
        {
        sp<-c(sp,rep(ParMat[j,i],ParMat[j,3]))
        }
Volmat2[,i]<-sp
}
repl_vol<-rep(f,15000)
Volmat2<-Volmat2[,-3]

Volmat_f[ff:(ff+14999),1:(length(newN)+3)]<-cbind(repl_vol,Volmat2)
colnames(Volmat2)<-
c("Lbound","Ubound","IncCat","LastCat",paste("Par",length(newN):3,sep=""))
Volmat<-rbind(Volmat,Volmat2[,1:4])
pp<-pp+5
ff<-ff+15000
}
colnames(ParMat_f)<-
c("Replicate","Lbound","Ubound","Freq","IncCat","LastCat",paste("Par_",0:2,se
p=""))
colnames(Volmat_f)<-
c("Replicate","Lbound","Ubound","IncCat","LastCat",paste("Par_",0:2,sep=""))

write.table(ParMat_f,
file="C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/ParetoMidp_Freq.txt", row.names=FALSE, col.names=TRUE)
write.table(Volmat_f,
file="C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/ParetoMidp.txt", row.names=FALSE, col.names=TRUE)
}

ParMidp()
```

Program 3:

The exported data of Program 2 is imported into the SAS program, Program 3. The calculations for the rest of the methods are performed in this program on the interval data.

```
proc import out = ParetoMidp
file = 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\ParetoMidp.txt'
dbms = dlm replace;
getnames = yes;
DATAROW=2;
run;

proc import out = ParetoMidp_freq
file = 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\ParetoMidp_Freq.txt'
dbms = dlm replace;
getnames = yes;
DATAROW=2;
run;

proc sort data=ParetoMidp; by replicate IncCat;
run;

data aa (keep=replicate uit);
set ParetoMidp; by replicate IncCat;
if last.inccat and incCat=LastCat;

if Par_2<0 or Par_1<0 then uit=1;
if Par2 > 100000 then uit=1;
if Par_1 = 0 and Par_2 = 0 then uit=1;
run;

proc means data=aa n sum;
run;

data bb (drop=replicate);
merge ParetoMidp aa; by replicate;
if uit = 1 then delete;

repl=replicate;
run;

data cc (drop=Nr);
set bb;
Nr = _N_;
Replicate = Int(Nr/15000-0.000001)+1;
if replicate GT 1000 then delete;
run;

*Midpoint method;
data midp (drop=uit repl);
set cc;

if Par_2=0 then Par_2=.;

if IncCat = 1 then do; midpoint1 = (Lbound-1+Ubound)/2; midpoint2 = (Lbound-
1+Ubound)/2; end;
else if IncCat LT LastCat then do; midpoint1 = (Lbound+Ubound)/2; midpoint2 =
(Lbound+Ubound)/2; end;
else if IncCat = LastCat then do; midpoint1 = Lbound*1.10; midpoint2 =
Lbound*2; end;
```

```
run;

*Lognormal means method;
proc reliability data = midp;
by Replicate;
distribution lognormal;
where IncCat GT 1;
ods output parmest = LN_estimates;
pplot (Lbound Ubound);
run;

data location (keep=mu Replicate) scale (keep=sigma Replicate);
set LN_estimates; by Replicate;
if key = 1 then do; mu = estimate; output location; end;
if key = 2 then do; sigma = estimate; output scale; end;
run;

data LocScale;
merge Location Scale; by Replicate;
run;

data lognorm;
merge midp LocScale; by Replicate;

if IncCat LT lastCat then do;
PDF_l=PDF('NORMAL',(log(Lbound)-mu)/sigma);
PDF_u=PDF('NORMAL',(log(Ubound)-mu)/sigma);
CDF_l=CDF('NORMAL',(log(Lbound)-mu)/sigma);
CDF_u=CDF('NORMAL',(log(Ubound)-mu)/sigma); end;
if IncCat EQ LastCat then do;
PDF_l=PDF('NORMAL',(log(Lbound)-mu)/sigma);
PDF_u=0;
CDF_l=CDF('NORMAL',(log(Lbound)-mu)/sigma);
CDF_u=1; end;


if IncCat = 1 then logN_midp = (Lbound-1+Ubound)/2;
else logN_midp=exp(mu-sigma*((PDF_u-PDF_l)/(CDF_u-CDF_l)));

if IncCat = LastCat and Lbound LE logN_midp then correct = 1;
else if Lbound LE logN_midp LE Ubound then correct = 1;
run;

proc means data=lognorm N noprint;
var correct;
by Replicate;
output out = uit_correct N=N_correct;
run;

data telcorrect;
set uit_correct;

if N_correct = 15000 then repl_Correct=1;
else repl_Correct =0;
run;

proc freq data=telcorrect;
tables repl_Correct;
run;

data log_par (drop = mu sigma PDF_l PDF_u CDF_l CDF_u correct _TYPE_ _FREQ_
N_correct);
merge lognorm uit_correct; by Replicate;
```

```
if N_correct NE 15000 then logN_midp = .;
run;

*Random lognormal method;
data Random_ln;
merge log_par LocScale; by Replicate;

array Rand_LN Inc_sp1 - Inc_sp5;

do over Rand_LN;

Rand_num = rand("Uniform");
if IncCat = LastCat then x_star=probnorm((log(Lbound)-mu)/sigma)+Rand_num*(1-
probnorm((log(Lbound)-mu)/sigma));
else x_star=probnorm((log(Lbound)-mu)/sigma)+Rand_num*(probnorm((log(Ubound)-
mu)/sigma)-probnorm((log(Lbound)-mu)/sigma));

Rand_LN=exp(mu+sigma*probit(x_star));

end;

if Inc_sp1 LE 100000 then Rand_logn=Inc_sp1;
else if Inc_sp2 LE 100000 then Rand_logn=Inc_sp2;
else if Inc_sp3 LE 100000 then Rand_logn=Inc_sp3;
else if Inc_sp4 LE 100000 then Rand_logn=Inc_sp4;
else if Inc_sp5 LE 100000 then Rand_logn=Inc_sp5;
else do; Rand_logn=min(of Inc_sp1-Inc_sp5); Groot=1; end;
run;

proc print data=Random_ln;
where groot=1;
run;

data rand_LN (drop=mu sigma Inc_sp1 Inc_sp2 Inc_sp3 Inc_sp4 Inc_sp5 Rand_num
x_star Rand_logn groot);
set Random_ln;
rand_ln=Rand_logn;
run;

proc print data=rand_LN;
where (IncCat NE LastCat) and (rand_ln NE .) and ((rand_ln GT Ubound) or
(rand_ln LT Lbound));
run;

*Random midpoint method;
data rand_midp (drop = sign U);
set rand_LN;
U = rand("Uniform");
if U LE 0.5 then sign = -1;
else sign = 1;
rand_midp = midpoint1 + sign*rand("Uniform")*(midpoint1-Lbound);
run;

*Random Pareto method;
proc freq data = rand_midp noprint;
tables IncCat / outcum out = freq_uit;
by replicate;
run;

data Pareto_freq;
set Paretomidp_freq;
if IncCat = 0 then delete;
if Par_2=0 then Par_2=.;
```

```
if Ubound = "Inf" then Ubound = .;
run;

proc sort data=Pareto_freq; by replicate;
run;

data bb_freq (drop=replicate);
merge Pareto_freq aa; by replicate;
if uit = 1 then delete;
repl=replicate;
run;

proc sort data=cc out=cc1 nodupkey; by repl;
run;

data cc1kort (keep=repl replicate);
set cc1;
run;

data cc_freq (drop=uit repl);
merge cc1kort bb_freq; by repl;
if Replicate = . then delete;
run;

proc sort data=cc_freq; by replicate IncCat;
run;

data freqdat (drop = CUM_PCT PERCENT COUNT PCum1 PCum2);
merge cc_freq freq_uit; by replicate IncCat;
lnY = log(Lbound);

PCum1 = lag(Cum_Freq);
PCum2 = lag2(Cum_Freq);
PCum=PCum1;
if PCum1 = . then PCum=PCum2;

if IncCat=1 then NN = 15000;
else NN = 15000 - PCum;
lnNN = log(NN);
run;

proc reg data = freqdat outest = est noprint;
model lnNN = LnY / selection = rsquare adjrsq b;
by Replicate;
run;

data Rand_partoA (drop = _MODEL_ _TYPE_ _DEPVAR_ _RMSE_ Intercept lnY LnNN
_IN_ _P_ _EDF_ _RSQ_ _ADJRSQ_ Rand_num X_star);
merge rand_midp Est; by replicate;

array Rand_Par Inc_sp1 - Inc_sp5;

do over Rand_Par;

Rand_num = rand("Uniform");
if IncCat = LastCat then X_star = (1-
(intercept/Lbound)**(abs(lnY)))+Rand_num*(1-
(1-(intercept/Lbound)**(abs(lnY))));
else X_star = (1-(intercept/Lbound)**(abs(lnY)))+Rand_num*((1-
(intercept/Ubound)**(abs(lnY)))-
(1-(intercept/Lbound)**(abs(lnY))));
Rand_Par = intercept*(1-X_star)**(-1/abs(lnY));
end;
```

```
if Inc_sp1 LE 100000 then Rand_pareto=Inc_sp1;
else if Inc_sp2 LE 100000 then Rand_pareto=Inc_sp2;
else if Inc_sp3 LE 100000 then Rand_pareto=Inc_sp3;
else if Inc_sp4 LE 100000 then Rand_pareto=Inc_sp4;
else if Inc_sp5 LE 100000 then Rand_pareto=Inc_sp5;
else do; Rand_pareto=min(of Inc_sp1-Inc_sp5); Groot=1; end;
run;

proc print data=Rand_partoA ;
where groot=1;
run;

data Rand_pareto (drop = Inc_sp1 Inc_sp2 Inc_sp3 Inc_sp4 Inc_sp5 Groot);
set Rand_partoA;
run;

proc export data = Rand_pareto
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\Dat_methods.txt"
dbms = dlm replace;
run;

data rand_pareto_part1 rand_pareto_part2 rand_pareto_part3 rand_pareto_part4
rand_pareto_part5;
set Rand_pareto;
if 1 le replicate le 200 then output rand_pareto_part1;
else if 201 le replicate le 400 then output rand_pareto_part2;
else if 401 le replicate le 600 then output rand_pareto_part3;
else if 601 le replicate le 800 then output rand_pareto_part4;
else if 801 le replicate le 1000 then output rand_pareto_part5;
run;

proc export data = rand_pareto_part1
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\rand_pareto_part1.txt"
dbms = dlm replace;
run;

proc export data = rand_pareto_part2
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\rand_pareto_part2.txt"
dbms = dlm replace;
run;

proc export data = rand_pareto_part3
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\rand_pareto_part3.txt"
dbms = dlm replace;
run;

proc export data = rand_pareto_part4
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\rand_pareto_part4.txt"
dbms = dlm replace;
run;

proc export data = rand_pareto_part5
outfile = "C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\rand_pareto_part5.txt"
dbms = dlm replace;
run;
```

Program 4:

The results of all the methods obtained in Program 3, are imported into Program 4, where the measures of inequality are calculated. R was used for the coding.

```
dat<-
read.table("C:\\Users\\Francois\\Documents\\SU2014\\Meestersprojek\\Programmi
ng SimBurr\\rand_pareto_part1.txt",header=T,sep = " ")

fix(measures)

function (dat)
{
require(ineq)
require(laeken)
GiniMat<-matrix(0,ncol=9,nrow=200)
QSRMat<-matrix(0,ncol=9,nrow=200)
TheilMat<-matrix(0,ncol=9,nrow=200)
AtkMat<-matrix(0,ncol=9,nrow=200)
colnames(GiniMat)<-
c("Par_0","Par_1","Par_2","midpoint1","midpoint2","logN_midp","rand_ln","rand
_midp","Rand_pareto")
colnames(QSRMat)<-
c("Par_0","Par_1","Par_2","midpoint1","midpoint2","logN_midp","rand_ln","rand
_midp","Rand_pareto")
colnames(TheilMat)<-
c("Par_0","Par_1","Par_2","midpoint1","midpoint2","logN_midp","rand_ln","rand
_midp","Rand_pareto")
colnames(AtkMat)<-
c("Par_0","Par_1","Par_2","midpoint1","midpoint2","logN_midp","rand_ln","rand
_midp","Rand_pareto")

for(i in 1:200)
{
GiniMat[i,1]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_0"],type="Gini")
GiniMat[i,2]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_1"],type="Gini")
GiniMat[i,3]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_2"],type="Gini")
GiniMat[i,4]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint1"],type="Gini")
GiniMat[i,5]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint2"],type="Gini")
GiniMat[i,6]<-ineq(dat[dat[,"Replicate"]==i,][,"logN_midp"],type="Gini")
GiniMat[i,7]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_ln"],type="Gini")
GiniMat[i,8]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_midp"],type="Gini")
GiniMat[i,9]<-ineq(dat[dat[,"Replicate"]==i,][,"Rand_pareto"],type="Gini")

QSRMat[i,1]<-qsr(dat[dat[,"Replicate"]==i,][,"Par_0"])$value
QSRMat[i,2]<-qsr(dat[dat[,"Replicate"]==i,][,"Par_1"])$value
QSRMat[i,3]<-qsr(dat[dat[,"Replicate"]==i,][,"Par_2"])$value
QSRMat[i,4]<-qsr(dat[dat[,"Replicate"]==i,][,"midpoint1"])$value
QSRMat[i,5]<-qsr(dat[dat[,"Replicate"]==i,][,"midpoint2"])$value
QSRMat[i,6]<-qsr(dat[dat[,"Replicate"]==i,][,"logN_midp"])$value
QSRMat[i,7]<-qsr(dat[dat[,"Replicate"]==i,][,"rand_ln"])$value
QSRMat[i,8]<-qsr(dat[dat[,"Replicate"]==i,][,"rand_midp"])$value
QSRMat[i,9]<-qsr(dat[dat[,"Replicate"]==i,][,"Rand_pareto"])$value

TheilMat[i,1]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_0"],type="Theil")
TheilMat[i,2]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_1"],type="Theil")
TheilMat[i,3]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_2"],type="Theil")
TheilMat[i,4]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint1"],type="Theil")
TheilMat[i,5]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint2"],type="Theil")
TheilMat[i,6]<-ineq(dat[dat[,"Replicate"]==i,][,"logN_midp"],type="Theil")
TheilMat[i,7]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_ln"],type="Theil")
TheilMat[i,8]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_midp"],type="Theil")
```

```
TheilMat[i,9]<-ineq(dat[dat[,"Replicate"]==i,][,"Rand_pareto"],type="Theil")

AtkMat[i,1]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_0"],type="Atkinson")
AtkMat[i,2]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_1"],type="Atkinson")
AtkMat[i,3]<-ineq(dat[dat[,"Replicate"]==i,][,"Par_2"],type="Atkinson")
AtkMat[i,4]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint1"],type="Atkinson")
AtkMat[i,5]<-ineq(dat[dat[,"Replicate"]==i,][,"midpoint2"],type="Atkinson")
AtkMat[i,6]<-ineq(dat[dat[,"Replicate"]==i,][,"logN_midp"],type="Atkinson")
AtkMat[i,7]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_ln"],type="Atkinson")
AtkMat[i,8]<-ineq(dat[dat[,"Replicate"]==i,][,"rand_midp"],type="Atkinson")
AtkMat[i,9]<-ineq(dat[dat[,"Replicate"]==i,][,"Rand_pareto"],type="Atkinson")
print(i)
}
#print(GiniMat[1:5,])
print(apply(GiniMat,2,mean))
print(apply(QSRMat,2,mean))
print(apply(TheilMat,2,mean))
print(apply(AtkMat,2,mean))
write.csv(GiniMat,file                                                          =
"C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/GiniMat_part1.csv")
write.csv(QSRMat,file                                                           =
"C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/QSRMat_part1.csv")
write.csv(TheilMat,file                                                         =
"C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/TheilMat_part1.csv")
write.csv(AtkMat,file                                                           =
"C:/Users/Francois/Documents/SU2014/Meestersprojek/Programming
SimBurr/AtkMat_part1.csv")
}

measures(dat)
```

Program 5:

The measures of inequality calculated in Program 4 are imported into Program 5. In this program, the measures of performance are calculated on the measures of inequality. This is the final program and was coded in SAS.

```
proc import OUT = Gini
datafile= 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\Burr c=3,k=1,n=15000\Results.xlsx'
dbms = xlsx replace;
getnames = yes;
sheet='Gini';
RUN;

data Gini (drop=A Par_2);
set Gini (obs=1000);
if Par_2 = "NA" then Par_2= .;
NPar_2 = input(Par_2, 12.);
nom = _N_;
run;

proc import OUT = QSR
datafile= 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\Burr c=3,k=1,n=15000\Results.xlsx'
dbms = xlsx replace;
getnames = yes;
sheet='QSR';
RUN;

data QSR (drop=A Par_2);
set QSR (obs=1000);
if Par_2 = "NA" then Par_2= .;
NPar_2 = input(Par_2, 12.);
nom = _N_;
run;

proc import OUT = Theil
datafile= 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\Burr c=3,k=1,n=15000\Results.xlsx'
dbms = xlsx replace;
getnames = yes;
sheet='Theil';
RUN;

data Theil (drop=A Par_2);
set Theil (obs=1000);
if Par_2 = "NA" then Par_2= .;
NPar_2 = input(Par_2, 12.);
nom = _N_;
run;

proc import OUT = Atk
datafile= 'C:\Users\Francois\Documents\SU2014\Meestersprojek\Programming
SimBurr\Burr c=3,k=1,n=15000\Results.xlsx'
dbms = xlsx replace;
getnames = yes;
sheet='Atk';
RUN;

data Atk (drop=A Par_2);
set Atk (obs=1000);
if Par_2 = "NA" then Par_2= .;
```

```
NPar_2 = input(Par_2, 12.);
nom = _N_;
run;

*****************************************************************************;
*Standard error of mean Gini;
data Gini_blok;
set Gini;
if 1 LE nom LE 100 then blok=1;
else if 101 LE nom LE 200 then blok=2;
else if 201 LE nom LE 300 then blok=3;
else if 301 LE nom LE 400 then blok=4;
else if 401 LE nom LE 500 then blok=5;
else if 501 LE nom LE 600 then blok=6;
else if 601 LE nom LE 700 then blok=7;
else if 701 LE nom LE 800 then blok=8;
else if 801 LE nom LE 900 then blok=9;
else if 901 LE nom LE 1000 then blok=10;
rename NPar_2 = Par_2;
run;

proc sort data=Gini_blok; by blok;
run;

proc means data=Gini_blok n mean noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BM_gini mean = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

proc means data=BM_gini n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=uit_gini n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto M_logN_midp
Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = uit_gini;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

*****************************************************************************;
*Standard error of mean QSR;
data QSR_blok;
set QSR;
if 1 LE nom LE 100 then blok=1;
else if 101 LE nom LE 200 then blok=2;
else if 201 LE nom LE 300 then blok=3;
else if 301 LE nom LE 400 then blok=4;
else if 401 LE nom LE 500 then blok=5;
else if 501 LE nom LE 600 then blok=6;
else if 601 LE nom LE 700 then blok=7;
else if 701 LE nom LE 800 then blok=8;
else if 801 LE nom LE 900 then blok=9;
else if 901 LE nom LE 1000 then blok=10;
```

```
rename NPar_2 = Par_2;
run;

proc sort data=QSR_blok; by blok;
run;

proc means data=QSR_blok n mean noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BM_qsr mean = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2
BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

proc means data=BM_qsr n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=uit_qsr n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto
M_logN_midp Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = uit_qsr;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

***************************************************************************;
*Standard error of mean Theil;
data Theil_blok;
set Theil;
if 1 LE nom LE 100 then blok=1;
else if 101 LE nom LE 200 then blok=2;
else if 201 LE nom LE 300 then blok=3;
else if 301 LE nom LE 400 then blok=4;
else if 401 LE nom LE 500 then blok=5;
else if 501 LE nom LE 600 then blok=6;
else if 601 LE nom LE 700 then blok=7;
else if 701 LE nom LE 800 then blok=8;
else if 801 LE nom LE 900 then blok=9;
else if 901 LE nom LE 1000 then blok=10;
rename NPar_2 = Par_2;
run;

proc sort data=Theil_blok; by blok;
run;

proc means data=Theil_blok n mean noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BM_theil mean = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

proc means data=BM_theil n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
```

```
BM_rand_ln BM_raw_data;
output out=uit_theil n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1
M_midpoint2 M_rand_midp M_Rand_pareto
M_logN_midp Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = uit_theil;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

****************************************************************************;
*Standard error of mean Atk;
data Atk_blok;
set Atk;
if 1 LE nom LE 100 then blok=1;
else if 101 LE nom LE 200 then blok=2;
else if 201 LE nom LE 300 then blok=3;
else if 301 LE nom LE 400 then blok=4;
else if 401 LE nom LE 500 then blok=5;
else if 501 LE nom LE 600 then blok=6;
else if 601 LE nom LE 700 then blok=7;
else if 701 LE nom LE 800 then blok=8;
else if 801 LE nom LE 900 then blok=9;
else if 901 LE nom LE 1000 then blok=10;
rename NPar_2 = Par_2;
run;

proc sort data=Atk_blok; by blok;
run;

proc means data=Atk_blok n mean noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BM_atk mean = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2
BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

proc means data=BM_atk n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=uit_atk n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto
M_logN_midp Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = uit_atk;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

****************************************************************************;
****************************************************************************;
*Standard error of RMSE Gini;
data Gini_aa;
set Gini_blok;
```

```
true_val=0.3333;
diff_sq_P0 = (Par_0 - true_val)**2;
diff_sq_P1 = (Par_1 - true_val)**2;
diff_sq_P2 = (Par_2 - true_val)**2;
diff_sq_P3 = (Par_3 - true_val)**2;
diff_sq_P4 = (Par_4 - true_val)**2;
diff_sq_Mid1 = (midpoint1 - true_val)**2;
diff_sq_Mid2 = (midpoint2 - true_val)**2;
diff_sq_Rmid = (rand_midp - true_val)**2;
diff_sq_Rpar = (Rand_pareto - true_val)**2;
diff_sq_Mln = (logN_midp - true_val)**2;
diff_sq_Rln = (rand_ln - true_val)**2;
diff_sq_Raw = (raw_data - true_val)**2;
run;

proc sort data=Gini_aa; by blok;
run;

proc means data=Gini_aa mean noprint;
by blok;
var diff_sq_P0 diff_sq_P1 diff_sq_P2 diff_sq_Mid1 diff_sq_Mid2 diff_sq_Rmid
diff_sq_Rpar diff_sq_Mln
diff_sq_Rln diff_sq_Raw;
output out=Gini_aa_uit mean=Mdiff_sq_P0 Mdiff_sq_P1 Mdiff_sq_P2 Mdiff_sq_Mid1
Mdiff_sq_Mid2
Mdiff_sq_Rmid Mdiff_sq_Rpar Mdiff_sq_Mln Mdiff_sq_Rln Mdiff_sq_Raw;
run;

data Gini_RMSE_b (drop= _type_ _freq_);
set Gini_aa_uit;
P0_RMSE_b=sqrt(Mdiff_sq_P0);
P1_RMSE_b=sqrt(Mdiff_sq_P1);
P2_RMSE_b=sqrt(Mdiff_sq_P2);
Mid1_RMSE_b=sqrt(Mdiff_sq_Mid1);
Mid2_RMSE_b=sqrt(Mdiff_sq_Mid2);
Rmid_RMSE_b=sqrt(Mdiff_sq_Rmid);
Rpar_RMSE_b=sqrt(Mdiff_sq_Rpar);
Mln_RMSE_b=sqrt(Mdiff_sq_Mln);
Rln_RMSE_b=sqrt(Mdiff_sq_Rln);
raw_RMSE_b=sqrt(Mdiff_sq_Raw);
run;

proc means data=Gini_RMSE_b mean std noprint;
var P0_RMSE_b P1_RMSE_b P2_RMSE_b Mid1_RMSE_b Mid2_RMSE_b Rmid_RMSE_b
Rpar_RMSE_b Mln_RMSE_b Rln_RMSE_b
raw_RMSE_b;
output out=Gini_RMSE mean=P0_RMSE_m P1_RMSE_m P2_RMSE_m Mid1_RMSE_m
Mid2_RMSE_m Rmid_RMSE_m Rpar_RMSE_m
Mln_RMSE_m Rln_RMSE_m raw_RMSE_m
std=P0_RMSE_se P1_RMSE_se P2_RMSE_se Mid1_RMSE_se Mid2_RMSE_se Rmid_RMSE_se
Rpar_RMSE_se Mln_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

proc print data = Gini_RMSE;
var Mid1_RMSE_m Mid2_RMSE_m P0_RMSE_m P1_RMSE_m P2_RMSE_m Mln_RMSE_m
Rmid_RMSE_m Rpar_RMSE_m Rln_RMSE_m
raw_RMSE_m;
run;

proc print data = Gini_RMSE;
var Mid1_RMSE_se Mid2_RMSE_se P0_RMSE_se P1_RMSE_se P2_RMSE_se Mln_RMSE_se
Rmid_RMSE_se Rpar_RMSE_se
```

```
Rln_RMSE_se raw_RMSE_se;
run;

***************************************************************************;
*Standard error of RMSE QSR;
data QSR_aa;
set QSR_blok;
true_val=5.4551;
diff_sq_P0 = (Par_0 - true_val)**2;
diff_sq_P1 = (Par_1 - true_val)**2;
diff_sq_P2 = (Par_2 - true_val)**2;
diff_sq_Mid1 = (midpoint1 - true_val)**2;
diff_sq_Mid2 = (midpoint2 - true_val)**2;
diff_sq_Rmid = (rand_midp - true_val)**2;
diff_sq_Rpar = (Rand_pareto - true_val)**2;
diff_sq_Mln = (logN_midp - true_val)**2;
diff_sq_Rln = (rand_ln - true_val)**2;
diff_sq_Raw = (raw_data - true_val)**2;
run;

proc sort data=QSR_aa; by blok;
run;

proc means data=QSR_aa mean noprint;
by blok;
var diff_sq_P0 diff_sq_P1 diff_sq_P2 diff_sq_Mid1 diff_sq_Mid2 diff_sq_Rmid
diff_sq_Rpar diff_sq_Mln
diff_sq_Rln diff_sq_Raw;
output out=QSR_aa_uit mean=Mdiff_sq_P0 Mdiff_sq_P1 Mdiff_sq_P2 Mdiff_sq_Mid1
Mdiff_sq_Mid2
Mdiff_sq_Rmid Mdiff_sq_Rpar Mdiff_sq_Mln Mdiff_sq_Rln Mdiff_sq_Raw;
run;

data QSR_RMSE_b (drop= _type_ _freq_);
set QSR_aa_uit;
P0_RMSE_b=sqrt(Mdiff_sq_P0);
P1_RMSE_b=sqrt(Mdiff_sq_P1);
P2_RMSE_b=sqrt(Mdiff_sq_P2);
Mid1_RMSE_b=sqrt(Mdiff_sq_Mid1);
Mid2_RMSE_b=sqrt(Mdiff_sq_Mid2);
Rmid_RMSE_b=sqrt(Mdiff_sq_Rmid);
Rpar_RMSE_b=sqrt(Mdiff_sq_Rpar);
Mln_RMSE_b=sqrt(Mdiff_sq_Mln);
Rln_RMSE_b=sqrt(Mdiff_sq_Rln);
raw_RMSE_b=sqrt(Mdiff_sq_Raw);
run;

proc means data=QSR_RMSE_b mean std noprint;
var P0_RMSE_b P1_RMSE_b P2_RMSE_b Mid1_RMSE_b Mid2_RMSE_b Rmid_RMSE_b
Rpar_RMSE_b Mln_RMSE_b Rln_RMSE_b
raw_RMSE_b;
output out=QSR_RMSE mean=P0_RMSE_m P1_RMSE_m P2_RMSE_m Mid1_RMSE_m
Mid2_RMSE_m Rmid_RMSE_m Rpar_RMSE_m
Mln_RMSE_m Rln_RMSE_m raw_RMSE_m
std=P0_RMSE_se P1_RMSE_se P2_RMSE_se Mid1_RMSE_se Mid2_RMSE_se Rmid_RMSE_se
Rpar_RMSE_se Mln_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

proc print data = QSR_RMSE;
var Mid1_RMSE_m Mid2_RMSE_m P0_RMSE_m P1_RMSE_m P2_RMSE_m Mln_RMSE_m
Rmid_RMSE_m Rpar_RMSE_m Rln_RMSE_m
raw_RMSE_m;
```

```
run;

proc print data = QSR_RMSE;
var Mid1_RMSE_se Mid2_RMSE_se P0_RMSE_se P1_RMSE_se P2_RMSE_se Mln_RMSE_se
Rmid_RMSE_se Rpar_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

****************************************************************************;
*Standard error of RMSE Theil;
data Theil_aa;
set Theil_blok;
true_val=0.2054;
diff_sq_P0 = (Par_0 - true_val)**2;
diff_sq_P1 = (Par_1 - true_val)**2;
diff_sq_P2 = (Par_2 - true_val)**2;
diff_sq_Mid1 = (midpoint1 - true_val)**2;
diff_sq_Mid2 = (midpoint2 - true_val)**2;
diff_sq_Rmid = (rand_midp - true_val)**2;
diff_sq_Rpar = (Rand_pareto - true_val)**2;
diff_sq_Mln = (logN_midp - true_val)**2;
diff_sq_Rln = (rand_ln - true_val)**2;
diff_sq_Raw = (raw_data - true_val)**2;
run;

proc sort data=Theil_aa; by blok;
run;

proc means data=Theil_aa mean noprint;
by blok;
var diff_sq_P0 diff_sq_P1 diff_sq_P2 diff_sq_Mid1 diff_sq_Mid2 diff_sq_Rmid
diff_sq_Rpar diff_sq_Mln
diff_sq_Rln diff_sq_Raw;
output out=Theil_aa_uit mean=Mdiff_sq_P0 Mdiff_sq_P1 Mdiff_sq_P2
Mdiff_sq_Mid1 Mdiff_sq_Mid2
Mdiff_sq_Rmid Mdiff_sq_Rpar Mdiff_sq_Mln Mdiff_sq_Rln Mdiff_sq_Raw;
run;

data Theil_RMSE_b (drop= _type_ _freq_);
set Theil_aa_uit;
P0_RMSE_b=sqrt(Mdiff_sq_P0);
P1_RMSE_b=sqrt(Mdiff_sq_P1);
P2_RMSE_b=sqrt(Mdiff_sq_P2);
Mid1_RMSE_b=sqrt(Mdiff_sq_Mid1);
Mid2_RMSE_b=sqrt(Mdiff_sq_Mid2);
Rmid_RMSE_b=sqrt(Mdiff_sq_Rmid);
Rpar_RMSE_b=sqrt(Mdiff_sq_Rpar);
Mln_RMSE_b=sqrt(Mdiff_sq_Mln);
Rln_RMSE_b=sqrt(Mdiff_sq_Rln);
raw_RMSE_b=sqrt(Mdiff_sq_Raw);
run;

proc means data=Theil_RMSE_b mean std noprint;
var P0_RMSE_b P1_RMSE_b P2_RMSE_b Mid1_RMSE_b Mid2_RMSE_b Rmid_RMSE_b
Rpar_RMSE_b Mln_RMSE_b Rln_RMSE_b
raw_RMSE_b;
output out=Theil_RMSE mean=P0_RMSE_m P1_RMSE_m P2_RMSE_m Mid1_RMSE_m
Mid2_RMSE_m Rmid_RMSE_m Rpar_RMSE_m
Mln_RMSE_m Rln_RMSE_m raw_RMSE_m
std=P0_RMSE_se P1_RMSE_se P2_RMSE_se Mid1_RMSE_se Mid2_RMSE_se Rmid_RMSE_se
Rpar_RMSE_se Mln_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;
```

```
proc print data = Theil_RMSE;
var Mid1_RMSE_m Mid2_RMSE_m P0_RMSE_m P1_RMSE_m P2_RMSE_m Mln_RMSE_m
Rmid_RMSE_m Rpar_RMSE_m Rln_RMSE_m
raw_RMSE_m;
run;

proc print data = Theil_RMSE;
var Mid1_RMSE_se Mid2_RMSE_se P0_RMSE_se P1_RMSE_se P2_RMSE_se Mln_RMSE_se
Rmid_RMSE_se Rpar_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

*****************************************************************************;
*Standard error of RMSE Atk;
data Atk_aa;
set Atk_blok;
true_val=0.0931;
diff_sq_P0 = (Par_0 - true_val)**2;
diff_sq_P1 = (Par_1 - true_val)**2;
diff_sq_P2 = (Par_2 - true_val)**2;
diff_sq_Mid1 = (midpoint1 - true_val)**2;
diff_sq_Mid2 = (midpoint2 - true_val)**2;
diff_sq_Rmid = (rand_midp - true_val)**2;
diff_sq_Rpar = (Rand_pareto - true_val)**2;
diff_sq_Mln = (logN_midp - true_val)**2;
diff_sq_Rln = (rand_ln - true_val)**2;
diff_sq_Raw = (raw_data - true_val)**2;
run;

proc sort data=Atk_aa; by blok;
run;

proc means data=Atk_aa mean noprint;
by blok;
var diff_sq_P0 diff_sq_P1 diff_sq_P2 diff_sq_Mid1 diff_sq_Mid2 diff_sq_Rmid
diff_sq_Rpar diff_sq_Mln
diff_sq_Rln diff_sq_Raw;
output out=Atk_aa_uit mean=Mdiff_sq_P0 Mdiff_sq_P1 Mdiff_sq_P2 Mdiff_sq_Mid1
Mdiff_sq_Mid2
Mdiff_sq_Rmid Mdiff_sq_Rpar Mdiff_sq_Mln Mdiff_sq_Rln Mdiff_sq_Raw;
run;

data Atk_RMSE_b (drop= _type_ _freq_);
set Atk_aa_uit;
P0_RMSE_b=sqrt(Mdiff_sq_P0);
P1_RMSE_b=sqrt(Mdiff_sq_P1);
P2_RMSE_b=sqrt(Mdiff_sq_P2);
Mid1_RMSE_b=sqrt(Mdiff_sq_Mid1);
Mid2_RMSE_b=sqrt(Mdiff_sq_Mid2);
Rmid_RMSE_b=sqrt(Mdiff_sq_Rmid);
Rpar_RMSE_b=sqrt(Mdiff_sq_Rpar);
Mln_RMSE_b=sqrt(Mdiff_sq_Mln);
Rln_RMSE_b=sqrt(Mdiff_sq_Rln);
raw_RMSE_b=sqrt(Mdiff_sq_Raw);
run;

proc means data=Atk_RMSE_b mean std noprint;
var P0_RMSE_b P1_RMSE_b P2_RMSE_b Mid1_RMSE_b Mid2_RMSE_b Rmid_RMSE_b
Rpar_RMSE_b Mln_RMSE_b Rln_RMSE_b
raw_RMSE_b;
output out=Atk_RMSE mean=P0_RMSE_m P1_RMSE_m P2_RMSE_m Mid1_RMSE_m
Mid2_RMSE_m Rmid_RMSE_m Rpar_RMSE_m
```

```
Mln_RMSE_m Rln_RMSE_m raw_RMSE_m
std=P0_RMSE_se P1_RMSE_se P2_RMSE_se Mid1_RMSE_se Mid2_RMSE_se Rmid_RMSE_se
Rpar_RMSE_se Mln_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

proc print data = Atk_RMSE;
var Mid1_RMSE_m Mid2_RMSE_m P0_RMSE_m P1_RMSE_m P2_RMSE_m Mln_RMSE_m
Rmid_RMSE_m Rpar_RMSE_m Rln_RMSE_m
raw_RMSE_m;
run;

proc print data = Atk_RMSE;
var Mid1_RMSE_se Mid2_RMSE_se P0_RMSE_se P1_RMSE_se P2_RMSE_se Mln_RMSE_se
Rmid_RMSE_se Rpar_RMSE_se
Rln_RMSE_se raw_RMSE_se;
run;

***************************************************************************;
***************************************************************************;
*Standard error of MAD Gini;
proc means data=Gini_blok n median noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BMed_gini median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

data Gini_Adif (drop = _type_ _freq_ Par_0 Par_1 Par_2 midpoint1 midpoint2
rand_midp Rand_pareto logN_midp rand_ln raw_data
BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp BM_rand_ln
BM_raw_data);
merge Gini_blok BMed_gini;
by blok;
ASB_dif_P0 = abs(Par_0-BM_Par_0);
ASB_dif_P1 = abs(Par_1-BM_Par_1);
ASB_dif_P2 = abs(Par_2-BM_Par_2);
ASB_dif_mid1 = abs(midpoint1-BM_midpoint1);
ASB_dif_mid2 = abs(midpoint2-BM_midpoint2);
ASB_dif_rmid = abs(rand_midp-BM_rand_midp);
ASB_dif_rpar = abs(Rand_pareto-BM_Rand_pareto);
ASB_dif_mln = abs(logN_midp-BM_logN_midp);
ASB_dif_rln = abs(rand_ln-BM_rand_ln);
ASB_dif_raw = abs(raw_data-BM_raw_data);
run;

proc sort data=Gini_Adif; by blok;
run;

proc means data = Gini_Adif n median noprint;
var ASB_dif_P0 ASB_dif_P1 ASB_dif_P2 ASB_dif_mid1 ASB_dif_mid2 ASB_dif_rmid
ASB_dif_rpar ASB_dif_mln
ASB_dif_rln ASB_dif_raw;
by blok;
output out=BMad_gini median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

proc means data=BMad_gini n mean std noprint;
```

```
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MAD_gini n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto M_logN_midp
Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = MAD_gini;
var M_midpoint1 M_midpoint2 M_Par_0 M_Par_1 M_Par_2 M_logN_midp M_rand_midp
M_Rand_pareto Mrand_ln Mraw_data;
run;

proc print data = MAD_gini;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

*****************************************************************************;
*Standard error of MAD QSR;
proc means data=QSR_blok n median noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BMed_QSR median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

data QSR_Adif (drop = _type_ _freq_ Par_0 Par_1 Par_2 midpoint1 midpoint2
rand_midp Rand_pareto logN_midp rand_ln raw_data
BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp BM_rand_ln
BM_raw_data);
merge QSR_blok BMed_QSR;
by blok;
ASB_dif_P0 = abs(Par_0-BM_Par_0);
ASB_dif_P1 = abs(Par_1-BM_Par_1);
ASB_dif_P2 = abs(Par_2-BM_Par_2);
ASB_dif_mid1 = abs(midpoint1-BM_midpoint1);
ASB_dif_mid2 = abs(midpoint2-BM_midpoint2);
ASB_dif_rmid = abs(rand_midp-BM_rand_midp);
ASB_dif_rpar = abs(Rand_pareto-BM_Rand_pareto);
ASB_dif_mln = abs(logN_midp-BM_logN_midp);
ASB_dif_rln = abs(rand_ln-BM_rand_ln);
ASB_dif_raw = abs(raw_data-BM_raw_data);
run;

proc sort data=QSR_Adif; by blok;
run;

proc means data = QSR_Adif n median noprint;
var ASB_dif_P0 ASB_dif_P1 ASB_dif_P2 ASB_dif_mid1 ASB_dif_mid2 ASB_dif_rmid
ASB_dif_rpar ASB_dif_mln
ASB_dif_rln ASB_dif_raw;
by blok;
output out=BMad_QSR median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;
```

```
proc means data=BMad_QSR n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MAD_QSR n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto M_logN_midp
Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = MAD_QSR;
var M_midpoint1 M_midpoint2 M_Par_0 M_Par_1 M_Par_2 M_logN_midp M_rand_midp
M_Rand_pareto Mrand_ln Mraw_data;
run;

proc print data = MAD_QSR;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

****************************************************************************;
*Standard error of MAD Theil;
proc means data=Theil_blok n median noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BMed_Theil median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

data Theil_Adif (drop = _type_ _freq_ Par_0 Par_1 Par_2 midpoint1 midpoint2
rand_midp Rand_pareto logN_midp rand_ln raw_data
BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp BM_rand_ln
BM_raw_data);
merge Theil_blok BMed_Theil;
by blok;
ASB_dif_P0 = abs(Par_0-BM_Par_0);
ASB_dif_P1 = abs(Par_1-BM_Par_1);
ASB_dif_P2 = abs(Par_2-BM_Par_2);
ASB_dif_mid1 = abs(midpoint1-BM_midpoint1);
ASB_dif_mid2 = abs(midpoint2-BM_midpoint2);
ASB_dif_rmid = abs(rand_midp-BM_rand_midp);
ASB_dif_rpar = abs(Rand_pareto-BM_Rand_pareto);
ASB_dif_mln = abs(logN_midp-BM_logN_midp);
ASB_dif_rln = abs(rand_ln-BM_rand_ln);
ASB_dif_raw = abs(raw_data-BM_raw_data);
run;

proc sort data=Theil_Adif; by blok;
run;

proc means data = Theil_Adif n median noprint;
var ASB_dif_P0 ASB_dif_P1 ASB_dif_P2 ASB_dif_mid1 ASB_dif_mid2 ASB_dif_rmid
ASB_dif_rpar ASB_dif_mln
ASB_dif_rln ASB_dif_raw;
by blok;
output out=BMad_Theil median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;
```

```
proc means data=BMad_Theil n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MAD_Theil n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1
M_midpoint2 M_rand_midp M_Rand_pareto M_logN_midp
Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = MAD_Theil;
var M_midpoint1 M_midpoint2 M_Par_0 M_Par_1 M_Par_2 M_logN_midp M_rand_midp
M_Rand_pareto Mrand_ln Mraw_data;
run;

proc print data = MAD_Theil;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

****************************************************************************;
*Standard error of MAD Atk;
proc means data=Atk_blok n median noprint;
var Par_0 Par_1 Par_2 midpoint1 midpoint2 rand_midp Rand_pareto logN_midp
rand_ln raw_data;
by blok;
output out=BMed_Atk median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
run;

data Atk_Adif (drop = _type_ _freq_ Par_0 Par_1 Par_2 midpoint1 midpoint2
rand_midp Rand_pareto logN_midp rand_ln raw_data
BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp BM_rand_ln
BM_raw_data);
merge Atk_blok BMed_Atk;
by blok;
ASB_dif_P0 = abs(Par_0-BM_Par_0);
ASB_dif_P1 = abs(Par_1-BM_Par_1);
ASB_dif_P2 = abs(Par_2-BM_Par_2);
ASB_dif_mid1 = abs(midpoint1-BM_midpoint1);
ASB_dif_mid2 = abs(midpoint2-BM_midpoint2);
ASB_dif_rmid = abs(rand_midp-BM_rand_midp);
ASB_dif_rpar = abs(Rand_pareto-BM_Rand_pareto);
ASB_dif_mln = abs(logN_midp-BM_logN_midp);
ASB_dif_rln = abs(rand_ln-BM_rand_ln);
ASB_dif_raw = abs(raw_data-BM_raw_data);
run;

proc sort data=Atk_Adif; by blok;
run;

proc means data = Atk_Adif n median noprint;
var ASB_dif_P0 ASB_dif_P1 ASB_dif_P2 ASB_dif_mid1 ASB_dif_mid2 ASB_dif_rmid
ASB_dif_rpar ASB_dif_mln
ASB_dif_rln ASB_dif_raw;
by blok;
output out=BMad_Atk median = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto
BM_logN_midp BM_rand_ln BM_raw_data;
```

```
run;

proc means data=BMad_Atk n mean std noprint;
var  BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MAD_Atk n=B mean= M_Par_0 M_Par_1 M_Par_2 M_midpoint1 M_midpoint2
M_rand_midp M_Rand_pareto M_logN_midp
Mrand_ln Mraw_data
std=S_Par_0 S_Par_1 S_Par_2 S_midpoint1 S_midpoint2 S_rand_midp S_Rand_pareto
S_logN_midp S_rand_ln S_raw_data;
run;

proc print data = MAD_Atk;
var M_midpoint1 M_midpoint2 M_Par_0 M_Par_1 M_Par_2 M_logN_midp M_rand_midp
M_Rand_pareto Mrand_ln Mraw_data;
run;

proc print data = MAD_Atk;
var S_midpoint1 S_midpoint2 S_Par_0 S_Par_1 S_Par_2 S_logN_midp S_rand_midp
S_Rand_pareto S_rand_ln S_raw_data;
run;

****************************************************************************;
****************************************************************************;
*MAD of MAD Gini;
*median of MAD's;
proc means data = BMad_gini n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMad_gini median = MM_Par_0 MM_Par_1 MM_Par_2 MM_midpoint1
MM_midpoint2 MM_rand_midp MM_Rand_pareto
MM_logN_midp MM_rand_ln MM_raw_data;
run;

data Gini_merge;
set BMad_gini;
if _N_=1 then set MMad_gini;
run;

data Gini_MM;
set Gini_merge;
P0_MM = abs(BM_Par_0-MM_Par_0);
P1_MM = abs(BM_Par_1-MM_Par_1);
P2_MM = abs(BM_Par_2-MM_Par_2);
mid1_MM = abs(BM_midpoint1-MM_midpoint1);
mid2_MM = abs(BM_midpoint2-MM_midpoint2);
RMid_MM = abs(BM_rand_midp-MM_rand_midp);
RPar_MM = abs(BM_Rand_pareto-MM_Rand_pareto);
LNMid_MM = abs(BM_logN_midp-MM_logN_midp);
RLN_MM = abs(BM_rand_ln-MM_rand_ln);
raw_MM = abs(BM_raw_data-MM_raw_data);
run;

proc means data = Gini_MM median noprint;
var P0_MM P1_MM P2_MM mid1_MM mid2_MM RMid_MM RPar_MM LNMid_MM RLN_MM raw_MM;
output out=MadMad_gini median = MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_midpoint1 MadM_midpoint2
MadM_rand_midp MadM_Rand_pareto MadM_logN_midp MadM_rand_ln MadM_raw_data;
run;

proc print data = MadMad_gini;
```

```
var MadM_midpoint1 MadM_midpoint2 MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_logN_midp MadM_rand_midp
MadM_Rand_pareto MadM_rand_ln MadM_raw_data;
run;

****************************************************************************;
*MAD of MAD QSR;
*median of MAD's;
proc means data = BMad_QSR n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMad_QSR median = MM_Par_0 MM_Par_1 MM_Par_2 MM_midpoint1
MM_midpoint2 MM_rand_midp MM_Rand_pareto
MM_logN_midp MM_rand_ln MM_raw_data;
run;

data QSR_merge;
set BMad_QSR;
if _N_=1 then set MMad_QSR;
run;

data QSR_MM;
set QSR_merge;
P0_MM = abs(BM_Par_0-MM_Par_0);
P1_MM = abs(BM_Par_1-MM_Par_1);
P2_MM = abs(BM_Par_2-MM_Par_2);
mid1_MM = abs(BM_midpoint1-MM_midpoint1);
mid2_MM = abs(BM_midpoint2-MM_midpoint2);
RMid_MM = abs(BM_rand_midp-MM_rand_midp);
RPar_MM = abs(BM_Rand_pareto-MM_Rand_pareto);
LNMid_MM = abs(BM_logN_midp-MM_logN_midp);
RLN_MM = abs(BM_rand_ln-MM_rand_ln);
raw_MM = abs(BM_raw_data-MM_raw_data);
run;

proc means data = QSR_MM median noprint;
var P0_MM P1_MM P2_MM mid1_MM mid2_MM RMid_MM RPar_MM LNMid_MM RLN_MM raw_MM;
output out=MadMad_QSR median = MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_midpoint1 MadM_midpoint2
MadM_rand_midp MadM_Rand_pareto MadM_logN_midp MadM_rand_ln MadM_raw_data;
run;

proc print data = MadMad_QSR;
var MadM_midpoint1 MadM_midpoint2 MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_logN_midp MadM_rand_midp
MadM_Rand_pareto MadM_rand_ln MadM_raw_data;
run;

****************************************************************************;
*MAD of MAD Theil;
*median of MAD's;
proc means data = BMad_Theil n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMad_Theil median = MM_Par_0 MM_Par_1 MM_Par_2 MM_midpoint1
MM_midpoint2 MM_rand_midp MM_Rand_pareto
MM_logN_midp MM_rand_ln MM_raw_data;
run;

data Theil_merge;
set BMad_Theil;
```

```
if _N_=1 then set MMad_Theil;
run;

data Theil_MM;
set Theil_merge;
P0_MM = abs(BM_Par_0-MM_Par_0);
P1_MM = abs(BM_Par_1-MM_Par_1);
P2_MM = abs(BM_Par_2-MM_Par_2);
mid1_MM = abs(BM_midpoint1-MM_midpoint1);
mid2_MM = abs(BM_midpoint2-MM_midpoint2);
RMid_MM = abs(BM_rand_midp-MM_rand_midp);
RPar_MM = abs(BM_Rand_pareto-MM_Rand_pareto);
LNMid_MM = abs(BM_logN_midp-MM_logN_midp);
RLN_MM = abs(BM_rand_ln-MM_rand_ln);
raw_MM = abs(BM_raw_data-MM_raw_data);
run;

proc means data = Theil_MM median noprint;
var P0_MM P1_MM P2_MM mid1_MM mid2_MM RMid_MM RPar_MM LNMid_MM RLN_MM raw_MM;
output out=MadMad_Theil median = MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_midpoint1 MadM_midpoint2
MadM_rand_midp MadM_Rand_pareto MadM_logN_midp MadM_rand_ln MadM_raw_data;
run;

proc print data = MadMad_Theil;
var MadM_midpoint1 MadM_midpoint2 MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_logN_midp MadM_rand_midp
MadM_Rand_pareto MadM_rand_ln MadM_raw_data;
run;

*****************************************************************************;
*MAD of MAD Atk;
*median of MAD's;
proc means data = BMad_Atk n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMad_Atk median = MM_Par_0 MM_Par_1 MM_Par_2 MM_midpoint1
MM_midpoint2 MM_rand_midp MM_Rand_pareto
MM_logN_midp MM_rand_ln MM_raw_data;
run;

data Atk_merge;
set BMad_Atk;
if _N_=1 then set MMad_Atk;
run;

data Atk_MM;
set Atk_merge;
P0_MM = abs(BM_Par_0-MM_Par_0);
P1_MM = abs(BM_Par_1-MM_Par_1);
P2_MM = abs(BM_Par_2-MM_Par_2);
mid1_MM = abs(BM_midpoint1-MM_midpoint1);
mid2_MM = abs(BM_midpoint2-MM_midpoint2);
RMid_MM = abs(BM_rand_midp-MM_rand_midp);
RPar_MM = abs(BM_Rand_pareto-MM_Rand_pareto);
LNMid_MM = abs(BM_logN_midp-MM_logN_midp);
RLN_MM = abs(BM_rand_ln-MM_rand_ln);
raw_MM = abs(BM_raw_data-MM_raw_data);
run;

proc means data = Atk_MM median noprint;
var P0_MM P1_MM P2_MM mid1_MM mid2_MM RMid_MM RPar_MM LNMid_MM RLN_MM raw_MM;
```

```
output out=MadMad_Atk median = MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_midpoint1 MadM_midpoint2
MadM_rand_midp MadM_Rand_pareto MadM_logN_midp MadM_rand_ln MadM_raw_data;
run;

proc print data = MadMad_Atk;
var MadM_midpoint1 MadM_midpoint2 MadM_Par_0 MadM_Par_1 MadM_Par_2
MadM_logN_midp MadM_rand_midp
MadM_Rand_pareto MadM_rand_ln MadM_raw_data;
run;

***************************************************************************;
***************************************************************************;
*Standard error of median for Gini;
*Mean of the median of blocks per method;
proc means data = BMed_gini n mean std noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MBMed_gini mean = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data
std = SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_midpoint1 SBM_midpoint2 SBM_rand_midp
SBM_Rand_pareto
SBM_logN_midp SBM_rand_ln SBM_raw_data;
run;

proc print data=MBMed_gini;
var SBM_midpoint1 SBM_midpoint2 SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_logN_midp
SBM_rand_midp SBM_Rand_pareto
SBM_rand_ln SBM_raw_data;
run;

***************************************************************************;
*Standard error of median for QSR;
*Mean of the median of blocks per method;
proc means data = BMed_QSR n mean std noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MBMed_QSR mean = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data
std = SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_midpoint1 SBM_midpoint2 SBM_rand_midp
SBM_Rand_pareto
SBM_logN_midp SBM_rand_ln SBM_raw_data;
run;

proc print data=MBMed_QSR;
var SBM_midpoint1 SBM_midpoint2 SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_logN_midp
SBM_rand_midp SBM_Rand_pareto
SBM_rand_ln SBM_raw_data;
run;

***************************************************************************;
*Standard error of median for Theil;
*Mean of the median of blocks per method;
proc means data = BMed_Theil n mean std noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MBMed_Theil mean = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
```

```
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data
std = SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_midpoint1 SBM_midpoint2 SBM_rand_midp
SBM_Rand_pareto
SBM_logN_midp SBM_rand_ln SBM_raw_data;
run;

proc print data=MBMed_Theil;
var SBM_midpoint1 SBM_midpoint2 SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_logN_midp
SBM_rand_midp SBM_Rand_pareto
SBM_rand_ln SBM_raw_data;
run;

****************************************************************************;
*Standard error of median for Atk;
*Mean of the median of blocks per method;
proc means data = BMed_Atk n mean std noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MBMed_Atk mean = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data
std = SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_midpoint1 SBM_midpoint2 SBM_rand_midp
SBM_Rand_pareto
SBM_logN_midp SBM_rand_ln SBM_raw_data;
run;

proc print data=MBMed_Atk;
var SBM_midpoint1 SBM_midpoint2 SBM_Par_0 SBM_Par_1 SBM_Par_2 SBM_logN_midp
SBM_rand_midp SBM_Rand_pareto
SBM_rand_ln SBM_raw_data;
run;

****************************************************************************;
****************************************************************************;
*MAD of median for Gini;
*Median of the median of blocks per method;
proc means data = BMed_gini n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMBMed_gini median = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data;
run;

data Gini_Medmerge;
set BMed_gini;
if _N_=1 then set MMBMed_gini;
run;

data Gini_absdev (drop = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2
BM_rand_midp BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data);
set Gini_Medmerge;
P0_absdev = abs(BM_Par_0-MBM_Par_0);
P1_absdev = abs(BM_Par_1-MBM_Par_1);
P2_absdev = abs(BM_Par_2-MBM_Par_2);
Mid1_absdev = abs(BM_midpoint1-MBM_midpoint1);
Mid2_absdev = abs(BM_midpoint2-MBM_midpoint2);
Rmid_absdev = abs(BM_rand_midp-MBM_rand_midp);
```

```
Rpar_absdev = abs(BM_Rand_pareto-MBM_Rand_pareto);
LNmid_absdev = abs(BM_logN_midp-MBM_logN_midp);
Rln_absdev = abs(BM_rand_ln-MBM_rand_ln);
raw_absdev = abs(BM_raw_data-MBM_raw_data);
run;

proc means data=Gini_absdev median noprint;
var P0_absdev P1_absdev P2_absdev Mid1_absdev Mid2_absdev Rmid_absdev
Rpar_absdev LNmid_absdev Rln_absdev
raw_absdev;
output out=Gini_Mabsdev median=P0_Mabsdev P1_Mabsdev P2_Mabsdev Mid1_Mabsdev
Mid2_Mabsdev Rmid_Mabsdev
Rpar_Mabsdev LNmid_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

proc print data=Gini_Mabsdev;
var Mid1_Mabsdev Mid2_Mabsdev P0_Mabsdev P1_Mabsdev P2_Mabsdev LNmid_Mabsdev
Rmid_Mabsdev
Rpar_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

******************************************************************************;
*MAD of median for QSR;
*Median of the median of blocks per method;
proc means data = BMed_QSR n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMBMed_QSR median = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data;
run;

data QSR_Medmerge;
set BMed_QSR;
if _N_=1 then set MMBMed_QSR;
run;

data QSR_absdev (drop = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2
BM_rand_midp BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data);
set QSR_Medmerge;
P0_absdev = abs(BM_Par_0-MBM_Par_0);
P1_absdev = abs(BM_Par_1-MBM_Par_1);
P2_absdev = abs(BM_Par_2-MBM_Par_2);
Mid1_absdev = abs(BM_midpoint1-MBM_midpoint1);
Mid2_absdev = abs(BM_midpoint2-MBM_midpoint2);
Rmid_absdev = abs(BM_rand_midp-MBM_rand_midp);
Rpar_absdev = abs(BM_Rand_pareto-MBM_Rand_pareto);
LNmid_absdev = abs(BM_logN_midp-MBM_logN_midp);
Rln_absdev = abs(BM_rand_ln-MBM_rand_ln);
raw_absdev = abs(BM_raw_data-MBM_raw_data);
run;

proc means data=QSR_absdev median noprint;
var P0_absdev P1_absdev P2_absdev Mid1_absdev Mid2_absdev Rmid_absdev
Rpar_absdev LNmid_absdev Rln_absdev
raw_absdev;
output out=QSR_Mabsdev median=P0_Mabsdev P1_Mabsdev P2_Mabsdev Mid1_Mabsdev
Mid2_Mabsdev Rmid_Mabsdev
Rpar_Mabsdev LNmid_Mabsdev Rln_Mabsdev raw_Mabsdev;
```

```
run;

proc print data=QSR_Mabsdev;
var Mid1_Mabsdev Mid2_Mabsdev P0_Mabsdev P1_Mabsdev P2_Mabsdev LNmid_Mabsdev
Rmid_Mabsdev
Rpar_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

***************************************************************************;
*MAD of median for Theil;
*Median of the median of blocks per method;
proc means data = BMed_Theil n median noprint;
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMBMed_Theil median = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data;
run;

data Theil_Medmerge;
set BMed_Theil;
if _N_=1 then set MMBMed_Theil;
run;

data Theil_absdev (drop = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1
BM_midpoint2 BM_rand_midp BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data);
set Theil_Medmerge;
P0_absdev = abs(BM_Par_0-MBM_Par_0);
P1_absdev = abs(BM_Par_1-MBM_Par_1);
P2_absdev = abs(BM_Par_2-MBM_Par_2);
Mid1_absdev = abs(BM_midpoint1-MBM_midpoint1);
Mid2_absdev = abs(BM_midpoint2-MBM_midpoint2);
Rmid_absdev = abs(BM_rand_midp-MBM_rand_midp);
Rpar_absdev = abs(BM_Rand_pareto-MBM_Rand_pareto);
LNmid_absdev = abs(BM_logN_midp-MBM_logN_midp);
Rln_absdev = abs(BM_rand_ln-MBM_rand_ln);
raw_absdev = abs(BM_raw_data-MBM_raw_data);
run;

proc means data=Theil_absdev median noprint;
var P0_absdev P1_absdev P2_absdev Mid1_absdev Mid2_absdev Rmid_absdev
Rpar_absdev LNmid_absdev Rln_absdev
raw_absdev;
output out=Theil_Mabsdev median=P0_Mabsdev P1_Mabsdev P2_Mabsdev Mid1_Mabsdev
Mid2_Mabsdev Rmid_Mabsdev
Rpar_Mabsdev LNmid_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

proc print data=Theil_Mabsdev;
var Mid1_Mabsdev Mid2_Mabsdev P0_Mabsdev P1_Mabsdev P2_Mabsdev LNmid_Mabsdev
Rmid_Mabsdev
Rpar_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

***************************************************************************;
*MAD of median for Atk;
*Median of the median of blocks per method;
proc means data = BMed_Atk n median noprint;
```

```
var BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2 BM_rand_midp
BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data;
output out=MMBMed_Atk median = MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data;
run;

data Atk_Medmerge;
set BMed_Atk;
if _N_=1 then set MMBMed_Atk;
run;

data Atk_absdev (drop = BM_Par_0 BM_Par_1 BM_Par_2 BM_midpoint1 BM_midpoint2
BM_rand_midp BM_Rand_pareto BM_logN_midp
BM_rand_ln BM_raw_data MBM_Par_0 MBM_Par_1 MBM_Par_2 MBM_midpoint1
MBM_midpoint2 MBM_rand_midp
MBM_Rand_pareto MBM_logN_midp MBM_rand_ln MBM_raw_data);
set Atk_Medmerge;
P0_absdev = abs(BM_Par_0-MBM_Par_0);
P1_absdev = abs(BM_Par_1-MBM_Par_1);
P2_absdev = abs(BM_Par_2-MBM_Par_2);
Mid1_absdev = abs(BM_midpoint1-MBM_midpoint1);
Mid2_absdev = abs(BM_midpoint2-MBM_midpoint2);
Rmid_absdev = abs(BM_rand_midp-MBM_rand_midp);
Rpar_absdev = abs(BM_Rand_pareto-MBM_Rand_pareto);
LNmid_absdev = abs(BM_logN_midp-MBM_logN_midp);
Rln_absdev = abs(BM_rand_ln-MBM_rand_ln);
raw_absdev = abs(BM_raw_data-MBM_raw_data);
run;

proc means data=Atk_absdev median noprint;
var P0_absdev P1_absdev P2_absdev Mid1_absdev Mid2_absdev Rmid_absdev
Rpar_absdev LNmid_absdev Rln_absdev
raw_absdev;
output out=Atk_Mabsdev median=P0_Mabsdev P1_Mabsdev P2_Mabsdev Mid1_Mabsdev
Mid2_Mabsdev Rmid_Mabsdev
Rpar_Mabsdev LNmid_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

proc print data=Atk_Mabsdev;
var Mid1_Mabsdev Mid2_Mabsdev P0_Mabsdev P1_Mabsdev P2_Mabsdev LNmid_Mabsdev
Rmid_Mabsdev
Rpar_Mabsdev Rln_Mabsdev raw_Mabsdev;
run;

*****************************************************************************;
*****************************************************************************;
*Gini - standard error for bais mean and median - se(mean_bias),
se(med_bias);
data Gini_bimean;
set BM_gini;
true_value=0.3333;
P0_BM=BM_Par_0-true_value;
P1_BM=BM_Par_1-true_value;
P2_BM=BM_Par_2-true_value;
Mid1_BM=BM_midpoint1-true_value;
Mid2_BM=BM_midpoint2-true_value;
Rmid_BM=BM_rand_midp-true_value;
Rpar_BM=BM_Rand_pareto-true_value;
LN_mid_BM=BM_logN_midp-true_value;
RLn_BM=BM_rand_ln-true_value;
raw_BM=BM_raw_data-true_value;
```

```
run;

proc means data=Gini_bimean std noprint;
var P0_BM P1_BM P2_BM Mid1_BM Mid2_BM Rmid_BM Rpar_BM LN_mid_BM RLn_BM
raw_BM;
output out= Gini_bimean_se std=P0_BMs P1_BMs P2_BMs Mid1_BMs Mid2_BMs
Rmid_BMs Rpar_BMs LN_mid_BMs RLn_BMs raw_BMs;
run;

proc print data=Gini_bimean_se;
var Mid1_BMs Mid2_BMs P0_BMs P1_BMs P2_BMs LN_mid_BMs Rmid_BMs Rpar_BMs
RLn_BMs raw_BMs;
run;

data Gini_bimed;
set BMed_gini;
true_value=0.3333;
P0_BMed=BM_Par_0-true_value;
P1_BMed=BM_Par_1-true_value;
P2_BMed=BM_Par_2-true_value;
Mid1_BMed=BM_midpoint1-true_value;
Mid2_BMed=BM_midpoint2-true_value;
Rmid_BMed=BM_rand_midp-true_value;
Rpar_BMed=BM_Rand_pareto-true_value;
LN_mid_BMed=BM_logN_midp-true_value;
RLn_BMed=BM_rand_ln-true_value;
raw_BMed=BM_raw_data-true_value;
run;

proc means data=Gini_bimed std noprint;
var P0_BMed P1_BMed P2_BMed Mid1_BMed Mid2_BMed Rmid_BMed Rpar_BMed
LN_mid_BMed RLn_BMed raw_BMed;
output out= Gini_bimed_se std=P0_BMeds P1_BMeds P2_BMeds Mid1_BMeds
Mid2_BMeds Rmid_BMeds Rpar_BMeds LN_mid_BMeds
RLn_BMeds raw_BMeds;
run;

proc print data=Gini_bimed_se;
var Mid1_BMeds Mid2_BMeds P0_BMeds P1_BMeds P2_BMeds LN_mid_BMeds Rmid_BMeds
Rpar_BMeds
RLn_BMeds raw_BMeds;
run;

***********************************************************************;
*QSR - standard error for bais mean and median - se(mean_bias), se(med_bias);
data QSR_bimean;
set BM_QSR;
true_value=5.4551;
P0_BM=BM_Par_0-true_value;
P1_BM=BM_Par_1-true_value;
P2_BM=BM_Par_2-true_value;
Mid1_BM=BM_midpoint1-true_value;
Mid2_BM=BM_midpoint2-true_value;
Rmid_BM=BM_rand_midp-true_value;
Rpar_BM=BM_Rand_pareto-true_value;
LN_mid_BM=BM_logN_midp-true_value;
RLn_BM=BM_rand_ln-true_value;
raw_BM=BM_raw_data-true_value;
run;

proc means data=QSR_bimean std noprint;
var P0_BM P1_BM P2_BM Mid1_BM Mid2_BM Rmid_BM Rpar_BM LN_mid_BM RLn_BM
raw_BM;
```

```
output out= QSR_bimean_se std=P0_BMs P1_BMs P2_BMs Mid1_BMs Mid2_BMs Rmid_BMs
Rpar_BMs LN_mid_BMs RLn_BMs raw_BMs;
run;

proc print data=QSR_bimean_se;
var Mid1_BMs Mid2_BMs P0_BMs P1_BMs P2_BMs LN_mid_BMs Rmid_BMs Rpar_BMs
RLn_BMs raw_BMs;
run;

data QSR_bimed;
set BMed_QSR;
true_value=5.4551;
P0_BMed=BM_Par_0-true_value;
P1_BMed=BM_Par_1-true_value;
P2_BMed=BM_Par_2-true_value;
Mid1_BMed=BM_midpoint1-true_value;
Mid2_BMed=BM_midpoint2-true_value;
Rmid_BMed=BM_rand_midp-true_value;
Rpar_BMed=BM_Rand_pareto-true_value;
LN_mid_BMed=BM_logN_midp-true_value;
RLn_BMed=BM_rand_ln-true_value;
raw_BMed=BM_raw_data-true_value;
run;

proc means data=QSR_bimed std noprint;
var P0_BMed P1_BMed P2_BMed Mid1_BMed Mid2_BMed Rmid_BMed Rpar_BMed
LN_mid_BMed RLn_BMed raw_BMed;
output out= QSR_bimed_se std=P0_BMeds P1_BMeds P2_BMeds Mid1_BMeds Mid2_BMeds
Rmid_BMeds Rpar_BMeds LN_mid_BMeds
RLn_BMeds raw_BMeds;
run;

proc print data=QSR_bimed_se;
var Mid1_BMeds Mid2_BMeds P0_BMeds P1_BMeds P2_BMeds LN_mid_BMeds Rmid_BMeds
Rpar_BMeds
RLn_BMeds raw_BMeds;
run;

*****************************************************************************;
*Theil - standard error for bais mean and median - se(mean_bias),
se(med_bias);
data Theil_bimean;
set BM_Theil;
true_value=0.2054;
P0_BM=BM_Par_0-true_value;
P1_BM=BM_Par_1-true_value;
P2_BM=BM_Par_2-true_value;
Mid1_BM=BM_midpoint1-true_value;
Mid2_BM=BM_midpoint2-true_value;
Rmid_BM=BM_rand_midp-true_value;
Rpar_BM=BM_Rand_pareto-true_value;
LN_mid_BM=BM_logN_midp-true_value;
RLn_BM=BM_rand_ln-true_value;
raw_BM=BM_raw_data-true_value;
run;

proc means data=Theil_bimean std noprint;
var P0_BM P1_BM P2_BM Mid1_BM Mid2_BM Rmid_BM Rpar_BM LN_mid_BM RLn_BM
raw_BM;
output out= Theil_bimean_se std=P0_BMs P1_BMs P2_BMs Mid1_BMs Mid2_BMs
Rmid_BMs Rpar_BMs LN_mid_BMs RLn_BMs raw_BMs;
run;
```

```
proc print data=Theil_bimean_se;
var Mid1_BMs Mid2_BMs P0_BMs P1_BMs P2_BMs LN_mid_BMs Rmid_BMs Rpar_BMs
RLn_BMs raw_BMs;
run;

data Theil_bimed;
set BMed_Theil;
true_value=0.2054;
P0_BMed=BM_Par_0-true_value;
P1_BMed=BM_Par_1-true_value;
P2_BMed=BM_Par_2-true_value;
Mid1_BMed=BM_midpoint1-true_value;
Mid2_BMed=BM_midpoint2-true_value;
Rmid_BMed=BM_rand_midp-true_value;
Rpar_BMed=BM_Rand_pareto-true_value;
LN_mid_BMed=BM_logN_midp-true_value;
RLn_BMed=BM_rand_ln-true_value;
raw_BMed=BM_raw_data-true_value;
run;

proc means data=Theil_bimed std noprint;
var P0_BMed P1_BMed P2_BMed Mid1_BMed Mid2_BMed Rmid_BMed Rpar_BMed
LN_mid_BMed RLn_BMed raw_BMed;
output out= Theil_bimed_se std=P0_BMeds P1_BMeds P2_BMeds Mid1_BMeds
Mid2_BMeds Rmid_BMeds Rpar_BMeds LN_mid_BMeds
RLn_BMeds raw_BMeds;
run;

proc print data=Theil_bimed_se;
var Mid1_BMeds Mid2_BMeds P0_BMeds P1_BMeds P2_BMeds LN_mid_BMeds Rmid_BMeds
Rpar_BMeds
RLn_BMeds raw_BMeds;
run;

***************************************************************************
;
*Atk - standard error for bais mean and median - se(mean_bias), se(med_bias);
data Atk_bimean;
set BM_Atk;
true_value=0.0931;
P0_BM=BM_Par_0-true_value;
P1_BM=BM_Par_1-true_value;
P2_BM=BM_Par_2-true_value;
Mid1_BM=BM_midpoint1-true_value;
Mid2_BM=BM_midpoint2-true_value;
Rmid_BM=BM_rand_midp-true_value;
Rpar_BM=BM_Rand_pareto-true_value;
LN_mid_BM=BM_logN_midp-true_value;
RLn_BM=BM_rand_ln-true_value;
raw_BM=BM_raw_data-true_value;
run;

proc means data=Atk_bimean std noprint;
var P0_BM P1_BM P2_BM Mid1_BM Mid2_BM Rmid_BM Rpar_BM LN_mid_BM RLn_BM
raw_BM;
output out= Atk_bimean_se std=P0_BMs P1_BMs P2_BMs Mid1_BMs Mid2_BMs Rmid_BMs
Rpar_BMs LN_mid_BMs RLn_BMs raw_BMs;
run;

proc print data=Atk_bimean_se;
var Mid1_BMs Mid2_BMs P0_BMs P1_BMs P2_BMs LN_mid_BMs Rmid_BMs Rpar_BMs
RLn_BMs raw_BMs;
run;
```

```
data Atk_bimed;
set BMed_Atk;
true_value=0.0931;
P0_BMed=BM_Par_0-true_value;
P1_BMed=BM_Par_1-true_value;
P2_BMed=BM_Par_2-true_value;
Mid1_BMed=BM_midpoint1-true_value;
Mid2_BMed=BM_midpoint2-true_value;
Rmid_BMed=BM_rand_midp-true_value;
Rpar_BMed=BM_Rand_pareto-true_value;
LN_mid_BMed=BM_logN_midp-true_value;
RLn_BMed=BM_rand_ln-true_value;
raw_BMed=BM_raw_data-true_value;
run;

proc means data=Atk_bimed std noprint;
var P0_BMed P1_BMed P2_BMed Mid1_BMed Mid2_BMed Rmid_BMed Rpar_BMed
LN_mid_BMed RLn_BMed raw_BMed;
output out= Atk_bimed_se std=P0_BMeds P1_BMeds P2_BMeds Mid1_BMeds Mid2_BMeds
Rmid_BMeds Rpar_BMeds LN_mid_BMeds
RLn_BMeds raw_BMeds;
run;

proc print data=Atk_bimed_se;
var Mid1_BMeds Mid2_BMeds P0_BMeds P1_BMeds P2_BMeds LN_mid_BMeds Rmid_BMeds
Rpar_BMeds
RLn_BMeds raw_BMeds;
run;
```