COMPARISON OF REGRESSION TECHNIQUES VIA MONTE CARLO SIMULATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OYA CAN MUTAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

IN

STATISTICS

JUNE 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. H. Öztaş Ayhan
Head of Department

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Birdal Şenoğlu
Co-Supervisor

Prof. Dr. Öztaş Ayhan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Ayşen Akkaya (METU, STAT)

Prof. Dr. H. Öztaş Ayhan (METU, STAT)

Asst. Prof. Dr. İnci Batmaz (METU, STAT)

Asst. Prof. Dr. Birdal Şenoğlu (OSMANGAZİ UNIV., STAT)

Dr. H. Savaş Sazak (METU, STAT)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Oya CAN MUTAN

ABSTRACT

COMPARISON OF REGRESSION TECHNIQUES
VIA MONTE CARLO SIMULATION

CAN MUTAN, Oya
M.S., Department of Statistics
Supervisor: Prof. Dr. H. Öztaş AYHAN
Co-Supervisor: Asst. Prof. Dr. Birdal ŞENOĞLU

June 2004, 88 pages

The ordinary least squares (OLS) is one of the most widely used methods for modelling the functional relationship between variables. However, this estimation procedure counts on some assumptions and the violation of these assumptions may lead to nonrobust estimates. In this study, the simple linear regression model is investigated for conditions in which the distribution of the error terms is Generalised Logistic. Some robust and nonparametric methods such as modified maximum likelihood (MML), least absolute deviations (LAD), Winsorized least squares, least trimmed squares (LTS), Theil and weighted Theil are compared via computer simulation. In order to evaluate the estimator performance, mean, variance, bias, mean square error (MSE) and relative mean square error (RMSE) are computed.

Key Words: Ordinary least squares (OLS), modified maximum likelihood (MML), least absolute deviation (LAD), Winsorized least squares, least trimmed

squares (LTS), Theil and weighted Theil regression, robust and nonparametric estimation, Generalised Logistic.

ÖZ

REGRESYON TEKNİKLERİNİN MONTE CARLO SİMULASYONU ARACILIĞIYLA KARŞILAŞTIRILMASI

CAN MUTAN, Oya
Yüksek Lisans, İstatistik Bölümü
Tez Yöneticisi: Prof. Dr. H. Öztaş AYHAN
Ortak Tez Yöneticisi: Yrd. Doç. Dr. Birdal ŞENOĞLU

Haziran 2004, 88 sayfa

En küçük kareler yöntemi, değişkenler arasındaki fonksiyonel ilşkiyi modellemede en çok kullanılan yöntemlerden biridir. Ancak, bu tahmin yöntemi bazı varsayımlara dayanır ve bu varsayımların ihlali sağlam (robust) olmayan tahminlere yol açabilir. Bu çalışmada, hata terimlerinin dağılımının Genelleştirilmiş Lojistik olduğu durumda, basit linear regresyon modeli incelenmiştir. Uyarlanmış en çok olabilirlik (MML), en küçük mutlak sapmalar (LAD), Winsorize edilmiş en küçük kareler, budanmış (trimmed) en küçük kareler, Theil ve ağırlıklı Theil gibi bazı sağlam ve parametrik olmayan yöntemler bilgisayar simulasyonu yardımıyla karşılaştırılmıştır. Tahmin edicilerin başarımlarını değerlendirmek için ortalama, varyans, yanlılık derecesi, ortalama karesel hata (mean square error) ve göreli ortalama karesel hata (relative mean square error) hesaplanmıştır.

Anahtar Kelimeler: En küçük kareler yöntemi (OLS), uyarlanmış en çok olabilirlik (MML), en küçük mutlak sapmalar (LAD), Winsorize edilmiş en küçük kareler, budanmış (trimmed) en küçük kareler (LTS), Theil ve ağırlıklı Theil regresyonu, sağlam (robust) ve parametrik olmayan tahmin yöntemleri, Genelleştirilmiş Lojistik.

To my beloved family for their everlasting support….

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. Öztaş Ayhan who gave me the possibility to complete this work.

I am deeply indebted to my co-supervisor Asst. Prof. Dr. Birdal Şenoğlu for his detailed, constructive guidance and stimulating support. He not only encouraged me to go ahead with my thesis but also answered all my questions patiently in every time of research which made him more than only an advisor to me.

I would also like to thank the Chairman of the Department, Prof. Dr. Öztaş Ayhan, on behalf of the Department of Statistics for providing me a comfortable working environment and necessary equipments to complete this thesis.

I also want to express my deep appreciation to Dr. Hakan Savaş Sazak for his full support, complete assistance and true friendship from the very first moment I have met him.

I owe my loving thanks to my understanding, harmonous roommate Pelin Özkan, who was just beside me each and every step of this study.

I am also grateful to my dearest friends M. Özgür Pekbaş, Burcu Çetinkaya, Fulya Gürsoy, Mine Şenoğuz, Deniz Kantarcı, M. Didem Avcıoğlu Aytürk and Dr. Barış Sürücü, who showed me that time and even far distances can strengthen a good friendship.

But most of all I would like to thank my mother and father for their sacrifising and endless love which can be replaced by nothing in the world. I also want to thank my sister Zeynep Pekcan and her husband Mert Pekcan, and the beloved members of my family Gürcan and Güngör Onan, Nebihe Ekni and Güler Mutan for their full support, trust and love.

Last but not least, I would like to give my special thanks to my dear husband, Serkan Mutan, who always encouraged me and let me feel that he would always stand by me.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Regression analysis, which is a simple method for investigating and modelling the functional relationship between variables, is used almost all fields, including engineering, economics, management, biological and social sciences. The introduction of this analysis was done by Sir Francis Galton around 1800. This method enables us to describe the behavior of a random variable of interest, which is called the dependent variable. Other variables that provide information about the behavior of the dependent variable are entered into the model as independent variables, and these are assumed to be known constants. In addition to these known constants, all regression equations contain unknown constants, called regression coefficients. Procedures applied by regression analysis try to draw conclusions about these coefficients, and understand the behavior of the model. The simplest linear regression model contains only one independent variable, and called simple linear regression model. This model is

$$y_i = \theta_0 + \theta_1 x_i + \varepsilon_i, \qquad 1 \le i \le n, \tag{1.1}$$

where the intercept θ_0 and the slope θ_1 are model regression parameters, and ε_i is a random error component. Customarily, x_i's are nonstochastic design values measured with negligible error and y_i's are dependent variables. The parameter θ_1 is equal to dy/dx, and expressed as the rate of change in y corresponding to a unit increase in x. It is assumed that the model (1.1) is an appropriate approximation to the true relation between y_i's and x_i's, and ε_i's measure the gap in that approximation.

In this thesis, we are interested in simple linear regression model where the independent variables, x_i's are assumed to be nonstochastic design values. However, different from the standard theory assumption that the error terms come from normal distribution, we deal with the case where the error terms have Generalised Logistic distribution. Then, the probability density function of the error terms is

$$f(\varepsilon) = \frac{b}{\sigma} \frac{\exp(-\varepsilon/\sigma)}{\{1+\exp(-\varepsilon/\sigma)\}^{b+1}}, \qquad -\infty < \varepsilon < \infty. \tag{1.2}$$

Note that for the cases, b<1, b=1 and b>1, Generalised Logistic distribution represents negatively skewed, symmetric and positively skewed distributions, respectively. Therefore, this distribution has a broad application area in real life.

The slope and y-intercept coefficients, $\hat{\theta}_1$ and $\hat{\theta}_0$, for the model (1.1) are estimated by using the ordinary least squares (OLS), modified maximum likelihood (MML), least absolute deviation (LAD), winsorized least squares, least trimmed squares (LTS), Theil's and weighted Theil's estimation methods. Chapter 2 covers brief information and review of literature for these methods. Since OLS is one of the most widely used regression techniques, we decide to explain this methodology in Chapter 3, seperately. Also, this chapter includes an application of this procedure to a real life data set. The alternative regression methods, which are needed because of the inappropriateness of the OLS estimation in many nonideal conditions, are discussed in Chapter 4. Each technique is applied to real life data sets. The simulation study, which is expressed in Chapter 5, is performed for four different sample sizes (n=10, 20 ,50 and 100) with ten types of error distributions.

In this study, our objective is to compare these eight regression techniques via Monte Carlo simulation for skewed distributions which has not done before. The performance of estimators are evaluated with respect to their mean, variance, bias, mean square error (MSE) and relative mean square error (RMSE).

CHAPTER 2

LITERATURE SURVEY

There are various methods one could use to construct a linear regression model and estimate the regression coefficients. One of the most popular methods is the OLS estimation procedure. It was discovered by Carl Friedrich Gauss in Germany and Adrien Marie Legendre in France around 1805. Early applications of the method were to astronomic and geodetic data. It was first published in 1805 as an appendix to a book by Legendre on determining the orbits of comets (Birkes and Dodge, 1993; Myers, 1986). The purpose of this procedure is to optimize fit by minimizing

$$S(\theta_0,\theta_1)=\sum_{i=1}^{n}(y_i-\theta_0-\theta_1 x_i)^2 . \tag{2.1}$$

Rousseeuw and Leroy (1987) explain why OLS method has been a powerful research tool as follows: At the time of its invention, there were no computers. Since OLS estimator could be calculated explicitly from the data, it became the only possible approach and also, at that time it was the most reasonable location estimator. After Gauss had introduced the normal (Gaussian) distribution as the error distribution for which the OLS is optimal, beautiful mathematical results are obtained. Henceforth, both the Gaussian assumptions and OLS has become the cornerstone of statistics. Even now, this procedure is still used because of its traditon and ease of computation.

More recently, there has been increasing interest in other methods by realizing that it is very diffucult for a real life data to satisfy the necessary assumptions. Another reason for that is the advances in computer technology, which have decreased the computational diffuculties of other methods. These alternative methods have considerable advantages over OLS estimation procedure and practical to use.

Nearly 50 years before the method of OLS, in 1757 Roger Joseph Boscovich constructed a method to estimate the shape of the earth and to deal with the inconsistent measurements in the sample data (Birkes and Dodge, 1993). Improving the Boscovich's method, Edgeworth devised a method in 1887. He said that since the residuals are squared in equation (1.2), the outliers have a large effect on OLS estimators. Hence, he suggested the LAD regression. The LAD estimates are the values of θ_0 and θ_1 that minimize

$$\sum_{i=1}^{n} \left| y_i - (\theta_0 - \theta_1 x_i) \right| . \tag{2.2}$$

This equation (2.2) had also used by Laplace for one-dimensional observations and the corresponding error law is now called the double exponential or Laplace distribution (Birkes and Dodge, 1993; Rousseeuw and Leroy, 1987).

Castillo et. al. (2001) investigate the influence of observations on LAD estimates and they recommend methods for regression diagnostics which are illustrated by numeric and real life examples. Mathew and Nordstrom (1993) propose to estimate the regression coefficients by minimizing the maximum of a weighted sum of squared deviations, or the sum of absolute deviations. Their aim is to establish robustness property of the LAD criterion. In another study, performed by Lawrence and Shier (1981), the methods of OLS and LAD are compared for estimating the Weibull parameters. Brown (1980) concentrates on the case of simple linear regression for LAD estimates and show how this method works in small or moderate samples. Also, Koenker and Bassett (1978), who investigate LAD estimates in great detail, achieve many interesting properties including asymptotic distributions.

Theil's regression method is a nonparametric procedure which is expected to perform well without regard to the distribution of the error terms. This procedure is based on ranks and uses the median as robust measures rather than using the mean as in OLS. To calculate the slope of a line that fits the data points, the slopes of all pairs of data points are computed and the median of all these pairwise slopes is expressed as the Theil's slope estimator, $\hat{\theta}_{1(THL)}$. The median of the $y_i - \hat{\theta}_{1(THL)} x_i$ terms is the estimator for the y-intercept of the regression line passing through all n observations (Birkes and Dodge, 1993; Nevitt and Tam, 1998).

Weighted Theil's regression method is a modified version of the Theil's original method to calculate the slope of the regression line. Nevitt and Tam (1998) say that in this method, each of the pairwise slopes is weighted using a weighting scheme, w_{ij}, where w_{ij} is equal to x_j-x_i, j-i or $|x_j$-$x_i|$. The median of these weighted pairwise slopes is then called as the weighted Theil's slope estimator. Also, the y-intercept estimator is calculated in a similar fashion as it is done in Theil's original method, but using $\hat{\theta}_{1(wtd.THL)}$ instead of $\hat{\theta}_{1(THL)}$.

Theil's and weighted Theil's regression methods are discussed in Hussain and Sprent (1983). They make simulation studies in which several methods are compared. They apply these techniques to both bivariate and multiple regression and try to show that the nonparametric slope estimators are robust. They state that during the extension of certain methods to multiple regression, new problems are faced, some of which are discussed in their paper. For simple linear regression, Guo (2000) extends these methods and presents a robust estimator to estimate the parameter of a first order autoregressive process under different distributions. In his simulation study, he compares Theil's and Hussain's estimator, the OLS estimator, and the proposed estimator. Wilcox (1998) shows that for simple linear regression, Theil estimator has high small-sample efficiency compared to the OLS estimator when the error term is heteroscedastic. In his paper, Wilcox considers four extensions of this estimator to two regressors, one of which is found to have advantages over the other three. Talwar (1993) makes a simulation study in which the performance of the Theil type estimators considered. In the simulation study, the distributions of the error terms are chosen as: standard normal, "t" with 4 degrees of freedom, symmetric stable and chi-square with 4 degrees of freedom. Moreover, he studies some heteroscedastic models.

For the modified (weighted) Theil's method, Jaeckel (1972) uses the x-distance between the ith and jth observations as a weighting scheme (i.e. $w_{ij}=x_j-x_i$), while Birkes and Dodge (1993) weights each pairwise slope by $w_{ij} = |x_j-x_i|$. Scholz (1978) uses $w_{ij}=(j-i)$, which is the number of steps between ith and jth observations.

MML estimation procedure was introduced by Tiku (1967). The MML estimators that have explicit algebraic forms are remarkably efficient for the cases even when the error terms have a skewed distribution. This method gives small weights to extreme values leading to control the intractable terms and achieve robustness properties.

Islam et. al. (2001) use the method of MML in a simple linear regression model. For two families of skew distributions, (i) Weibull with support IR: $(0, \infty)$ and (ii) Generalised Logistic with support IR: $(-\infty, \infty)$, they derive MML estimators of parameters and compare these with OLS estimators. They show that these estimators are remarkably efficient and robust. Also, Tiku et. al. (2001) develop MML procedure for both short and long-tailed symmetric distributions in the context of simple linear regression.

Winsorized regression is another robust regression technique involving iterative modification of the sample data. The aim of Winsorization is to diminish the effect of contamination on the estimators by reducing the effect of outlier in the sample. This method is also based on the OLS estimation procedure and apply smoothing techniques to decrease the influence of outlying observations. The most widely used smoothing technique is a process in which a specified percentage of the response variable, at each extreme of the ordered residuals, is modified iteratively. Iterations cover computing OLS estimates, obtaining residuals and replacing extreme observations with the modified ones (Yale and Forsythe, 1976; Nevitt and Tam, 1998).

Yale and Forsythe (1976) use this technique to estimate a simple linear regression model. Then, they compare this method with OLS through relative efficiency measurements got from Monte Carlo samples and they also apply this method to a real life data set. This procedure is also carried out by Rivest (1994) for various skewed distributions. Also, in order to remove the high influence of outlying observations, Hoo et. al. (2002) develop this approach, discuss the concept of robust statistics and present the procedures for robust multivariate outlier filtering. One simulated and two industrial examples are used to exhibit this procedure.

Another form of robust regression is the method of LTS which is introduced by Rousseeuw in 1984. The LTS equation is similar to the equation (2.1). The only difference is that in LTS estimation procedure, the largest squared residuals are not used in the summation, therefore the fit is not so much affected from the outliers. First, the cases corresponding to a specified percentage of the largest residuals under an initial OLS estimation are deleted. After removing these observations, OLS estimation is applied to the rest of the data and the resulting estimators are LTS estimates of slope and y-intercept (Rousseeuw and Leroy, 1987; Nevitt and Tam, 1998).

Agullo (2001) proposes two algorithms to compute these estimators. He discusses the implementation of these algorithms using orthogonal decomposition procedures and suggests several accelerations. Then, he applies these algorithms to real and simulated data sets, the results of which show that these new techniques significantly reduce the computational cost. Similarly, Hossjer (1995) describes an algorithm for computing the LTS estimate exactly in simple linear regression. Then, he presents some numerical examples to compare the exact algorithm, which can easily be extended to nonlinear regression, with approximate ones. Another study is performed by Zaman et. al. (2001). They state that robust regression techniques are rarely used in applied econometrics and that is why they apply this method to three sets of economic data. Then, they compare the results with OLS.

In order to estimate regression coefficients for simple linear regression model, Nevitt and Tam (1998) make a simulation study for conditions in which the normality assumption of the error terms is not satisfied. They compare some robust and nonparametrical methods such as LAD, LTS, Winsorized least squares, Theil, modified Theil and etc. In their simulation study, they use three sample sizes n=10, 30, 50 for different types of error distributions (unit normal, 10% contaminated normal, 30% contaminated normal, lognormal, t-5df). They evaluate the performance of these estimators with respect to the variance, bias, mean square error and relative mean square errors.

CHAPTER 3

ORDINARY LEAST SQUARES REGRESSION

One of the most popular methods to model the functional relationship between variables is the OLS estimation procedure which is very simple and straightforward to apply. However, for OLS estimators to be ideal some conditions are needed. One of these conditions is that the error terms, ε_i's are assumed to be independently, identically distributed (iid) random variables with mean zero and a constant variance σ^2.

Using sample data, the unknown parameters θ_0 and θ_1 are estimated by OLS procedure. This procedure involves minimizing the sum of squared deviations between the observed y_i and the estimates of their true means. In other words, the OLS method estimates the parameters θ_0 and θ_1 that minimize the sum of squares of the residuals $S(\theta_0, \theta_1)$, which is given in equation (2.1).

Taking the partial derivative of (2.1) with respect to θ_0 and θ_1, and equating to zero we get the OLS normal equations, which are given as,

$$n\hat{\theta}_0 + \hat{\theta}_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$\hat{\theta}_0 \sum_{i=1}^{n} x_i + \hat{\theta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i. \tag{3.1}$$

The values of $\hat{\theta}_0$ and $\hat{\theta}_1$ that satisfy these equations are given by

$$\hat{\theta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

and (3.2)

$$\hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x} .$$

Under the previously described assumptions that x_i's are nonstochastic, $E(\varepsilon_i)=0$ and $E(\varepsilon_i^2)=\sigma^2$, the estimators are unbiased (Myers, 1986), and their variances are

$$\mathrm{var}(\hat{\theta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{3.3}$$

and

$$\mathrm{var}(\hat{\theta}_0) = \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]. \tag{3.4}$$

The estimate of σ^2 gives us idea about the quality of fit and it is calculated from the error sum of squares,

$$SS(error) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 , \tag{3.5}$$

where $\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_i$ and the estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right)^2}{n-2}. \tag{3.6}$$

The degrees of freedom for the error sum of squares is the number of observations minus the number of estimated parameters. (For simple linear regression; intercept estimator $\hat{\theta}_0$ and slope estimator $\hat{\theta}_1$). Since $\hat{\sigma}^2$ is associated with the error sum of squares, any violation of the assumptions on ε_i's or misspecification of the model has serious consequences. It may cause $\hat{\sigma}^2$ to be a poor estimator of σ^2 or may yield an unstable model (Montgomery and Peck, 1991).

So far we have not made any distribution assumptions. However, for testing hypothesis and constructing confidence intervals, one more additional assumption is necessary: the random errors, ε_i's are assumed to be normally and independently distributed which leads to normally distributed response variables, y_i's (Rawlings et. al., 1998). The properties of OLS estimators are stronger under the normality assumption. If the error terms are normal and iid with zero mean and constant variance σ^2, then the OLS estimators of θ_0 and θ_1 attain uniformly minimum variance in the range of all unbiased estimators. Under these assumptions the likelihood function of ε_i's is,

$$L \propto \left(\frac{1}{\sigma}\right)^n \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2\right\}, \tag{3.7}$$

and the Fisher information matrix $I(\theta_0, \theta_1, \sigma)$, whose components are $-E\left(\partial^2 \ln L / \partial\theta_0^2\right)$, $-E\left(\partial^2 \ln L / \partial\theta_0 \partial\theta_1\right)$, $-E\left(\partial^2 \ln L / \partial\theta_0 \partial\sigma\right)$, $-E\left(\partial^2 \ln L / \partial\theta_1^2\right)$, $-E\left(\partial^2 \ln L / \partial\theta_1 \partial\sigma\right)$, and $-E\left(\partial^2 \ln L / \partial\sigma^2\right)$ is

$$I = \frac{n}{\sigma^2}\begin{bmatrix} 1 & \bar{x} & 0 \\ \bar{x} & (1/n)\sum_{i=1}^{n} x_i^2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{3.8}$$

The diagonal elements of the $I^{-1}(\theta_0, \theta_1, \sigma)$ matrix, I_{11}^{-1} and I_{22}^{-1}, give the asymptotic variances of $\hat{\theta}_0$ and $\hat{\theta}_1$, which are equal to the equations (3.5) and (3.4), respectively.

Under normality, the solutions of the equations given below are the maximum likelihood (ML) estimators which are also equal to the OLS estimators,

$$\frac{\partial \ln L}{\partial \theta_0} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i) = 0 \tag{3.9}$$

$$\frac{\partial \ln L}{\partial \theta_1} = \frac{1}{\sigma^2}(y_i - \theta_0 - \theta_1 x_i)x_i = 0 \tag{3.10}$$

$$\frac{\partial \ln L}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2 = 0\,. \tag{3.11}$$

While the violation of the assumptions lead to undesirable and unreliable results they should be checked. These assumptions and several types of model deficiencies can be detected with the help of the residual analysis. A residual, which can be expressed as the variability not explained by the regression model, is the discrepancy between the response and the fit.

Suppose that the distribution of the errors is not normal. If the errors are coming from a population that has a mean of 0, then the OLS estimates may not be optimal, but they at least have the property of being unbiased. If we further assume that the variance of the error population is finite, then the OLS estimates have the property of being consistent and asymptotically normal. However, under these conditions, the OLS estimates and tests may lose much of their efficiency and they can result in poor performance. To deal with these situations, two approaches can be applied. One is to try to correct nonnormality, if nonnormality is determined and the other is to use alternative regression methods, which do not depend on the assumption of the normality (Birkes and Dodge, 1993).

Stewart and Gill (1998) conducted a study to find the relationship between real consumers expenditure (y) and the real disposable income (x). They use 28 observations from 1963 to 1990. The data points are plotted in Figure 3.1 from which we can see that there is a linear relation between x and y.

Table 3.1 Data Set for consumers' expenditure and income, UK, 1963-1990

year	y_i	x_i	year	y_i	x_i
1963	170.874	185.426	1977	224.892	247.695
1964	176.044	193.247	1978	236.909	265.925
1965	178.493	196.998	1979	247.212	281.084
1966	181.550	201.207	1980	247.185	285.411
1967	185.985	204.171	1981	247.402	283.176
1968	191.209	207.772	1982	249.852	281.722
1969	192.366	209.684	1983	261.200	289.204
1970	197.873	217.675	1984	266.486	299.756
1971	204.139	220.344	1985	276.742	309.821
1972	216.752	238.744	1986	295.622	323.622
1973	228.615	254.329	1987	311.234	334.702
1974	225.317	252.360	1988	334.591	354.627
1975	224.580	253.814	1989	345.406	371.676
1976	225.666	253.012	1990	347.527	378.325

y (billion pounds, 1990 prices)

x (billion pounds, 1990 prices)

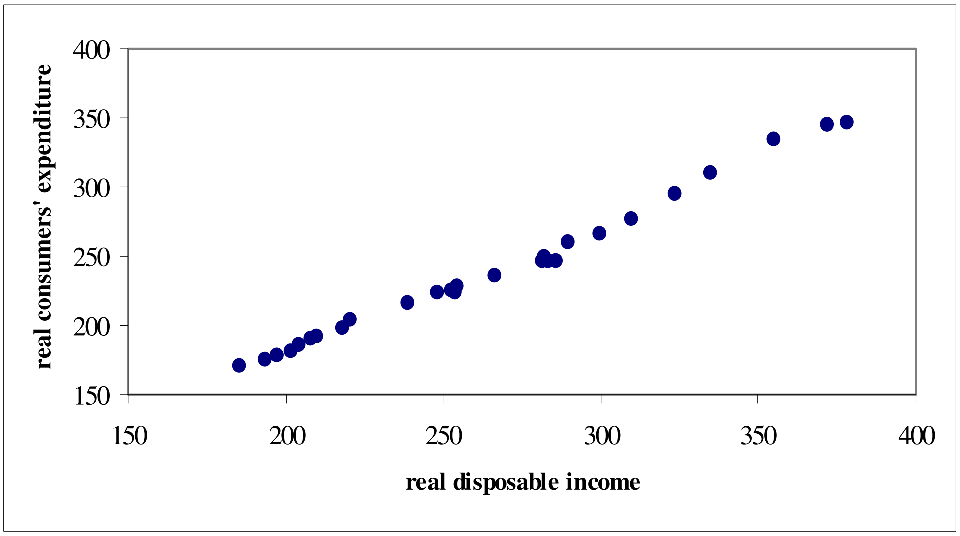


Figure 3.1 Plot of the consumers' expenditure and income, UK, 1963-1990

From equation (3.2), $\hat{\theta}_1$ and $\hat{\theta}_0$ are calculated as,

$\hat{\theta}_1$ =0.926319, and $\hat{\theta}_0$=-5.67485.

Therefore, the estimated regression equation is

$$\hat{y}_i = -5.67485 + 0.926319 x_i, \qquad i = 1, 2, \ldots, 28,$$

with $\hat{V}\left(\hat{\theta}_0\right) = 26.28202$ and $\hat{V}\left(\hat{\theta}_1\right) = 0.0003616$, which are obtained from equations (3.3) and (3.4).

CHAPTER 4

ALTERNATIVE REGRESSION TECHNIQUES

When the response variable y comes from the normal distribution, OLS method suits well and the estimators of θ_0 and θ_1 have good properties as explained in Chapter 3. However, in real life it is very diffucult to find such a data set that satisfies all the assumptions necessary to apply this method. OLS has a 0% breakdown value, which means that a small percentage of contamination can cause the estimators to take values from $-\infty$ to $+\infty$ (Rousseeuw and Leroy, 1987) and also, when higher dimensional data are investigated the outlying points may not be discovered. Hence, if the observations are not normally distributed or they contain outliers, the OLS method is no longer convenient. That is why robust regression procedures are needed to remove the adverse effect of these situations. These methods are insensitive to changes in a small percentage of the observations. An estimator is said to be robust if it is fully efficient (nearly so) under an assumed model but maintains high efficiency for possible alternatives.

In this thesis, we investigate some alternative estimation procedures such as MML, LAD, Winsorized least squares (10% and 20% smoothed), LTS (20% trimmed), Theil and weighted Theil which will be briefly explained below.

4.1 Modified Maximum Likelihood

For model (1.1), Islam et. al. (2001) consider the situation that error terms are iid and come from the Generalised Logistic distribution (b>0), which is given in equation (1.2)

The likelihood function for a random sample coming from $GL(b,\sigma)$ is

$$L = \left(\frac{b}{\sigma}\right)^n \frac{e^{-\sum_{i=1}^{n} z_i}}{\prod_{i=1}^{n} \left\{1 + e^{-z_i}\right\}^{b+1}}, \tag{4.1.1}$$

where $z_i = \dfrac{\varepsilon_i}{\sigma} = \dfrac{y_i - \theta_0 - \theta_1 x_i}{\sigma}$.

To get the maximum likelihood (ML) estimators of θ_0, θ_1 and σ, we should take the derivatives of log likelihood function with respect to these parameters and equate them to zero. The resulting likelihood equations are

$$\frac{\partial \ln L}{\partial \theta_0} = \frac{n}{\sigma} - \frac{(b+1)}{\sigma} \sum_{i=1}^{n} g(z_i) = 0 \tag{4.1.2}$$

$$\frac{\partial \ln L}{\partial \theta_1} = \frac{1}{\sigma} \sum_{i=1}^{n} x_i - \frac{(b+1)}{\sigma} \sum_{i=1}^{n} x_i g(z_i) = 0 \tag{4.1.3}$$

$$\frac{\partial \ln L}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma} \sum_{i=1}^{n} z_i - \frac{(b+1)}{\sigma} \sum_{i=1}^{n} z_i g(z_i) = 0, \tag{4.1.4}$$

where $g(z) = e^{-z} / \left(1 + e^{-z}\right)$.

The ML estimators which are the solutions to these equations are diffucult to derive. These equations have no explicit solutions. Therefore, iterative methods should be used. However, iterations can be problematic because of

(i) multiple roots,
(ii) convergence to wrong values,

(iii) nonconvergence of iterations.

These have been demonstrated by Barnett (1966), Lee et al. (1980), Vaughan (1992). Puthenpura and Sinha (1986) investigated the situations where the sample contains outliers, and they showed that for those conditions iterations with likelihood equations might not converge at all. To alleviate the diffuculties arising from the likelihood methodology, the method of MML is used (Tiku 1967; 1968; 1980; Tiku and Suresh, 1992). While applying MML method, first the likelihood equations are expressed in terms of order statistics, then by using the first two terms of a Taylor series expansion the intractable terms in the likelihood equations are linearized, and solving the resulting equations give us the MML estimators. The properties of the MML estimators are

(i) The MML estimators are asymptotically fully efficient. In other words, asymptotically, they are unbiased and minimum variance bound estimators (Vaughan and Tiku, 2000).

(ii) The MML estimators have no or negligible bias for small samples and are highly efficient (Lee et al. 1980; Vaughan 1992, Tiku et al. 1986).

(iii) The MML estimators which are explicit functions of the sample observations are easy to compute (Vaughan and Tiku, 2000).

To derive the modified likelihood equations, for a given θ_1, first $w_i=y_i-\theta_1 x_i$ values are ordered ($1 \le i \le n$).

$$w_{(1)} \le w_{(2)} \le ... \le w_{(n)}, \quad w_{(i)} = y_{[i]} - \theta_1 x_{[i]}. \tag{4.1.5}$$

Then, order variates $z_{(i)}$'s are calculated, where $z_{(i)} = \{w_{(i)} - \theta_0\}/\sigma$. ($y_{[i]}$, $x_{[i]}$) are concomitants of $z_{(i)}$ and is that pair of (y_j,x_j) values which determines $w_{(i)}$. As complete sums are invariant to ordering, the likelihood equations (4.1.2)-(4.1.4) can be expressed as,

$$\frac{\partial \ln L}{\partial \theta_0} = \frac{n}{\sigma} - \frac{(b+1)}{\sigma}\sum_{i=1}^{n} g\left(z_{(i)}\right) = 0 \tag{4.1.6}$$

$$\frac{\partial \ln L}{\partial \theta_1} = \frac{1}{\sigma}\sum_{i=1}^{n} x[i] - \frac{(b+1)}{\sigma}\sum_{i=1}^{n} x[i] g\left(z_{(i)}\right) = 0 \tag{4.1.7}$$

$$\frac{\partial \ln L}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma}\sum_{i=1}^{n} z_{(i)} - \frac{(b+1)}{\sigma}\sum_{i=1}^{n} z_{(i)} g\left(z_{(i)}\right) = 0\,. \tag{4.1.8}$$

As it is stated in Islam et al. (2001), since the function $g(z_{(i)})$ is almost linear in a small interval of $z_{(i)}$, it can be linearized by using the first two terms of a taylor series expansion around $t_{(i)}$, where $t_{(i)}$ is the expected value of ith order variate, $t_{(i)}=E(z_{(i)})$, $(1 \le i \le n)$.

$$g\left(z_{(i)}\right) \cong g\left(t_{(i)}\right) + \left(z_{(i)} - t_{(i)}\right)\left[\frac{d}{dz} g(z)\right]_{z=t_{(i)}} \tag{4.1.9}$$

$$= \alpha_i - \beta_i z_{(i)},$$

where

$$\alpha_{(i)} = \frac{1 + e^{t_{(i)}} + t_{(i)} e^{t_{(i)}}}{\left(1 + e^{t_{(i)}}\right)^2}, \quad \text{and} \quad \beta_i = \frac{e^{t_{(i)}}}{\left(1 + e^{t_{(i)}}\right)^2}\,. \tag{4.1.10}$$

β_i's are the weight coefficients. By giving small weights to extreme values they lead MML estimators to achieve robustness properties. The approximate values of $t_{(i)}$, which is the expected value of ith order variate, can be obtained from the equations,

$$F\left(t_{(i)}\right) = \int_{-\infty}^{t_{(i)}} \frac{be^{-z}}{\left(1 + e^{-z}\right)^{(b+1)}}\, dz = \frac{i}{n+1}, \tag{4.1.11}$$

where

$$t_{(i)} = -\ln\left(q_i^{-1/b} - 1\right), \quad q_i = \frac{i}{n+1}. \tag{4.1.12}$$

By placing (4.1.9) in the likelihood equations (4.1.6)-(4.1.8), we get the modified likelihood equations,

$$\frac{\partial \ln L}{\partial \theta_0} \cong \frac{\partial \ln L^*}{\partial \theta_0} = \frac{n}{\sigma} - \frac{(b+1)}{\sigma}\sum_{i=1}^{n}\left\{\alpha_i - \beta_i z_{(i)}\right\} = 0 \tag{4.1.13}$$

$$\frac{\partial \ln L}{\partial \theta_1} \cong \frac{\partial \ln L^*}{\partial \theta_1} = \frac{1}{\sigma}\sum_{i=1}^{n} x_{[i]} - \frac{(b+1)}{\sigma}\sum_{i=1}^{n} x_{[i]}\left\{\alpha_i - \beta_i z_{(i)}\right\} = 0 \tag{4.1.14}$$

$$\frac{\partial \ln L}{\partial \sigma} \cong \frac{\partial \ln L^*}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma}\sum_{i=1}^{n} z_{(i)} - \frac{(b+1)}{\sigma}\sum_{i=1}^{n} z_{(i)}\left\{\alpha_i - \beta_i z_{(i)}\right\} = 0. \tag{4.1.15}$$

As the differences $g(z_{(i)})$-$(\alpha_i$-$\beta_i z_{(i)})$ goes to zero as n tends to infinity, modified likelihood equations (4.1.13)-(4.1.15) are asymptotically equivalent to maximum likelihood equations (4.1.6)-(4.1.8) (Vaughan and Tiku, 2000). That is,

$$\lim_{n\to\infty} \frac{1}{n}\left|\frac{\partial \ln L}{\partial \theta_0} - \frac{\partial \ln L^*}{\partial \theta_0}\right| = 0, \ \lim_{n\to\infty} \frac{1}{n}\left|\frac{\partial \ln L}{\partial \theta_1} - \frac{\partial \ln L^*}{\partial \theta_1}\right| = 0,$$

$$\lim_{n\to\infty} \frac{1}{n}\left|\frac{\partial \ln L}{\partial \sigma} - \frac{\partial \ln L^*}{\partial \sigma}\right| = 0. \tag{4.1.16}$$

Therefore, we can conclude that MML estimators are asymptotically equivalent to the corresponding ML estimators.

MML estimators are the solutions of the equations (4.1.13)-(4.1.15),

$$\hat{\theta}_{0(MML)} = \bar{y}_{[.]} - \hat{\theta}_{1(MML)}\bar{x}_{[.]} - (\Delta / m)\hat{\sigma}_{(MML)} \tag{4.1.17}$$

$$\hat{\theta}_{1(MML)} = K - D\hat{\sigma}_{(MML)} \tag{4.1.18}$$

$$\hat{\sigma}_{(MML)} = \frac{\left\{-B + \sqrt{\left(B^2 + 4nC\right)}\right\}}{2\sqrt{n(n-2)}}, \tag{4.1.19}$$

where

$$m = \sum_{i=1}^{n} \beta_i \,,\ \Delta_i = \alpha_i - (b+1)^{-1},\ \Delta = \sum_{i=1}^{n} \Delta_i$$

$$\bar{y}_{[.]} = \frac{1}{m}\sum_{i=1}^{n} \beta_i y_{[i]} \,,\ \bar{x}_{[.]} = \frac{1}{m}\sum_{i=1}^{n} \beta_i x_{[i]}$$

$$K = \sum_{i=1}^{n} \beta_i \left(x_{[i]} - \bar{x}_{[.]}\right) y_{[i]} \bigg/ \sum_{i=1}^{n} \beta_i \left(x_{[i]} - \bar{x}_{[.]}\right)^2$$

$$D = \sum_{i=1} \Delta_i \left(x_{[i]} - \bar{x}_{[.]}\right) \bigg/ \sum_{i=1}^{n} \beta_i \left(x_{[i]} - \bar{x}_{[.]}\right)^2 \tag{4.1.20}$$

$$B = (b+1)\sum_{i=1}^{n} \Delta_i \left\{y_{[i]} - \bar{y}_{[.]} - K\left(x_{[i]} - \bar{x}_{[.]}\right)\right\}$$

$$C = (b+1)\sum_{i=1}^{n} \beta_i \left\{y_{[i]} - \bar{y}_{[.]} - K\left(x_{[i]} - \bar{x}_{[.]}\right)\right\}^2$$

$$= (b+1)\left\{\sum_{i=1}^{n} \beta_i \left(y_{[i]} - \bar{y}_{[.]}\right)^2 - K\sum_{i=1}^{n} \beta_i \left(x_{[i]} - \bar{x}_{[.]}\right) y_{[i]}\right\}.$$

Note that β_i>0, therefore m>0. Note that for b=1 the Generalised Logistic distribution turns out to be logistic distribution and Δ=0 for all n.

The MML estimators are computed in two iterations. In the first iteration, the order statistics $w_{(i)}$'s are calculated by ordering $w_i = y_i - \hat{\theta}_{1(OLS)} x_i$, $(1 \le i \le n)$ where $\hat{\theta}_{1(OLS)}$ is given in the equation (3.2). Then, $\hat{\theta}_{1(MML)}$ is obtained from the equation (4.1.18). In the second iteration, the estimate $\hat{\theta}_{1(OLS)}$ is replaced by $\hat{\theta}_{1(MML)}$ and the computations are repeated. In Islam et al. (2001), it is said that in order to stabilize the estimates sufficiently, two iterations are enough.

It should be noted that MML method is suitable when the error terms come from skewed, long or short-tailed symmetric distributions even for small samples. In order to understand this methodology more clearly, it is applied to a data set (n=25) which is taken from Draper and Smith (1966). For this data set, assuming the distribution of the error terms Generalised Logistic with b=1 is appropriate. The plot of this data is given in Figure 4.1.1 where the response variable y represents pounds of steam used monthly and the design variable x represents average atmospheric temperature (degrees Fahrenheit).

Table 4.1.1 Steam (y_i) and Temperature (x_i) Data

i	y_i	x_i	i	y_i	x_i	i	y_i	x_i
1	10.98	35.3	10	9.14	57.5	19	6.83	70.0
2	11.13	29.7	11	8.24	46.4	20	8.88	74.5
3	12.51	30.8	12	12.19	28.9	21	7.68	72.1
4	8.40	58.8	13	11.88	28.1	22	8.47	58.1
5	9.27	61.4	14	9.57	39.1	23	8.86	44.6
6	8.73	71.3	15	10.94	46.8	24	10.36	33.4
7	6.36	74.4	16	9.58	48.5	25	11.08	28.6
8	8.50	76.7	17	10.09	59.3			
9	7.82	70.7	18	8.11	70.0			

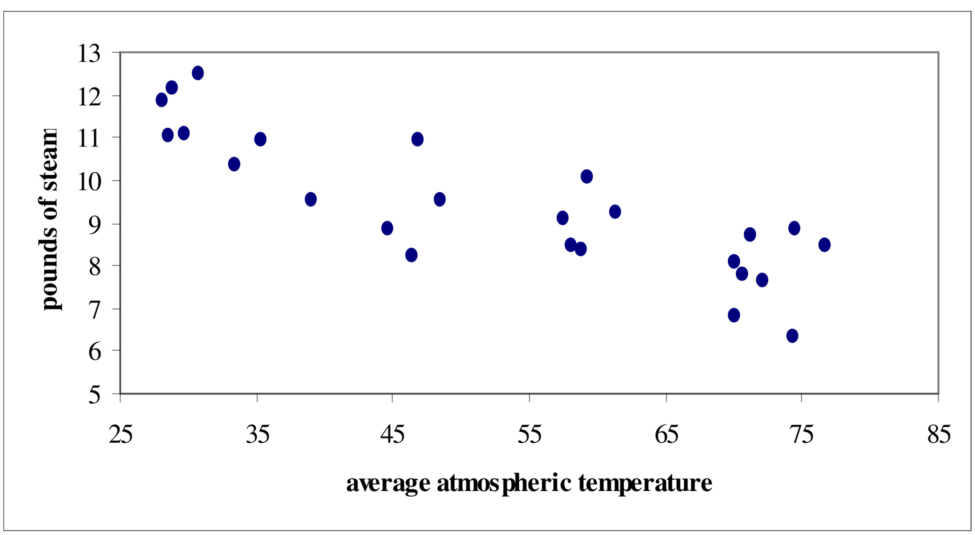


Figure 4.1.1 The plot of pounds of steam (y_i) versus average atmospheric temperature (x_i).

According to the equations (4.1.17)-(4.1.20), the first iteration results for $\hat{\theta}_{0(MML)}$=13.60513, $\hat{\theta}_{1(MML)}$=–0.07920 and $\hat{\sigma}_{(MML)}$=0.53598, while the second iteration results are $\hat{\theta}_{0(MML)}$=13.60552, $\hat{\theta}_{1(MML)}$=-0.07921 and $\hat{\sigma}_{(MML)}$=0.53595. Therefore, the estimated regression line is

$$\hat{y}_i = 13.60552 - 0.07921 x_i, \quad i=1,2,\ldots,25.$$

4.2 Least Absolute Deviation

In the method of LAD, the aim is to minimize the sum of absolute values of errors and the resulting estimators are called as the *median estimators* which are the well-known robust estimators of location. The S function is $S(\theta_0, \theta_1) = \sum_{i=1}^{n} \left| y_i - (\theta_0 + \theta_1 x_i) \right|$. For a fixed θ_1, the θ_0, which minimizes the S function, is the sample median of $\{y_i\text{-}\theta_1 x_i\}$. In other words, it is one of the $\{y_i\text{-}\theta_1 x_i\}$ if n is odd, or any value between the two $\{y_i\text{-}\theta_1 x_i\}$, if n is even (Brown, 1980).

Although the concept of LAD is not more diffucult than the concept of the OLS estimation, calculation of the LAD estimates is more troublesome. Since there are no exact formulas for LAD estimates, an algorithm is used . Birkes and Dodge (1993) explain this algorithm for the model (1.1). It is known that LAD regression line passes through two of the data points. Therefore, the algorithm begins with one of the data points, denoted by (x_1, y_1), and tries to find the best line passing through it. This line also passes through another data point, say (x_2,y_2). The next step is to find the best line passing through (x_2, y_2) and another data point say (x_3, y_3). Then the best line passing through (x_3, y_3) is obtained and the procedure goes on. As the algorithm continues, the lines that are obtained are increasingly better. Finally, the most recent line obtained will be the same as the previous line. This is the best line among all lines, called LAD regression line.

Birkes and Dodge (1993) describe the procedure for finding the best line among all lines passing through a given data point (x_0,y_0). For each data point (x_i,y_i), the slope of the line passing through the two points (x_0, y_0) and (x_i, y_i) is calculated and it is equal to the $(y_i-y_0)/(x_i-x_0)$. If $x_i = x_0$ for some i, the slope is not defined. The data points are reindexed in such a way that

$$(y_1 - y_0)/(x_1 - x_0) \le (y_2 - y_0)/(x_2 - x_0) \le \ldots \le (y_n - y_0)/(x_n - x_0)$$

$$T = \sum_{i=1}^{n} |x_i - x_0| .$$

We try to find the index k that satisfies the conditions

$$|x_1 - x_0| + \ldots + |x_{k-1} - x_0| < \frac{1}{2}T \tag{4.2.1}$$

$$|x_1 - x_0| + \ldots + |x_{k-1} - x_0| + |x_k - x_0| > \frac{1}{2}T \ .$$

The best line passing through (x_0, y_0) is the line $\hat{Y} = \theta_0^* + \theta_1^* X$ where

$$\theta_1^* = \frac{y_k - y_0}{x_k - x_0} \tag{4.2.2}$$

$$\theta_0^* = y_0 - \theta_1^* x_0.$$

It will be simpler to understand if we apply this iteration procedure to a set of data. In Birkes and Dodge (1993), 14 countries in North and Central America with populations over one million people in 1985 are analyzed. For each country, birth rate (the number of births per year per 1000 people) for 1980-1985 and urban percentage (the percentage of the population living in cities over 100,000) in 1980 are given. Since one data point (Trinidad-Tobago) stands apart from the rest of the points, OLS method gives unreliable results. Hence, an alternative regression method which restrains the influence of outlying data points is needed.

Table 4.2.1 Birth Rate Data

Country	Birth Rate (y_i)	Urban Percentage (x_i)
Canada	16.2	55.0
Costa Rica	30.5	27.3
Cuba	16.9	33.3
Dominican Republic	33.1	37.1
El Salvador	40.2	11.5
Guatemala	38.4	14.2
Haiti	41.3	13.9
Honduras	43.9	19.0
Jamaica	28.3	33.1
Mexico	33.9	43.2
Nicaragua	44.2	28.5
Panama	28.0	37.7
Trinidad-Tobago	24.6	6.8
United States	16.0	56.5

The first step in applying LAD simple regression algorithm is to find the best line passing through the first data point, say Canada (55.0, 16.2). The slopes, which are equal to (y_i-16.2)/(x_i-55.0), are shown in Table 4.2.2 for 13 countries in increasing order. To find the k index that satisfies the equation (4.2.2), we calculate $\sum_{i=1}^{n} |x_i - 55.0| = 355.9$, divide it by 2, and look for the country for which the cumulative sum of $|x_i$-55.0$|$ first exceeds 177.95, and we conclude that this country is El Salvador. Hence,

$$\theta_1^* = -0.5517 \text{ and } \theta_0^* = 16.2 - (-0.5517)(55.0) = 46.54 .$$

We can say that the best line through the Canada point also passes through the El Salvador data point.

Table 4.2.2 Calculations Used in Finding the Best Line Through The Canada Data Point

Country	$\frac{y_i - 16.2}{x_i - 55.0}$	$\lvert x_i - 55.0 \rvert$	Cumulative sum
Mexico	-1.5000	11.8	11.8
Nicaragua	-1.0566	26.5	38.3
Dominican Republic	-0.9441	17.9	56.2
Honduras	-0.7694	36.0	92.2
Panama	-0.6821	17.3	109.5
Haiti	-0.6107	41.1	150.6
Jamaica	-0.5525	21.9	172.5
El Salvador	***-0.5517***	***43.5***	***216.0***
Guatemala	-0.5447	40.8	256.8
Costa Rica	-0.5162	27.7	284.5
Trinidad-Tobago	-0.1743	48.2	332.7
United States	-0.1333	1.5	334.2
Cuba	-0.0323	21.7	355.9

Now we should find the best line passing through the El Salvador data point, (11.5, 40.2). The slopes $(y_i\text{-}40.2)/(x_i\text{-}11.5)$ are calculated and put in increasing order. If a table like Table 4.2.2 is constructed, we will see that the cumulative sum of $|x_i\text{-}11.5|$ is equal to 265.5 and the country whose cumulative first exceeds 265.5/2=132.75 is the United States. Therefore, we can conclude that the best line passing through the El Salvador data point also passes through the United States data point.

The next step is to find the best line passing through the United States data point. Doing the same calculations, we find that the best line passing through the United States data point also passes through the El Salvador data point. Since this is the same line we found at the previous step, the algorithm stops. The LAD regression line passes through El Salvador and the United States data point. Therefore, the LAD estimators are

$$\hat{\theta}_{1(LAD)} = (40.2 - 16.0)/(11.5 - 56.5) = -0.5378$$
$$\hat{\theta}_{0(LAD)} = 40.2 - (-0.5378)(11.5) = 46.38.$$

and the LAD regression line is

$$\hat{y}_i = 46.38 - 0.5378 x_i, \qquad i=1,2\ldots,14.$$

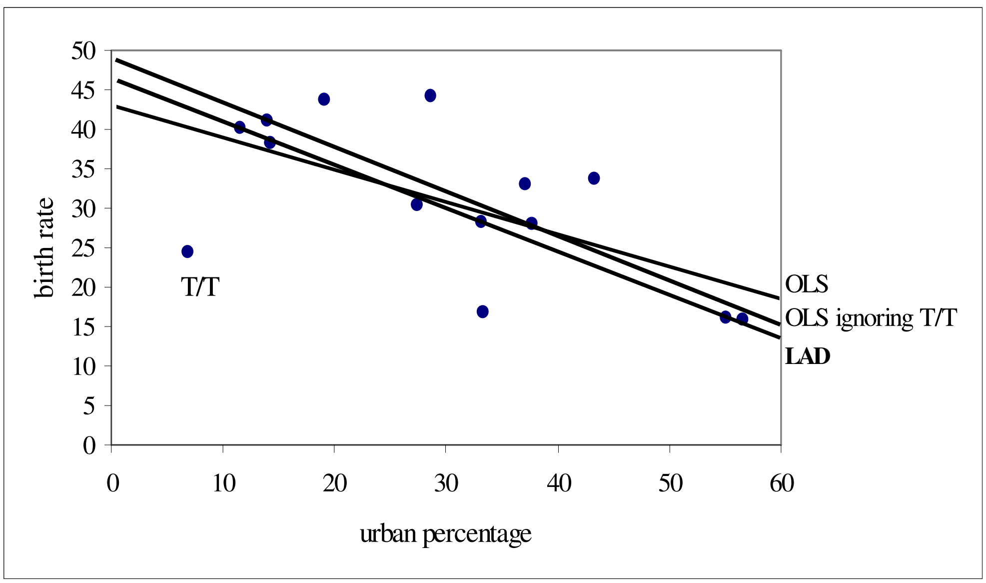


Figure 4.2.1 Plot of the birth rate data with three estimated regression lines, one by OLS using all the data $\hat{y}_i = 43 - 0.40 x_i$, one by OLS ignoring Trinidad-Tobago $\hat{\mathbf{y}}_i = 49 - 0.55 x_i$, and the other by LAD $\hat{y}_i = 46 - 0.54 x_i$.

With the help of this algorithm we can find the LAD regression line for most data sets. However, because of nonuniqueness or degeneracy, problems may arise. Nonuniqueness means that there is more than one best line passing through a data point and degeneracy means that the best line through a data point also passes through two or more other data points. In the previously described algorithm, at each step we find the best line passing through a given data point. The other data point, which this line passes through, is found and this data point is used in the next step. But when there is nonuniqueness, there is more than one best line and when there is degeneracy, the best line passes through more than one other data point. In these cases, there is more than one choice for the data point to be used in the next step. False choices may lead the algoritm go around in circles, or stop with a line that is not the LAD regression line (Birkes and Dodge,1993).

As it is stated in Birkes and Dodge (1993), one of the optimal properties of the LAD estimators is that they minimize the sum of the absolute residuals. In addition to this, if the error terms come from Laplace or double exponential distribution, then these estimators are the ML estimators. If the probability of the random errors being near 0 is positive, the LAD estimates are consistent and asymptotically normal, provided that the $(X'X)^{-1}$ matrix has small entries and Birkes and Dodge (1993) remark that the consistency of the estimators does not require that the population of errors have finite variance.

Because of the robustness property of LAD estimation with respect to the distribution of the response variable (although not with respect to the explanatory variables), LAD estimates are sometimes recommended as starting values for iterative estimation algorithms. This method is especially suitable when the error terms come from heavy-tailed distributions and when the sample is very large. A heavy-tailed distribution is a distribution whose probability density function (pdf) graph have thicker tails than the pdf of a normal distribution. Under these conditions, LAD estimates have low variance (Birkes and Dodge, 1993).

4.3 Winsorized Least Squares

Another alternative method to OLS estimation procedure is the Winsorized regression which is applied by altering the data values based upon the magnitude of the residuals. By diminishing the effect of outliers, this method tries to decrease the influence of the contamination on the estimators. For simple linear regression, Winsorization is applied by ordering the values and modifying the most extreme values to the next most extreme values. In other words, the response variable which stands apart from the rest of the points (i.e., the residual for that point will be large in absolute value) is smoothed by replacing the observed residual for that *y*-value with the next closest, smaller residual in the data set in an iterative way (Yale and Forsythe, 1976; Nevitt and Tam, 1998).

In Yale and Forsythe (1976), it is stated that Winsorization does not harm a good linear relationship on an non-contaminated sample, and improves the estimates of θ_0 and θ_1 when the sample contains observations from a population other than the target population.

In Yale and Forsythe (1976) the general Winsorization procedure is applied in three different methods, the *iteration* method, the *levels* method, and the *iteration at increasing levels* method. They explained these methods as,

Let $\hat{\varepsilon}_i = y_i - \hat{y}_i$,and the ordered residuals are $\hat{\varepsilon}_{(1)} \le \hat{\varepsilon}_{(2)} \le ... \le \hat{\varepsilon}_{(n)}$

where

y_i =ith value of response variable *y* in a sample of *n* points

$\hat{y}_i$ =estimate of y_i at a given x_i ($\hat{y}_i = \hat{\theta}_{0(OLS)} + \hat{\theta}_{1(OLS)} x_i$, the estimates of true intercept (θ_0) and slope (θ_1) are calculated by the usual OLS procedure).

***Iteration* method**: The number of points Winsorized at each extreme of the ordered residuals remains constant at each of (*l*) repetitions. The observed residual for an extreme *y*-value is replaced with the next closest residual and new *y*-value is computed by $y_i^{'} = \hat{y}_i + \hat{\varepsilon}_i^{'}$. Then by using these new *y*-values, new slope and intercept estimates for the regression line are calculated and new residuals are obtained for the second iteration. Then *x*, $y^{''}$ (where $y_i^{''} = \hat{y}_i^{'} + \hat{\varepsilon}_i^{''}$) are computed for the third iteration, and so on. For each iteration, the baseline data are modified by generating a new set of residuals ($\hat{\varepsilon}_i, \hat{\varepsilon}_i^{'}, \hat{\varepsilon}_i^{''}$ and so on). Therefore, each iteration depends on the previous iterations. The pilot studies that Nevitt and Tam (1998) conducted about Winsorized regression show that the parameter estimates after five degrees of data smoothing are very similar. Therefore, the samples are exposed to five steps of Winsorization after deciding the initial estimates based on OLS analyses. To reduce confusion in terminology, Yale and Forsythe (1976) use the expression 'degree of Winsorization' to denote a single step or iteration of Winsorization.

***Levels* method**: The Winsorization procedure is repeated for a specified number of iterations and the number of points Winsorized at each extreme defines the 'level' of Winsorizing the data. The level increases at each repetition, each time moving more residuals in from each extreme. (Yale and Forsythe, 1976)

***Iterations at increasing levels* method**: This method is similar to the iterations method. However, instead of holding the number of Winsorized points constant as it is made in iteration method, the number of Winsorized points increases by a given level (e.g., two times) at each repetition.

In Yale and Forsythe (1976), it is stated that after some steps of Winsorization, the results of all three methods would be the same for a given sample at a given value of Winsorized points. However, the simulation results show that when the samples are small (n=10), the iteration method displays a stable efficiency compared to the other Winsorization methods, which deteriorate after one or two steps of Winsorization. As sample size gets larger, iteration method remains at a constant loss or gain after one step of Winsorization.

Daniel and Wood (1971) used the pilot plant data (n=20) to show that there is a good linear fit between the design variable x (organic acid content determined by extraction) and the response variable y (acid content determined by titration). In order to compare the Winsorized and ordinary least squares methods, Yale and Forsythe (1976) contaminated the two data points in the same data set and then made the necessary calculations. The plot of the contaminated and uncontaminated data set is given in Figure 4.3.1(a) and (b).

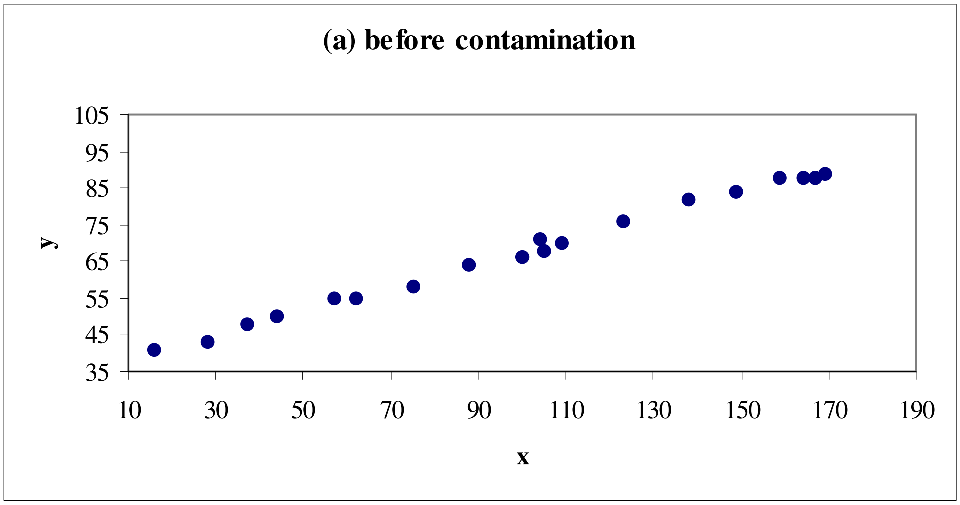
Table 4.3.1 Pilot-Plant Data

i	y_i	x_i	i	y_i	x_i
1	76	123	11	82	138
2	70	109	12	68	105
3	55	62	13	88	159
4	71	104	14	58	75
5	55	57	15	64	88
6	48	37	16	88	164
7	50	44	17	89	169
8	66	100	18	88	167
9	41	16	19	84	149
10	43	28	20	88	167

Table 4.3.2 Contaminated Pilot-Plant Data

i	y_i	x_i	i	y_i	x_i
1*	100	123	11	82	138
2	70	109	12	68	105
3*	75	62	13	88	159
4	71	104	14	58	75
5	55	57	15	64	88
6	48	37	16	88	164
7	50	44	17	89	169
8	66	100	18	88	167
9	41	16	19	84	149
10	43	28	20	88	167

* Contaminated points



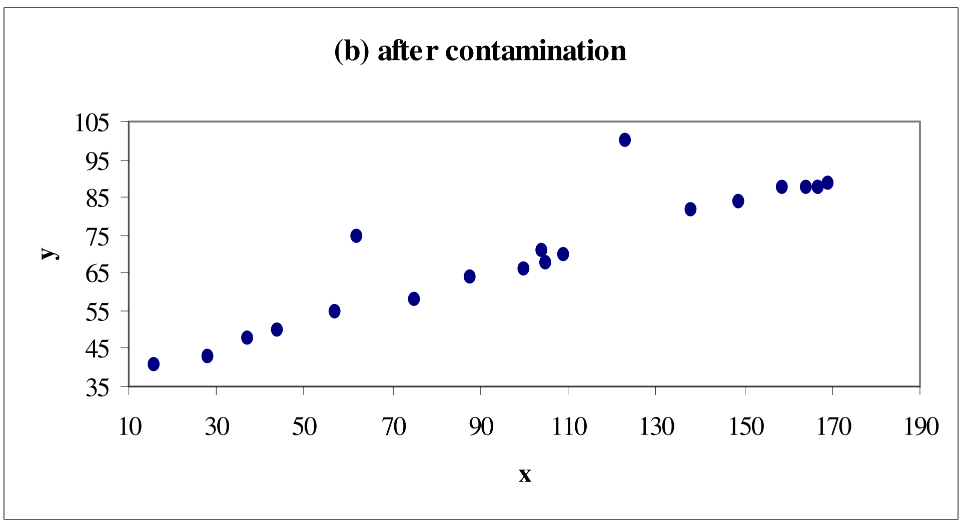


Figure 4.3.1 The plot of the (a) uncontaminated, (b) contaminated pilot-plant data

If we carry out the OLS methodology for the uncontaminated data set, we get the equation $\hat{y}_i = 35.458 + 0.322x_i$, and applying five degrees of Winsorization with the iteration method (Winsorization percentage= 10%), we obtain the Winsorized least squares regression line as $\hat{y}_i = 35.460 + 0.321x_i$. After the contamination, The OLS line turns out to be $\hat{y}_i = 38.377 + 0.315x_i$, which is far from uncontaminated OLS line. On the other hand, the Winsorized line is $\hat{y}_i = 35.841 + 0.320x_i$ whose estimates are closer to the uncontaminated OLS estimates.

4.4 Least Trimmed Squares

LTS, developed by Rousseeuw, is an estimation procedure which achieves the purpose of being insensitive to changes in small percentage of data points. The aim is to minimize

$$\sum_{i=1}^{h}(y_i - \theta_0 - \theta_1 x_i)^2 . \tag{4.4.1}$$

To minimize the function (4.4.1), we should choose a subsample of h observations and compute some θ_0 and θ_1 that minimize the sum of squared residuals for the selected subsample. By applying this procedure to all subsamples, we have $\binom{n}{h}$ estimates for both θ_0 and θ_1, and the estimate which makes the objective function smallest is the final estimate (Cizek and Visek, 2000). Unfortunately, it is very diffucult to obtain the all subsamples unless a very small sample is analyzed. Let for any given θ_0 and θ_1, the squared errors are $\varepsilon_i^2 = (y_i - \theta_0 - \theta_1 x_i)^2$ and they are ordered in an increasing sequence, $0 \le \varepsilon_{(1)}^2 \le \varepsilon_{(2)}^2 \le \ldots \le \varepsilon_{(n)}^2$ .Nevitt and Tam (1998) delete data points corresponding to a specified percentage of the largest residuals under an initial OLS estimation. Then, they calculate the LTS estimates of θ_0 and θ_1 by performing OLS estimation on the remaining data.

In LTS procedure, the outlying cases are deleted to reduce their adverse effect on the estimators rather than smoothing the data as in Winsorized regression and the only difference between OLS and LTS estimation is that in LTS, the largest squared residuals are not used (n-h observations will not affect the estimator). In Rousseeuw and Leroy (1987), it is stated that the best robustness properties are achieved when h is approximately $n/2$, in which case the breakdown point attains 50%.

Rousseeuw and Leroy (1987) find a real life data with outliers, from Belgian Statistical Survey published by the Ministry of Economy. They investigate the total number (in tens of millions) of international phone calls made.

Table 4.4.1 Number of International Calls from Belgium

year (x_i)	number of calls (y_i)	year (x_i)	number of calls (y_i)
50	0.44	62	1.61
51	0.47	63	2.12
52	0.47	64	11.90
53	0.59	65	12.40
54	0.66	66	14.20
55	0.73	67	15.90
56	0.81	68	18.20
57	0.88	69	21.20
58	1.06	70	4.30
59	1.20	71	2.40
60	1.35	72	2.70
61	1.49	73	2.90

The plot in Figure 4.4.1 shows an upward trend over the years, but the data from years 1964 to 1969 seems to lie apart from the rest of the data. Rousseeuw and Leroy (1987) say that during these years, another recording system is used, giving the total number of minutes which leads to cause outliers in y-direction.

The dashed line in Figure 4.4.1 is the OLS line and the outlying data points between the years 1964-1969 affects this line and as a result, OLS line has a large slope and does not have a good fit.

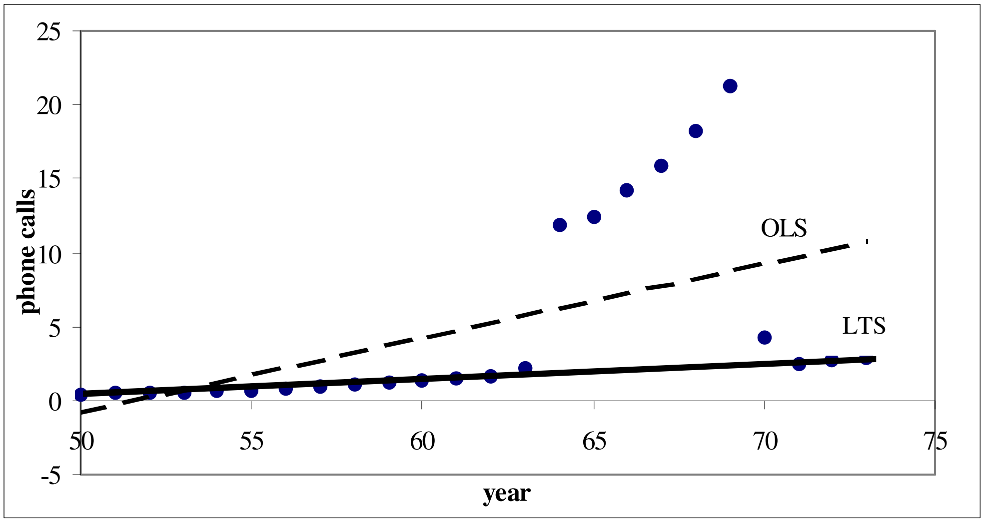


Figure 4.4.1 Number of international phone calls from Belgium in the years 1950-1973 with the OLS (dashed line) and LTS fit (solid line)

The solid line in Figure 4.4.1 shows the LTS line and compared to OLS line, this line is considered as the right one, because it is not affected by the data points coming from years 1964-1969.

Cizek and Visek (2000) say that such an effect might be caused even by a single observation and they deal with an artificial data set with 10 observations one of which is an outlier. Looking at the Figure 4.4.2, it is simple to determine the contaminated data points. However, this may cause problems for high dimensional data and applying the usual OLS procedure and eliminating the observation with the highest residual may cause problems, because some of the good observations yield larger residuals than the outlying data point. For that cases, only looking at the residuals will not be reliable. Zaman et al. (2001) say that one way of avoiding this problem is to deal with the *leverage* of the observations. They state that observations whose x_i is close to the center have little leverage. Rousseeuw and Van Zomeren (1990) express the leverage of the ith observation as the distance between x_i and the center of the x_i of all the observations. Since these points are also affecting the OLS analysis, Zaman et al. 2001) state that the points with high leverage and LTS residuals should be eliminated.

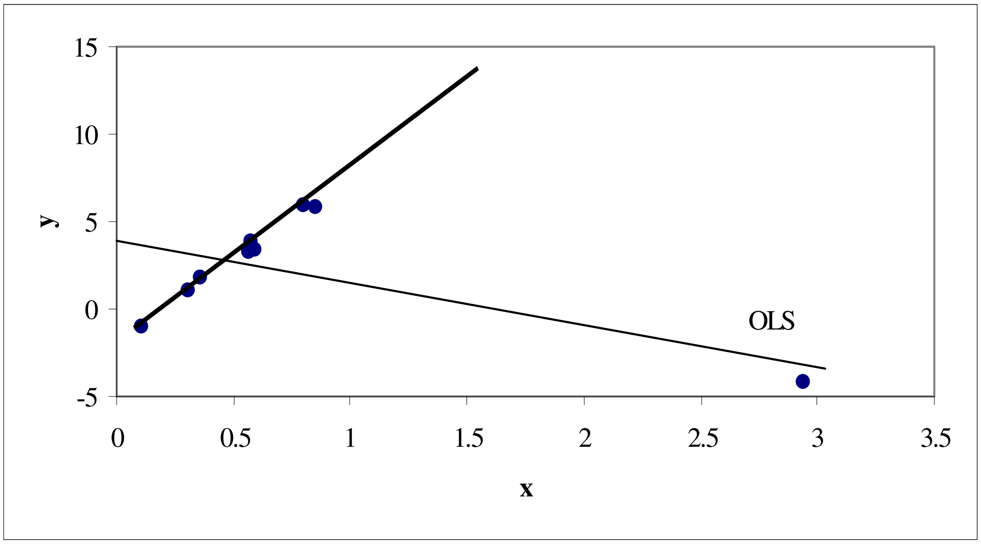


Figure 4.4.2 Plot of the data set with one outlier. The pale line represents the OLS fit while the solid line shows the underlying model.

Another problem mentioned in Zaman et al. (2001) is that although LTS method is good at finding out the outliers, it may sometimes eliminate too many observations and this may not give the true regression relation about the data in which the researcher is interested.

Zaman et al. (2001) also touch on the subject of analyzing the outliers. They say that outlier analysis is not an entirely mechanical task. The occurrence of outliers should be carefully analyzed and the different treatments should be applied as a result of these analyses. If an outlier occurs because of a measurement error, then deleting that data point yields good results. However, if the outlier is a valid observation, it should be considered during the analyses and eliminating it may be problematic.

4.5 Theil's and Weighted Theil's Regression

Theil's regression is a nonparametric method which is used as an alternative to robust methods for data sets with outliers. Although the nonparametric procedures perform reasonably well for almost any possible distribution of errors and they lead to robust regression lines, they require a lot of computation. This method is suggested by Theil (1950), and it is proved to be useful when outliers are suspected, but when there are more than few variables, the application becomes diffucult.

Sprent (1993) states that for a simple linear regression model to obtain the slope of a line that fits the data points, the set of all slopes of lines joining pairs of data points (x_i, y_i), (x_j, y_j), $x_j \neq x_i$, for $1 \leq i < j \leq n$ should be calculated,

$$b_{ij} = \frac{y_j - y_i}{x_j - x_i}. \tag{4.5.1}$$

Hussain and Sprent (1983) say that no generality is losen if we take $1 \leq i < j \leq n$ assuming that the x_i are arranged in ascending order. Note that $b_{ij} = b_{ji}$. According to these results the the *Theil's slope estimator* is

$$\hat{\theta}_{1(THL)} = med\{b_{ij} \mid x_j \neq x_i\},$$

where $x_1 \leq x_2 \leq ... \leq x_n$.

It is known that median estimators are less affected compared to the mean estimators. Therefore, these estimators are resistant to outliers in the sample data.

Nevitt and Tam (1998) state that there are several methods for computing the y-intercept. One of these methods is to calculate

$$a_{ij} = \frac{x_j y_i - x_i y_j}{x_j - x_i}, \qquad i < j, \quad x_i \neq x_j, \tag{4.5.2}$$

and taking the median of these a_{ij} values will give us the y-intercept.

Nevitt and Tam (1998) also investigate a different approach which is proposed by Theil (1950). For these approach, these a_{ij} values need not to be explicitly calculated. For a $\hat{\theta}_{1(THL)}$ slope estimator, the $y_i - \hat{\theta}_{1(THL)} x_i$ is computed for each observation, and the median of these terms are expressed as the y-intercept of the regression line.

To reduce the effect of outlying observations, some modifications are applied to Theil's method and each of the pairwise slopes, b_{ij}'s, are weighted by some weighting procedures. The *weighted Theil slope estimator* for the n observations in the sample data is the weighted median of these b_{ij}'s. Note that $\hat{\theta}_{1(OLS)} = \sum w_{ij} b_{ij}$ where

$$w_{ij} = \frac{(x_i - x_j)^2}{\sum (x_i - x_j)^2}, \tag{4.5.3}$$

$\sum$ represents $n(n$-1)/2 pairs of integers i and j with $1 \le i < j \le n$.

Birkes and Dodge (1993) explain how a weighted median can be calculated as follows:

First x_i's are ordered in an increasing sequence, so that $x_1 \le x_2 \le ... \le x_n$. Note that the weights, w_i's, are nonnegative and add up to 1. Obtaining the index k where k

$$\begin{aligned} &w_1 + w_2 + ... + w_{k-1} < 0.5 \\ &w_1 + w_2 + ... + w_{k-1} + w_k > 0.5. \end{aligned} \tag{4.5.4}$$

x_k is the *weighted median*. Note that if $w_1 + w_2 + ... + w_{k-1} = 0.5$, then the weighted median is $(x_{k-1} + x_k)/2$. If the weights are exactly equal to eachother (i.e. w_i=1/n), the weighted median will be the ordinary median.

The weighted Theil slope estimator of θ_I is the pairwise slopes $b_{ij}=(y_i-y_j)/(x_i-x_j)$, with weights

$$w_{ij} = \left|x_i - x_j\right| \Big/ \sum \left|x_i - x_j\right|,$$

and $\hat{\theta}_{0(wtd.THL)}$ is the ordinary median of $y_i - \hat{\theta}_{1(wtd.THL)} x_i$.

To understand this estimation procedure more clearly, let us consider an example given in Birkes and Dodge (1993). In this example, heights and forearm lengths (in centimeters) of 33 female applicants are taken on applicants to a police department in a city in United States. The plot of this sample data is given in Figure (4.5.1).

Table 4.5.1 Forearm Length Data

i	y_i	x_i	i	y_i	x_i	i	y_i	x_i
1	165.8	28.1	12	167.4	30.1	23	163.1	26.9
2	169.8	29.1	13	159.2	27.3	24	165.8	26.3
3	170.7	29.5	14	170.0	30.9	25	175.4	30.1
4	170.9	28.2	15	166.3	28.8	26	159.8	27.1
5	157.5	27.3	16	169.0	28.8	27	166.0	28.1
6	165.9	29.0	17	156.2	25.6	28	161.2	29.2
7	158.7	27.8	18	159.6	25.4	29	160.4	27.8
8	166.0	26.9	19	155.0	26.6	30	164.3	27.8
9	158.7	27.1	20	161.1	26.6	31	165.5	28.6
10	161.5	27.8	21	170.3	29.3	32	167.2	27.1
11	167.3	27.3	22	167.8	28.6	33	167.2	29.7

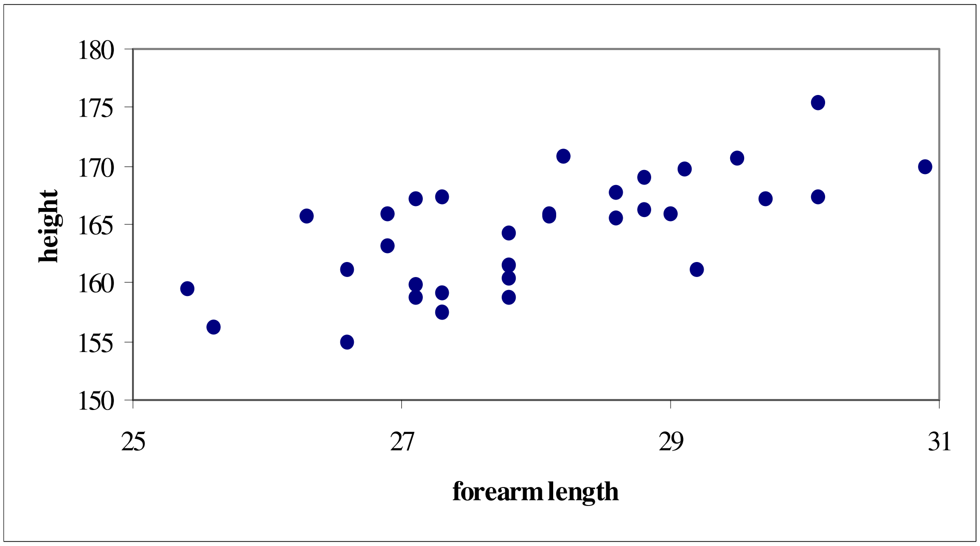


Figure 4.5.1 The plot of the forearm length data

There are 33(32)/2=528 pairwise slopes (b_{ij}) for 33 data points. However, for 18 points, the two data points have the same x-value. Hence Birkes and Dodge (1993) deal with 510 slopes. To each slope, the weight $w_{ij} = \left|x_i - x_j\right| / \sum\left|x_i - x_j\right|$ where $\sum\left|x_i - x_j\right| = 802.2$. As it is shown in Table 3.5.2, the 510 pairwise slopes are are put in increasing order and cumulative sum of weights are calculated. Then, the first cumulative sum that exceeds 0.5 is detected and the corresponding slope is taken as the weighted Theil's slope estimator. To calculate the y-intercept estimator, the 33 differences $y_i - \hat{\theta}_{1(wtd.THL)} x_i$ are computed. Then, the median of these differences is the $\hat{\theta}_{0(wtd.THL)}$.

According to Table 4.5.2, $\hat{\theta}_{1(wtd.THL)} = 2.683$ and $\hat{\theta}_{0(wtd.THL)} = 89.71$. Hence the estimated regression line is

$$\hat{y}_i = 89.71 + 2.683 x_i, \qquad i=1,2,\ldots,33.$$

Table 4.5.2 Pairwise slopes for the forearm length data, in increasing order, and their weights

Slope b_{ij}	weight w_{ij}	cumulative sum of weights
-86.000	0.000125	0.000125
-48.500	0.000249	0.000374
-40.000	0.000249	0.000623
-36.500	0.000249	0.000873
-36.000	0.000374	0.001247
-31.000	0.000249	0.001496
.	.	.
.	.	.
.	.	.
2.604	0.006607	0.497632
2.667	0.001496	0.499127
2.667	0.000374	0.499501
2.683	***0.005111***	***0.504612***
2.684	0.002368	0.506981
2.684	0.004737	0.511718
.	.	.
.	.	.
.	.	.
37.500	0.000249	0.999252
39.000	0.000125	0.999377
43.000	0.000249	0.999626
49.000	0.000125	0.999751
51.000	0.000125	0.999875
91.000	0.000125	1.000000

CHAPTER 5

SIMULATION RESULTS

This simulation study is performed for sample sizes n=10, 20, 50, and 100. Simulated bivariate data sets consist of (x, y) vectors. As in Nevitt and Tam (1998), the design points, x_i's, are generated to follow an equally spaced, sequential additive series (x_i=1, 2,…, n) and they are common to all the $N = [100000/n]$ random samples. The y-vector is generated as $y_i = x_i + \varepsilon_i$, $i = 1,2,...,n$, which means that the y-intercept parameter θ_0=0.0, and slope parameter θ_1=1.0, and the random error, ε_i comes from the Generalised Logistic distribution given in equation (1.2) where σ=1.0.

For each simulated data set, the estimators of θ_0 and θ_1 are calculated. The estimation techniques that we take into account are OLS, MML, LAD, 10% and 20% Winsorized least squares, 20% LTS, Theil's and weighted Theil's regression. The algorithms for calculating these estimators are explained in Chapter 3 and Chapter 4. But, note that for the weighted Theil's regression all the weighting schemes described in Chapter 2 is similar , because $x_i = i$ for all i.

In this simulation study, by using these procedures, the y-intercept and slope estimators are computed and for each estimator mean, variance, bias, mean square error (MSE) and relative mean square error (RMSE) are calculated where

$$MSE(\hat{\theta}) = \mathrm{var}(\hat{\theta}) + [bias(\hat{\theta})]^2, \tag{5.1}$$

$$RMSE(\hat{\theta}) = \frac{MSE_{OLS} - MSE_{\theta'}}{MSE_{OLS}} \quad . \tag{5.2}$$

RMSE is useful to measure the quality of the parameter estimation. Positive values of the RMSE can be expressed as there is a proportional decrease in the MSE of a given estimator with respect to OLS estimation.

The simulated means, variances, bias, MSE and RMSE of the estimators are computed not only for some selected values of b (i.e. b=0.5, 1.0, 2.0, 4.0, 6.0) and n but also for some plausible alternative models. For these alternatives, the underlying true distribution is assumed to be Generalised Logistic with b=4.0 and σ=1.0. The alternative models are the following:

Model (1): $GL(2.0, \sigma)$

Model (2): $GL(6.0, \sigma)$

Model (3): Outlier Model; $(n\text{-}r)$ observations from $GL(4.0, \sigma)$ and r observations from $GL(4.0, 2\sigma)$ where $r = [0.5+0.1n]$

Model (4): Mixture Model; $0.90GL(4.0, \sigma) + 0.10GL(4.0, 2\sigma)$

Model (5): Contamination Model; $0.90GL(4.0, \sigma) + 0.10Exp(1.0)$

The simulation results are presented in Tables 5.1 to 5.16.

Table 5.1 Simulation Results for Population y-intercept (θ_0=0.0), n=10

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	-1.35790908	3.05212307	-1.35790908	4.89603996	0.00
MML	-0.07071680	2.58863997	-0.07071680	2.59364080	0.47
LAD	-1.18183005	3.84321761	-1.18183005	5.23993969	-0.07
WIN10	-1.26986527	2.85374475	-1.26986527	4.46630239	0.09
WIN20	-1.22712183	2.88966417	-1.22712183	4.39549208	0.10
LTS	-1.16030133	3.46522045	-1.16030133	4.81151962	0.02
Theil	-1.15656674	3.08057690	-1.15656674	4.41822338	0.10
Wtd.Theil	-1.15861833	3.01778817	-1.15861833	4.36018467	0.11
			b=1.0		
OLS	0.01113564	1.52703631	0.01113564	1.52716029	0.00
MML	0.01202309	1.46309984	0.01202309	1.46324444	0.04
LAD	0.01277667	2.07120061	0.01277667	2.07136393	-0.36
WIN10	0.01224890	1.50336337	0.01224890	1.50351346	0.02
WIN20	0.01564275	1.52032161	0.01564275	1.52056634	0.00
LTS	0.01397771	1.85209811	0.01397771	1.85229349	-0.21
Theil	0.01804119	1.64712095	0.01804119	1.64744639	-0.08
Wtd.Theil	0.01715941	1.60685456	0.01715941	1.60714900	-0.05
			b=2.0		
OLS	1.01589739	1.05300212	1.01589739	2.08504963	0.00
MML	0.06878061	0.99240702	0.06878061	0.99713778	0.52
LAD	0.94080049	1.49571097	0.94080049	2.38081646	-0.14
WIN10	0.98121172	1.04432547	0.98121172	2.00710201	0.04
WIN20	0.96158624	1.07764959	0.96158624	2.00229764	0.04
LTS	0.94144052	1.29475987	0.94144052	2.18107009	-0.05
Theil	0.92856163	1.14888215	0.92856163	2.01110888	0.04
Wtd.Theil	0.92776453	1.12787485	0.92776453	1.98862183	0.05
			b=4.0		
OLS	1.84628749	0.90380037	1.84628749	4.31257772	0.00
MML	0.11580633	0.84518981	0.11580633	0.85860091	0.80
LAD	1.72633433	1.23978865	1.72633433	4.22001886	0.02
WIN10	1.79243755	0.88978583	1.79243755	4.10261822	0.05
WIN20	1.76322556	0.91384697	1.76322556	4.02281141	0.07
LTS	1.72314262	1.11030734	1.72314262	4.07952785	0.05
Theil	1.72246742	0.95119095	1.72246742	3.91808486	0.09
Wtd.Theil	1.72100151	0.94088227	1.72100151	3.90272856	0.10
			b=6.0		
OLS	2.28032160	0.86650014	2.28032160	6.06636667	0.00
MML	0.12818065	0.86842042	0.12818065	0.88485068	0.85
LAD	2.16095829	1.17338908	2.16095829	5.84312963	0.04
WIN10	2.22577548	0.84881389	2.22577548	5.80289030	0.04
WIN20	2.19426584	0.87477988	2.19426584	5.68958235	0.06
LTS	2.15382791	1.05702949	2.15382791	5.69600391	0.06
Theil	2.14759231	0.90386653	2.14759231	5.51601934	0.09
Wtd.Theil	2.14837289	0.89197922	2.14837289	5.50748539	0.09

Table 5.2 Simulation Results for Population Slope (θ_1=1.0) , n=10

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	0.99520075	0.07899439	-0.00479925	0.07901742	0.00
MML	0.99701673	0.06704813	-0.00298327	0.06705704	0.15
LAD	0.99753475	0.10191830	-0.00246525	0.10192438	-0.29
WIN10	0.99461389	0.07395110	-0.00538611	0.07398012	0.06
WIN20	0.99576592	0.07473260	-0.00423408	0.07475053	0.05
LTS	0.99438173	0.08997677	-0.00561827	0.09000834	-0.14
Theil	0.99487734	0.07716928	-0.00512266	0.07719552	0.02
Wtd.Theil	0.99520135	0.07510103	-0.00479865	0.07512406	0.05
			b=1.0		
OLS	0.99774343	0.03938367	-0.00225657	0.03938876	0.00
MML	0.99773103	0.03783365	-0.00226897	0.03783879	0.04
LAD	0.99821037	0.05457048	-0.00178963	0.05457368	-0.39
WIN10	0.99768955	0.03894486	-0.00231045	0.03895019	0.01
WIN20	0.99733055	0.03923141	-0.00266945	0.03923853	0.00
LTS	0.99694699	0.04813686	-0.00305301	0.04814618	-0.22
Theil	0.99694109	0.04084575	-0.00305891	0.04085511	-0.04
Wtd.Theil	0.99711037	0.03968543	-0.00288963	0.03969378	-0.01
			b=2.0		
OLS	0.99779218	0.02724502	-0.00220782	0.02724990	0.00
MML	0.99837774	0.02539159	-0.00162226	0.02539423	0.07
LAD	0.99672824	0.03874801	-0.00327176	0.03875872	-0.42
WIN10	0.99742478	0.02707307	-0.00257522	0.02707971	0.01
WIN20	0.99722248	0.02788302	-0.00277752	0.02789073	-0.02
LTS	0.99621183	0.03349630	-0.00378817	0.03351065	-0.23
Theil	0.99855244	0.02839547	-0.00144756	0.02839757	-0.04
Wtd.Theil	0.99854952	0.02751860	-0.00145048	0.02752071	-0.01
			b=4.0		
OLS	0.99776155	0.02318613	-0.00223845	0.02319114	0.00
MML	0.99775302	0.01938449	-0.00224698	0.01938954	0.16
LAD	0.99694765	0.03263323	-0.00305235	0.03264254	-0.41
WIN10	0.99766153	0.02280002	-0.00233847	0.02280549	0.02
WIN20	0.99726212	0.02341475	-0.00273788	0.02342224	-0.01
LTS	0.99749571	0.02884074	-0.00250429	0.02884701	-0.24
Theil	0.99683577	0.02325349	-0.00316423	0.02326350	0.00
Wtd.Theil	0.99715388	0.02283428	-0.00284612	0.02284238	0.02
			b=6.0		
OLS	1.00097728	0.02245250	0.00097728	0.02245345	0.00
MML	1.00089633	0.01839460	0.00089633	0.01839540	0.18
LAD	0.99921155	0.03114307	-0.00078845	0.03114369	-0.39
WIN10	1.00038290	0.02198772	0.00038290	0.02198786	0.02
WIN20	1.00046933	0.02265343	0.00046933	0.02265365	-0.01
LTS	1.00030982	0.02739100	0.00030982	0.02739110	-0.22
Theil	1.00087583	0.02247736	0.00087583	0.02247813	0.00
Wtd.Theil	1.00081646	0.02205406	0.00081646	0.02205473	0.02

Table 5.3 Simulation Results for Population y-intercept (θ_0=0.0), n=20

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	-1.44414985	1.45051706	-1.44414985	3.53608584	0.00
MML	-0.07758320	1.14249659	-0.07758320	1.14851570	0.66
LAD	-1.23305440	1.65566576	-1.23305440	3.17608881	0.10
WIN10	-1.33181548	1.28596294	-1.33181548	3.05969548	0.14
WIN20	-1.26769400	1.26526833	-1.26769400	2.87231636	0.19
LTS	-1.20981658	1.47555542	-1.20981658	2.93921161	0.17
Theil	-1.15439212	1.32156301	-1.15439212	2.65418410	0.25
Wtd.Theil	-1.18608689	1.31052303	-1.18608689	2.71732521	0.23
			b=1.0		
OLS	-0.02066782	0.68529016	-0.02066782	0.68571734	0.00
MML	-0.01984782	0.63303852	-0.01984782	0.63343245	0.08
LAD	-0.05782721	0.86069155	-0.05782721	0.86403555	-0.26
WIN10	-0.01782469	0.64233476	-0.01782469	0.64265245	0.06
WIN20	-0.01880665	0.64046061	-0.01880665	0.64081430	0.07
LTS	-0.02434742	0.76236910	-0.02434742	0.76296186	-0.11
Theil	0.00374635	0.68644977	0.00374635	0.68646377	0.00
Wtd.Theil	-0.01744905	0.67759228	-0.01744905	0.67789674	0.01
			b=2.0		
OLS	0.99684298	0.48135325	0.99684298	1.47504914	0.00
MML	0.01303942	0.44175714	0.01303942	0.44192716	0.70
LAD	0.85901630	0.63726270	0.85901630	1.37517166	0.07
WIN10	0.94778544	0.45698255	0.94778544	1.35527980	0.08
WIN20	0.92161173	0.46233749	0.92161173	1.31170571	0.11
LTS	0.89980936	0.55116057	0.89980936	1.36081743	0.08
Theil	0.90475583	0.49478579	0.90475583	1.31336892	0.11
Wtd.Theil	0.88632137	0.48823041	0.88632137	1.27379596	0.14
			b=4.0		
OLS	1.83858383	0.41962820	1.83858383	3.80001879	0.00
MML	0.05477724	0.37036264	0.05477724	0.37336320	0.90
LAD	1.67404258	0.56226772	1.67404258	3.36468625	0.11
WIN10	1.77616501	0.39456365	1.77616501	3.54932570	0.07
WIN20	1.74494708	0.40003735	1.74494708	3.44487762	0.09
LTS	1.71363401	0.48595095	1.71363401	3.42249250	0.10
Theil	1.71278596	0.43006590	1.71278596	3.36370158	0.11
Wtd.Theil	1.69523227	0.42437243	1.69523227	3.29818487	0.13
			b=6.0		
OLS	2.28246546	0.38689718	2.28246546	5.59654570	0.00
MML	0.06725699	0.37106678	0.06725699	0.37559029	0.93
LAD	2.11076283	0.50976229	2.11076283	4.96508217	0.11
WIN10	2.21693468	0.36464152	2.21693468	5.27944088	0.06
WIN20	2.17746019	0.36858928	2.17746019	5.10992241	0.09
LTS	2.13622952	0.44817752	2.13622952	5.01165390	0.10
Theil	2.14304304	0.38656768	2.14304304	4.97920132	0.11
Wtd.Theil	2.12521410	0.37856147	2.12521410	4.89509630	0.13

Table 5.4 Simulation Results for Population Slope (θ_1=1.0), n=20

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	1.00378525	0.00990308	0.00378525	0.00991741	0.00
MML	1.00328755	0.00793530	0.00328755	0.00794611	0.20
LAD	1.00818872	0.01163360	0.00818872	0.01170066	-0.18
WIN10	1.00390410	0.00872273	0.00390410	0.00873797	0.12
WIN20	1.00379121	0.00864780	0.00379121	0.00866217	0.13
LTS	1.00403941	0.01032578	0.00403941	0.01034210	-0.04
Theil	1.00091553	0.00869807	0.00091553	0.00869891	0.12
Wtd.Theil	1.00400138	0.00860551	0.00400138	0.00862152	0.13
			b=1.0		
OLS	1.00101125	0.00488663	0.00101125	0.00488765	0.00
MML	1.00090313	0.00456718	0.00090313	0.00456799	0.07
LAD	1.00439906	0.00633174	0.00439906	0.00635109	-0.30
WIN10	1.00077474	0.00462350	0.00077474	0.00462410	0.05
WIN20	1.00072992	0.00464339	0.00072992	0.00464392	0.05
LTS	1.00131536	0.00558088	0.00131536	0.00558261	-0.14
Theil	0.99840969	0.00476225	-0.00159031	0.00476478	0.03
Wtd.Theil	1.00056648	0.00467195	0.00056648	0.00467227	0.04
			b=2.0		
OLS	1.00068212	0.00336664	0.00068212	0.00336711	0.00
MML	1.00120294	0.00302550	0.00120294	0.00302695	0.10
LAD	1.00371110	0.00457900	0.00371110	0.00459277	-0.36
WIN10	1.00113630	0.00320116	0.00113630	0.00320245	0.05
WIN20	1.00118852	0.00323568	0.00118852	0.00323709	0.04
LTS	1.00114846	0.00390879	0.00114846	0.00391011	-0.16
Theil	0.99915916	0.00329082	-0.00084084	0.00329152	0.02
Wtd.Theil	1.00094259	0.00322122	0.00094259	0.00322211	0.04
			b=4.0		
OLS	0.99932957	0.00295673	-0.00067043	0.00295718	0.00
MML	0.99918562	0.00236030	-0.00081438	0.00236097	0.20
LAD	1.00140953	0.00393521	0.00140953	0.00393719	-0.33
WIN10	0.99924999	0.00275742	-0.00075001	0.00275798	0.07
WIN20	0.99890184	0.00277408	-0.00109816	0.00277528	0.06
LTS	0.99843144	0.00335407	-0.00156856	0.00335653	-0.14
Theil	0.99746454	0.00276144	-0.00253546	0.00276786	0.06
Wtd.Theil	0.99907380	0.00271429	-0.00092620	0.00271514	0.08
			b=6.0		
OLS	1.00019443	0.00269123	0.00019443	0.00269127	0.00
MML	1.00013745	0.00208156	0.00013745	0.00208157	0.23
LAD	1.00153196	0.00355300	0.00153196	0.00355535	-0.32
WIN10	0.99987429	0.00249447	-0.00012571	0.00249449	0.07
WIN20	0.99989915	0.00252427	-0.00010085	0.00252428	0.06
LTS	1.00002146	0.00313086	0.00002146	0.00313086	-0.16
Theil	0.99818456	0.00246528	-0.00181544	0.00246858	0.08
Wtd.Theil	0.99985164	0.00241485	-0.00014836	0.00241487	0.10

Table 5.5 Simulation Results for Population *y*-intercept (θ_0=0.0), n=50

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	-1.38284743	0.56084418	-1.38284743	2.47311115	0.00
MML	-0.00152372	0.43467537	-0.00152372	0.43467769	0.82
LAD	-1.10082912	0.63566935	-1.10082912	1.84749413	0.25
WIN10	-1.26426291	0.48408639	-1.26426291	2.08244705	0.16
WIN20	-1.23468959	0.47832555	-1.23468959	2.00278401	0.19
LTS	-1.14312136	0.56339777	-1.14312136	1.87012422	0.24
Theil	-1.09844017	0.51189435	-1.09844017	1.71846521	0.31
Wtd.Theil	-1.10105491	0.50584912	-1.10105491	1.71817100	0.31
			b=1.0		
OLS	0.00352276	0.27417752	0.00352276	0.27418992	0.00
MML	0.00095244	0.25251332	0.00095244	0.25251421	0.08
LAD	-0.00456360	0.34581143	-0.00456360	0.34583226	-0.26
WIN10	0.00066642	0.25587413	0.00066642	0.25587457	0.07
WIN20	0.00110313	0.25404704	0.00110313	0.25404826	0.07
LTS	0.00160311	0.30169332	0.00160311	0.30169588	-0.10
Theil	-0.00456038	0.27905077	-0.00456038	0.27907157	-0.02
Wtd.Theil	-0.00597929	0.27712980	-0.00597929	0.27716556	-0.01
			b=2.0		
OLS	1.00050163	0.17853022	1.00050163	1.17953372	0.00
MML	0.00992685	0.16355318	0.00992685	0.16365172	0.86
LAD	0.88920945	0.23897938	0.88920945	1.02967286	0.13
WIN10	0.95236641	0.16855441	0.95236641	1.07555616	0.09
WIN20	0.94230467	0.16928717	0.94230467	1.05722523	0.10
LTS	0.90732235	0.20956881	0.90732235	1.03280270	0.12
Theil	0.88780248	0.18464047	0.88780248	0.97283369	0.18
Wtd.Theil	0.88740474	0.18260407	0.88740474	0.97009122	0.18
			b=4.0		
OLS	1.82753181	0.15858567	1.82753181	3.49845815	0.00
MML	0.01639831	0.14329638	0.01639831	0.14356528	0.96
LAD	1.66977441	0.20121989	1.66977441	2.98936653	0.15
WIN10	1.76001430	0.14562000	1.76001430	3.24327040	0.07
WIN20	1.74273443	0.14697777	1.74273443	3.18410110	0.09
LTS	1.68400753	0.18568318	1.68400753	3.02156448	0.14
Theil	1.67046666	0.15934859	1.67046666	2.94980741	0.16
Wtd.Theil	1.67058027	0.15718254	1.67058027	2.94802094	0.16
			b=6.0		
OLS	2.27176499	0.15472563	2.27176499	5.31564188	0.00
MML	0.01765234	0.14187641	0.01765234	0.14218801	0.97
LAD	2.10852599	0.19834721	2.10852599	4.64422894	0.13
WIN10	2.19787216	0.14219321	2.19787216	4.97283506	0.06
WIN20	2.17958164	0.14328554	2.17958164	4.89386177	0.08
LTS	2.11546707	0.17825735	2.11546707	4.65345812	0.12
Theil	2.09887052	0.15174763	2.09887052	4.55700493	0.14
Wtd.Theil	2.10001135	0.15080361	2.10001135	4.56085110	0.14

Table 5.6 Simulation Results for Population Slope (θ_1=1.0), n=50

b=0.5

methods	mean	variance	bias	MSE	RMSE
OLS	1.00054276	0.00065847	0.00054276	0.00065877	0.00
MML	1.00040746	0.00051141	0.00040746	0.00051157	0.22
LAD	1.00035143	0.00075797	0.00035143	0.00075809	-0.15
WIN10	1.00047839	0.00057637	0.00047839	0.00057660	0.13
WIN20	1.00044549	0.00057005	0.00044549	0.00057025	0.13
LTS	1.00055623	0.00067358	0.00055623	0.00067389	-0.02
Theil	1.00042427	0.00056142	0.00042427	0.00056160	0.15
Wtd.Theil	1.00052345	0.00055632	0.00052345	0.00055659	0.16

b=1.0

methods	mean	variance	bias	MSE	RMSE
OLS	1.00002897	0.00032064	0.00002897	0.00032064	0.00
MML	1.00010550	0.00029658	0.00010550	0.00029659	0.08
LAD	1.00010693	0.00039900	0.00010693	0.00039901	-0.24
WIN10	1.00012457	0.00030073	0.00012457	0.00030075	0.06
WIN20	1.00011516	0.00029613	0.00011516	0.00029614	0.08
LTS	1.00011885	0.00034900	0.00011885	0.00034902	-0.09
Theil	1.00012863	0.00030196	0.00012863	0.00030198	0.06
Wtd.Theil	1.00015545	0.00030104	0.00015545	0.00030107	0.06

b=2.0

methods	mean	variance	bias	MSE	RMSE
OLS	1.00001740	0.00021249	0.00001740	0.00021249	0.00
MML	0.99997532	0.00019142	-0.00002468	0.00019142	0.10
LAD	0.99993181	0.00029203	-0.00006819	0.00029204	-0.37
WIN10	0.99998546	0.00019990	-0.00001454	0.00019990	0.06
WIN20	0.99991757	0.00020262	-0.00008243	0.00020263	0.05
LTS	0.99964923	0.00025154	-0.00035077	0.00025166	-0.18
Theil	0.99990326	0.00020199	-0.00009674	0.00020200	0.05
Wtd.Theil	0.99990892	0.00019713	-0.00009108	0.00019714	0.07

b=4.0

methods	mean	variance	bias	MSE	RMSE
OLS	1.00043142	0.00018816	0.00043142	0.00018834	0.00
MML	1.00013518	0.00014607	0.00013518	0.00014609	0.22
LAD	1.00047255	0.00023626	0.00047255	0.00023648	-0.26
WIN10	1.00037026	0.00016866	0.00037026	0.00016880	0.10
WIN20	1.00039446	0.00017048	0.00039446	0.00017063	0.09
LTS	1.00048137	0.00021860	0.00048137	0.00021883	-0.16
Theil	1.00037861	0.00016594	0.00037861	0.00016608	0.12
Wtd.Theil	1.00037503	0.00016540	0.00037503	0.00016555	0.12

b=6.0

methods	mean	variance	bias	MSE	RMSE
OLS	1.00044525	0.00018154	0.00044525	0.00018174	0.00
MML	1.00028348	0.00013762	0.00028348	0.00013770	0.24
LAD	1.00011241	0.00023273	0.00011241	0.00023275	-0.28
WIN10	1.00041771	0.00016705	0.00041771	0.00016723	0.08
WIN20	1.00039744	0.00016677	0.00039744	0.00016693	0.08
LTS	1.00045383	0.00021114	0.00045383	0.00021134	-0.16
Theil	1.00042760	0.00016129	0.00042760	0.00016147	0.11
Wtd.Theil	1.00037038	0.00015835	0.00037038	0.00015848	0.13

Table 5.7 Simulation Results for Population y-intercept (θ_0=0.0), n=100

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	-1.39965451	0.26402918	-1.39965451	2.22306180	0.00
MML	-0.01836704	0.20235375	-0.01836704	0.20269109	0.91
LAD	-1.11568105	0.27979252	-1.11568105	1.52453673	0.31
WIN10	-1.30319607	0.23383625	-1.30319607	1.93215621	0.13
WIN20	-1.27647114	0.22959039	-1.27647114	1.85896897	0.16
LTS	-1.14494061	0.26646733	-1.14494061	1.57735634	0.29
Theil	-1.11357152	0.24186532	-1.11357152	1.48190689	0.33
Wtd.Theil	-1.11471128	0.24258696	-1.11471128	1.48516822	0.33
			b=1.0		
OLS	0.01006219	0.12918240	0.01006219	0.12928365	0.00
MML	0.00799097	0.11829767	0.00799097	0.11836153	0.08
LAD	0.00471181	0.16145238	0.00471181	0.16147459	-0.25
WIN10	0.00857508	0.12012789	0.00857508	0.12020142	0.07
WIN20	0.00814378	0.11897530	0.00814378	0.11904161	0.08
LTS	0.00532338	0.14199577	0.00532338	0.14202411	-0.10
Theil	0.01035039	0.13174637	0.01035039	0.13185349	-0.02
Wtd.Theil	0.00911384	0.13164656	0.00911384	0.13172962	-0.02
			b=2.0		
OLS	0.99967134	0.08896463	0.99967134	1.08830738	0.00
MML	0.00899292	0.08385907	0.00899292	0.08393995	0.92
LAD	0.88981199	0.11975086	0.88981199	0.91151625	0.16
WIN10	0.95935357	0.08306799	0.95935357	1.00342727	0.08
WIN20	0.94958776	0.08268750	0.94958776	0.98440444	0.10
LTS	0.90224504	0.10221050	0.90224504	0.91625661	0.16
Theil	0.88684011	0.09070868	0.88684011	0.87719405	0.19
Wtd.Theil	0.88573796	0.09066300	0.88573796	0.87519473	0.20
			b=4.0		
OLS	1.85146105	0.07859017	1.85146105	3.50649810	0.00
MML	0.03063465	0.07090033	0.03063465	0.07183881	0.98
LAD	1.68634045	0.10028265	1.68634045	2.94402671	0.16
WIN10	1.79211688	0.07331807	1.79211688	3.28500104	0.06
WIN20	1.77637684	0.07319573	1.77637684	3.22871041	0.08
LTS	1.70520496	0.08712067	1.70520496	2.99484468	0.15
Theil	1.68799376	0.07632731	1.68799376	2.92565036	0.17
Wtd.Theil	1.68809986	0.07639490	1.68809986	2.92607594	0.17
			b=6.0		
OLS	2.28320193	0.07380101	2.28320193	5.28681231	0.00
MML	0.01479289	0.06830579	0.01479289	0.06852462	0.99
LAD	2.10547471	0.09950072	2.10547471	4.53252459	0.14
WIN10	2.22366595	0.07009426	2.22366595	5.01478434	0.05
WIN20	2.20598507	0.07025642	2.20598507	4.93662643	0.06
LTS	2.12316489	0.08913086	2.12316489	4.59696007	0.13
Theil	2.10666060	0.07464702	2.10666060	4.51266575	0.15
Wtd.Theil	2.10689688	0.07472091	2.10689688	4.51373529	0.15

Table 5.8 Simulation Results for Population Slope (θ_1=1.0) , n=100

methods	mean	variance	bias	MSE	RMSE
			b=0.5		
OLS	1.00036514	0.00007616	0.00036514	0.00007630	0.00
MML	1.00025415	0.00005724	0.00025415	0.00005731	0.25
LAD	1.00029850	0.00007722	0.00029850	0.00007731	-0.01
WIN10	1.00033963	0.00006573	0.00033963	0.00006584	0.14
WIN20	1.00032222	0.00006474	0.00032222	0.00006484	0.15
LTS	1.00023198	0.00007694	0.00023198	0.00007700	-0.01
Theil	1.00025487	0.00006130	0.00025487	0.00006136	0.20
Wtd.Theil	1.00028944	0.00006127	0.00028944	0.00006136	0.20
			b=1.0		
OLS	0.99989992	0.00003870	-0.00010008	0.00003871	0.00
MML	0.99992180	0.00003583	-0.00007820	0.00003584	0.07
LAD	1.00003541	0.00004820	0.00003541	0.00004821	-0.25
WIN10	0.99991167	0.00003641	-0.00008833	0.00003642	0.06
WIN20	0.99991930	0.00003618	-0.00008070	0.00003618	0.07
LTS	0.99997717	0.00004306	-0.00002283	0.00004306	-0.11
Theil	0.99989498	0.00003645	-0.00010502	0.00003646	0.06
Wtd.Theil	0.99992186	0.00003620	-0.00007814	0.00003621	0.07
			b=2.0		
OLS	1.00001836	0.00002761	0.00001836	0.00002761	0.00
MML	1.00001097	0.00002472	0.00001097	0.00002472	0.11
LAD	0.99988222	0.00003566	-0.00011778	0.00003567	-0.29
WIN10	1.00000334	0.00002592	0.00000334	0.00002592	0.06
WIN20	0.99999011	0.00002587	-0.00000989	0.00002587	0.06
LTS	0.99989790	0.00003179	-0.00010210	0.00003180	-0.15
Theil	0.99998367	0.00002542	-0.00001633	0.00002542	0.08
Wtd.Theil	1.00000679	0.00002544	0.00000679	0.00002544	0.08
			b=4.0		
OLS	0.99974108	0.00002381	-0.00025892	0.00002388	0.00
MML	0.99974775	0.00001876	-0.00025225	0.00001883	0.21
LAD	0.99979413	0.00003116	-0.00020587	0.00003121	-0.31
WIN10	0.99976575	0.00002203	-0.00023425	0.00002208	0.08
WIN20	0.99976569	0.00002215	-0.00023431	0.00002221	0.07
LTS	0.99967504	0.00002668	-0.00032496	0.00002679	-0.12
Theil	0.99976581	0.00002137	-0.00023419	0.00002142	0.10
Wtd.Theil	0.99976468	0.00002136	-0.00023532	0.00002142	0.10
			b=6.0		
OLS	0.99988520	0.00002118	-0.00011480	0.00002119	0.00
MML	0.99985820	0.00001569	-0.00014180	0.00001571	0.26
LAD	0.99994510	0.00002805	-0.00005490	0.00002805	-0.32
WIN10	0.99984723	0.00001960	-0.00015277	0.00001962	0.07
WIN20	0.99985349	0.00001973	-0.00014651	0.00001975	0.07
LTS	0.99993420	0.00002616	-0.00006580	0.00002617	-0.24
Theil	0.99987119	0.00001873	-0.00012881	0.00001875	0.12
Wtd.Theil	0.99987131	0.00001875	-0.00012869	0.00001877	0.11

Table 5.9 Simulation Results for Population y-intercept (θ_0=0.0), n=10

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	1.82341588	0.87857330	1.82341588	4.20341873	0.00
MML	0.10407335	0.84159422	0.10407335	0.85242546	0.80
LAD	1.72297597	1.22348714	1.72297597	4.19213343	0.00
WIN10	1.77233100	0.87625277	1.77233100	4.01740980	0.04
WIN20	1.74737322	0.90057725	1.74737322	3.95389032	0.06
LTS	1.71464789	1.09964919	1.71464789	4.03966665	0.04
Theil	1.70249498	0.94470257	1.70249498	3.84319162	0.09
Wtd.Theil	1.70245922	0.92452508	1.70245922	3.82289243	0.09
Model (2)					
OLS	1.84259415	0.90246361	1.84259415	4.29761696	0.00
MML	0.11966418	0.84494877	0.11966418	0.85926831	0.80
LAD	1.72820079	1.23171306	1.72820079	4.21839094	0.02
WIN10	1.79290366	0.88244021	1.79290366	4.09694386	0.05
WIN20	1.76531780	0.90781879	1.76531780	4.02416563	0.06
LTS	1.72666311	1.10567868	1.72666311	4.08704424	0.05
Theil	1.72281265	0.94396216	1.72281265	3.91204548	0.09
Wtd.Theil	1.72194803	0.92164069	1.72194803	3.88674569	0.10
Model (3)					
OLS	2.38821244	3.64697790	2.38821244	9.35053635	0.00
MML	-0.36438006	1.69677460	-0.36438006	1.82954741	0.80
LAD	1.87150836	1.75552189	1.87150836	5.25806522	0.44
WIN10	2.02908421	1.40248132	2.02908421	5.51966381	0.41
WIN20	1.98960865	1.50317454	1.98960865	5.46171713	0.42
LTS	1.89602494	1.45324254	1.89602494	5.04815292	0.46
Theil	1.86823869	1.41042519	1.86823869	4.90074110	0.48
Wtd.Theil	1.87164664	1.41372168	1.87164664	4.91678286	0.47
Model (4)					
OLS	2.38949537	3.52450705	2.38949537	9.23419476	0.00
MML	-0.30060887	1.87900162	-0.30060887	1.96936727	0.79
LAD	1.91173530	1.91526592	1.91173530	5.56999779	0.40
WIN10	2.12411523	1.95953989	2.12411523	6.47140551	0.30
WIN20	2.04301763	1.85043538	2.04301763	6.02435637	0.35
LTS	1.92874980	1.88336492	1.92874980	5.60344076	0.39
Theil	1.89468884	1.53403687	1.89468884	5.12388277	0.45
Wtd.Theil	1.89600134	1.61890531	1.89600134	5.21372652	0.44
Model (5)					
OLS	1.76066971	0.91065747	1.76066971	4.01061535	0.00
MML	0.06763975	0.80795312	0.06763975	0.81252825	0.80
LAD	1.63014996	1.21501935	1.63014996	3.87240815	0.03
WIN10	1.70242560	0.87352937	1.70242560	3.77178240	0.06
WIN20	1.67110336	0.90690970	1.67110336	3.69949603	0.08
LTS	1.63041520	1.09978795	1.63041520	3.75804162	0.06
Theil	1.62314343	0.93718189	1.62314343	3.57177639	0.11
Wtd.Theil	1.62748432	0.93124384	1.62748432	3.57994914	0.11

Table 5.10 Simulation Results for Population Slope (θ_I=1.0), n=10

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	1.00082338	0.02274927	0.00082338	0.02274995	0.00
MML	1.00029504	0.01947838	0.00029504	0.01947846	0.14
LAD	0.99803877	0.03240623	-0.00196123	0.03241008	-0.43
WIN10	1.00076485	0.02264601	0.00076485	0.02264659	0.01
WIN20	1.00011504	0.02322487	0.00011504	0.02322488	-0.02
LTS	0.99965769	0.02846733	-0.00034231	0.02846745	-0.25
Theil	1.00070703	0.02321882	0.00070703	0.02321932	-0.02
Wtd.Theil	1.00091350	0.02269426	0.00091350	0.02269509	0.00
Model (2)					
OLS	0.99784297	0.02327893	-0.00215703	0.02328358	0.00
MML	0.99847400	0.01951578	-0.00152600	0.01951811	0.16
LAD	0.99721116	0.03284302	-0.00278884	0.03285079	-0.41
WIN10	0.99735731	0.02291116	-0.00264269	0.02291814	0.02
WIN20	0.99687117	0.02347238	-0.00312883	0.02348217	-0.01
LTS	0.99699426	0.02878170	-0.00300574	0.02879073	-0.24
Theil	0.99732453	0.02340834	-0.00267547	0.02341550	-0.01
Wtd.Theil	0.99747282	0.02278687	-0.00252718	0.02279325	0.02
Model (3)					
OLS	1.00001514	0.10602017	0.00001514	0.10602017	0.00
MML	1.00037241	0.03606265	0.00037241	0.03606279	0.66
LAD	1.00084770	0.04685701	0.00084770	0.04685773	0.56
WIN10	1.00128305	0.03739742	0.00128305	0.03739906	0.65
WIN20	1.00103474	0.04066702	0.00103474	0.04066809	0.62
LTS	1.00050485	0.03844330	0.00050485	0.03844355	0.64
Theil	1.00089288	0.03654171	0.00089288	0.03654251	0.66
Wtd.Theil	1.00049984	0.03662110	0.00049984	0.03662135	0.66
Model (4)					
OLS	0.99873722	0.09092926	-0.00126278	0.09093086	0.00
MML	0.99980682	0.03758510	-0.00019318	0.03758514	0.59
LAD	0.99710566	0.04996851	-0.00289434	0.04997689	0.45
WIN10	0.99836892	0.04777493	-0.00163108	0.04777759	0.48
WIN20	0.99755496	0.04589827	-0.00244504	0.04590425	0.50
LTS	0.99838340	0.04884230	-0.00161660	0.04884492	0.46
Theil	0.99843699	0.03812905	-0.00156301	0.03813150	0.58
Wtd.Theil	0.99881095	0.04057294	-0.00118905	0.04057435	0.55
Model (5)					
OLS	0.99803323	0.02359288	-0.00196677	0.02359675	0.00
MML	0.99774587	0.01896220	-0.00225413	0.01896728	0.20
LAD	0.99836212	0.03227962	-0.00163788	0.03228230	-0.37
WIN10	0.99812001	0.02271135	-0.00187999	0.02271489	0.04
WIN20	0.99790508	0.02356682	-0.00209492	0.02357121	0.00
LTS	0.99770099	0.02873067	-0.00229901	0.02873595	-0.22
Theil	0.99890429	0.02293480	-0.00109571	0.02293600	0.03
Wtd.Theil	0.99831951	0.02273407	-0.00168049	0.02273689	0.04

Table 5.11 Simulation Results for Population y-intercept (θ_0=0.0), n=20

Model (1)

methods	mean	variance	bias	MSE	RMSE
OLS	1.84241402	0.42172620	1.84241402	3.81621552	0.00
MML	0.05831144	0.38315803	0.05831144	0.38655826	0.90
LAD	1.67163384	0.55530876	1.67163384	3.34966850	0.12
WIN10	1.77728617	0.40321496	1.77728617	3.56196117	0.07
WIN20	1.74057889	0.41011870	1.74057889	3.43973351	0.10
LTS	1.70207143	0.50135076	1.70207143	3.39839792	0.11
Theil	1.71601546	0.43175936	1.71601546	3.37646842	0.12
Wtd.Theil	1.69771314	0.42580917	1.69771314	3.30803919	0.13

Model (2)

methods	mean	variance	bias	MSE	RMSE
OLS	1.83497846	0.39892814	1.83497846	3.76607418	0.00
MML	0.04898102	0.36676607	0.04898102	0.36916521	0.90
LAD	1.66658247	0.53035730	1.66658247	3.30785441	0.12
WIN10	1.76829398	0.37953293	1.76829398	3.50639653	0.07
WIN20	1.73378599	0.38452888	1.73378599	3.39054275	0.10
LTS	1.69674182	0.47555029	1.69674182	3.35448313	0.11
Theil	1.71048617	0.41155654	1.71048617	3.33731937	0.11
Wtd.Theil	1.69074059	0.40190458	1.69074059	3.26050830	0.13

Model (3)

methods	mean	variance	bias	MSE	RMSE
OLS	2.37819266	1.55426908	2.37819266	7.21006966	0.00
MML	-0.36361694	0.69063264	-0.36361694	0.82284993	0.89
LAD	1.79352653	0.71799529	1.79352653	3.93473268	0.45
WIN10	2.01075649	0.62275946	2.01075649	4.66590118	0.35
WIN20	1.93510580	0.60929751	1.93510580	4.35393190	0.40
LTS	1.91223490	0.73670071	1.91223490	4.39334297	0.39
Theil	1.83835900	0.57988888	1.83835900	3.95945263	0.45
Wtd.Theil	1.81722736	0.57316029	1.81722736	3.87547565	0.46

Model (4)

methods	mean	variance	bias	MSE	RMSE
OLS	2.39029217	1.62804925	2.39029217	7.34154606	0.00
MML	-0.34465533	0.79335397	-0.34465533	0.91214126	0.88
LAD	1.83152187	0.79431540	1.83152187	4.14878798	0.44
WIN10	2.07252312	0.79956567	2.07252312	5.09491777	0.31
WIN20	1.96993959	0.69521302	1.96993959	4.57587481	0.38
LTS	1.91653275	0.82153022	1.91653275	4.49462795	0.39
Theil	1.86028206	0.62205631	1.86028206	4.08270550	0.44
Wtd.Theil	1.84182572	0.62561566	1.84182572	4.01793766	0.45

Model (5)

methods	mean	variance	bias	MSE	RMSE
OLS	1.73643553	0.40175638	1.73643553	3.41696477	0.00
MML	-0.01596100	0.34671524	-0.01596100	0.34696999	0.90
LAD	1.54505444	0.54832268	1.54505444	2.93551588	0.14
WIN10	1.66390181	0.37758678	1.66390181	3.14615607	0.08
WIN20	1.62132883	0.38408977	1.62132883	3.01279688	0.12
LTS	1.57139087	0.48204103	1.57139087	2.95131040	0.14
Theil	1.58298039	0.40500310	1.58298039	2.91083002	0.15
Wtd.Theil	1.56555951	0.40121266	1.56555951	2.85218930	0.17

Table 5.12 Simulation Results for Population Slope (θ_1=1.0), n=20

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	0.99899119	0.00289677	-0.00100881	0.00289779	0.00
MML	0.99906760	0.00239106	-0.00093240	0.00239193	0.18
LAD	1.00182581	0.00387190	0.00182581	0.00387523	-0.34
WIN10	0.99918479	0.00274359	-0.00081521	0.00274426	0.05
WIN20	0.99926454	0.00277240	-0.00073546	0.00277294	0.04
LTS	0.99928057	0.00342555	-0.00071943	0.00342606	-0.18
Theil	0.99752706	0.00272878	-0.00247294	0.00273490	0.06
Wtd.Theil	0.99925691	0.00269779	-0.00074309	0.00269834	0.07
Model (2)					
OLS	0.99947792	0.00279693	-0.00052208	0.00279720	0.00
MML	0.99933279	0.00225583	-0.00066721	0.00225628	0.19
LAD	1.00203979	0.00378384	0.00203979	0.00378800	-0.35
WIN10	0.99971396	0.00263619	-0.00028604	0.00263627	0.06
WIN20	0.99959052	0.00267510	-0.00040948	0.00267527	0.04
LTS	0.99965507	0.00336390	-0.00034493	0.00336402	-0.20
Theil	0.99746329	0.00264601	-0.00253671	0.00265245	0.05
Wtd.Theil	0.99933320	0.00258749	-0.00066680	0.00258793	0.08
Model (3)					
OLS	1.00153363	0.01227777	0.00153363	0.01228012	0.00
MML	1.00034237	0.00369254	0.00034237	0.00369266	0.70
LAD	1.00338638	0.00523196	0.00338638	0.00524343	0.57
WIN10	1.00047517	0.00442591	0.00047517	0.00442613	0.64
WIN20	1.00059831	0.00439403	0.00059831	0.00439439	0.64
LTS	1.00071979	0.00540699	0.00071979	0.00540751	0.56
Theil	0.99852324	0.00391649	-0.00147676	0.00391867	0.68
Wtd.Theil	1.00049186	0.00388176	0.00049186	0.00388200	0.68
Model (4)					
OLS	0.99988544	0.01146949	-0.00011456	0.01146950	0.00
MML	1.00052583	0.00390226	0.00052583	0.00390253	0.66
LAD	1.00147700	0.00565080	0.00147700	0.00565298	0.51
WIN10	1.00014997	0.00524104	0.00014997	0.00524106	0.54
WIN20	0.99985701	0.00481144	-0.00014299	0.00481146	0.58
LTS	1.00029492	0.00579628	0.00029492	0.00579637	0.50
Theil	0.99830461	0.00410964	-0.00169539	0.00411251	0.64
Wtd.Theil	1.00013423	0.00413948	0.00013423	0.00413950	0.64
Model (5)					
OLS	1.00070298	0.00285154	0.00070298	0.00285204	0.00
MML	1.00092363	0.00218044	0.00092363	0.00218129	0.24
LAD	1.00335610	0.00391336	0.00335610	0.00392462	-0.38
WIN10	1.00087905	0.00266696	0.00087905	0.00266773	0.07
WIN20	1.00099003	0.00271432	0.00099003	0.00271530	0.05
LTS	1.00146508	0.00343791	0.00146508	0.00344005	-0.21
Theil	0.99934036	0.00260590	-0.00065964	0.00260633	0.09
Wtd.Theil	1.00103009	0.00260286	0.00103009	0.00260392	0.09

Table 5.13 Simulation Results for Population *y*-intercept (θ_0=0.0), *n*=50

Model (1)

methods	mean	variance	bias	MSE	RMSE
OLS	1.83162248	0.16490535	1.83162248	3.51974630	0.00
MML	0.01227662	0.14784478	0.01227662	0.14799549	0.96
LAD	1.67247653	0.20917244	1.67247653	3.00635028	0.15
WIN10	1.76242244	0.15093480	1.76242244	3.25706768	0.08
WIN20	1.74404180	0.15229693	1.74404180	3.19397879	0.09
LTS	1.67921448	0.19058892	1.67921448	3.01035023	0.15
Theil	1.66986835	0.16012195	1.66986835	2.94858217	0.16
Wtd.Theil	1.67087317	0.15928252	1.67087317	2.95109963	0.16

Model (2)

OLS	1.82531703	0.16126959	1.82531703	3.49305177	0.00
MML	0.01251316	0.14113364	0.01251316	0.14129022	0.96
LAD	1.66783941	0.20286505	1.66783941	2.98455334	0.15
WIN10	1.75409806	0.14862959	1.75409806	3.22548962	0.08
WIN20	1.73764074	0.14980407	1.73764074	3.16919947	0.09
LTS	1.67938280	0.18372254	1.67938280	3.00404906	0.14
Theil	1.66904056	0.15683734	1.66904056	2.94253373	0.16
Wtd.Theil	1.66802931	0.15589558	1.66802931	2.93821740	0.16

Model (3)

OLS	2.37208986	0.55625153	2.37208986	6.18306208	0.00
MML	-0.35506275	0.24410290	-0.35506275	0.37017244	0.94
LAD	1.81084907	0.25537592	1.81084907	3.53455019	0.43
WIN10	2.01921368	0.23540896	2.01921368	4.31263304	0.30
WIN20	1.96912169	0.22901362	1.96912169	4.10645390	0.34
LTS	1.90693963	0.29257521	1.90693963	3.92899394	0.37
Theil	1.80852342	0.20809075	1.80852342	3.47884774	0.44
Wtd.Theil	1.80855644	0.20747589	1.80855644	3.47835231	0.44

Model (4)

OLS	2.38219881	0.59725171	2.38219881	6.27212286	0.00
MML	-0.32591116	0.24783514	-0.32591116	0.35405323	0.94
LAD	1.83251023	0.27130485	1.83251023	3.62939858	0.42
WIN10	2.05317426	0.27252531	2.05317426	4.48804998	0.28
WIN20	1.99340653	0.25036865	1.99340653	4.22403812	0.33
LTS	1.91490901	0.29558477	1.91490901	3.96246123	0.37
Theil	1.81860471	0.22080489	1.81860471	3.52812791	0.44
Wtd.Theil	1.81888676	0.21960624	1.81888676	3.52795529	0.44

Model (5)

OLS	1.73327935	0.15354824	1.73327935	3.15780544	0.00
MML	-0.05003753	0.12317045	-0.05003753	0.12567420	0.96
LAD	1.54645932	0.19634642	1.54645932	2.58788276	0.18
WIN10	1.65616655	0.14084049	1.65616655	2.88372803	0.09
WIN20	1.63547802	0.14309300	1.63547802	2.81788135	0.11
LTS	1.55805504	0.18097806	1.55805504	2.60851359	0.17
Theil	1.55784202	0.14805384	1.55784202	2.57492566	0.19
Wtd.Theil	1.55786788	0.14717667	1.55786788	2.57412910	0.19

Table 5.14 Summary Measures for Estimating Population Slope (θ_l=1.0), n=50

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	1.00010026	0.00018855	0.00010026	0.00018856	0.00
MML	1.00008583	0.00014566	0.00008583	0.00014567	0.23
LAD	1.00008857	0.00024521	0.00008857	0.00024522	-0.30
WIN10	1.00014544	0.00017123	0.00014544	0.00017125	0.09
WIN20	1.00016713	0.00017520	0.00016713	0.00017522	0.07
LTS	1.00022185	0.00022336	0.00022185	0.00022341	-0.19
Theil	1.00015306	0.00016988	0.00015306	0.00016991	0.10
Wtd.Theil	1.00011683	0.00016901	0.00011683	0.00016902	0.10
Model (2)					
OLS	1.00039244	0.00018430	0.00039244	0.00018446	0.00
MML	1.00035739	0.00014094	0.00035739	0.00014107	0.24
LAD	1.00056422	0.00023806	0.00056422	0.00023838	-0.29
WIN10	1.00054801	0.00016677	0.00054801	0.00016707	0.09
WIN20	1.00052798	0.00016608	0.00052798	0.00016636	0.10
LTS	1.00049222	0.00020885	0.00049222	0.00020909	-0.13
Theil	1.00045562	0.00016371	0.00045562	0.00016392	0.11
Wtd.Theil	1.00050092	0.00016267	0.00050092	0.00016292	0.12
Model (3)					
OLS	1.00016212	0.00069994	0.00016212	0.00069997	0.00
MML	1.00011134	0.00021369	0.00011134	0.00021370	0.70
LAD	1.00004721	0.00030717	0.00004721	0.00030717	0.56
WIN10	1.00004983	0.00027548	0.00004983	0.00027548	0.61
WIN20	1.00005984	0.00027205	0.00005984	0.00027205	0.61
LTS	1.00035071	0.00035777	0.00035071	0.00035789	0.49
Theil	1.00002861	0.00023147	0.00002861	0.00023147	0.67
Wtd.Theil	1.00001752	0.00023179	0.00001752	0.00023179	0.67
Model (4)					
OLS	0.99960870	0.00069845	-0.00039130	0.00069861	0.00
MML	0.99933177	0.00021374	-0.00066823	0.00021419	0.69
LAD	0.99882877	0.00031659	-0.00117123	0.00031796	0.55
WIN10	0.99930054	0.00029677	-0.00069946	0.00029726	0.57
WIN20	0.99931216	0.00028947	-0.00068784	0.00028994	0.59
LTS	0.99942333	0.00036267	-0.00057667	0.00036300	0.48
Theil	0.99927235	0.00023144	-0.00072765	0.00023197	0.67
Wtd.Theil	0.99925345	0.00023582	-0.00074655	0.00023638	0.66
Model (5)					
OLS	1.00022197	0.00018221	0.00022197	0.00018225	0.00
MML	1.00027764	0.00013270	0.00027764	0.00013278	0.27
LAD	1.00072455	0.00023640	0.00072455	0.00023692	-0.30
WIN10	1.00031710	0.00016285	0.00031710	0.00016295	0.11
WIN20	1.00031328	0.00016645	0.00031328	0.00016655	0.09
LTS	1.00053120	0.00021594	0.00053120	0.00021622	-0.19
Theil	1.00026858	0.00016048	0.00026858	0.00016056	0.12
Wtd.Theil	1.00027061	0.00015649	0.00027061	0.00015656	0.14

Table 5.15 Simulation Results for Population *y*-intercept, (θ_0=0) *n*=100

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	1.82845223	0.07792812	1.82845223	3.42116570	0.00
MML	0.00564459	0.07347494	0.00564459	0.07350680	0.98
LAD	1.66871119	0.10485132	1.66871119	2.88944840	0.16
WIN10	1.77072644	0.07262221	1.77072644	3.20809436	0.06
WIN20	1.75581765	0.07285842	1.75581765	3.15575409	0.08
LTS	1.68971658	0.09189245	1.68971658	2.94703460	0.14
Theil	1.67227280	0.07919315	1.67227280	2.87568951	0.16
Wtd.Theil	1.67051733	0.07885517	1.67051733	2.86948323	0.16
Model (2)					
OLS	1.83111656	0.07460266	1.83111656	3.42759061	0.00
MML	0.00903944	0.06871808	0.00903944	0.06879979	0.98
LAD	1.66548645	0.09803158	1.66548645	2.87187672	0.16
WIN10	1.77216482	0.06885005	1.77216482	3.20941830	0.06
WIN20	1.75726902	0.06830402	1.75726902	3.15629840	0.08
LTS	1.68746817	0.08514012	1.68746817	2.93268895	0.14
Theil	1.67320144	0.07156049	1.67320144	2.87116361	0.16
Wtd.Theil	1.67213905	0.07147506	1.67213905	2.86752415	0.16
Model (3)					
OLS	2.39282465	0.30370972	2.39282465	6.02931929	0.00
MML	-0.34904075	0.11504465	-0.34904075	0.23687410	0.96
LAD	1.80048943	0.12318436	1.80048943	3.36494660	0.44
WIN10	2.08236599	0.13656811	2.08236599	4.47281647	0.26
WIN20	2.01625919	0.12128466	2.01625919	4.18658590	0.31
LTS	1.93016338	0.14624208	1.93016338	3.87177277	0.36
Theil	1.80647612	0.10336834	1.80647612	3.36672425	0.44
Wtd.Theil	1.80662310	0.10259851	1.80662310	3.36648560	0.44
Model (4)					
OLS	2.39431596	0.28977638	2.39431596	6.02252531	0.00
MML	-0.33194250	0.11926392	-0.33194250	0.22944973	0.96
LAD	1.82025754	0.13009319	1.82025754	3.44343066	0.43
WIN10	2.09439206	0.14509648	2.09439206	4.53157473	0.25
WIN20	2.02678323	0.12471370	2.02678323	4.23256397	0.30
LTS	1.92653561	0.14291726	1.92653561	3.85445666	0.36
Theil	1.81879234	0.09809802	1.81879234	3.40610361	0.43
Wtd.Theil	1.81848717	0.09849638	1.81848717	3.40539193	0.44
Model (5)					
OLS	1.75741363	0.07212670	1.75741363	3.16062927	0.00
MML	-0.04067939	0.05738617	-0.04067939	0.05904099	0.98
LAD	1.57093954	0.10100739	1.57093954	2.56885839	0.19
WIN10	1.69377458	0.06759924	1.69377458	2.93647146	0.07
WIN20	1.67333531	0.06807805	1.67333531	2.86812901	0.09
LTS	1.57797706	0.09162159	1.57797706	2.58163309	0.18
Theil	1.57574642	0.07436635	1.57574642	2.55734301	0.19
Wtd.Theil	1.57543635	0.07411832	1.57543635	2.55611801	0.19

Table 5.16 Simulation Results for Population Slope (θ_1=1.0), n=100

methods	mean	variance	bias	MSE	RMSE
Model (1)					
OLS	1.00024164	0.00002410	0.00024164	0.00002415	0.00
MML	1.00017786	0.00001903	0.00017786	0.00001907	0.21
LAD	1.00024247	0.00002991	0.00024247	0.00002997	-0.24
WIN10	1.00023460	0.00002199	0.00023460	0.00002204	0.09
WIN20	1.00022948	0.00002191	0.00022948	0.00002196	0.09
LTS	1.00013089	0.00002686	0.00013089	0.00002687	-0.11
Theil	1.00017178	0.00002113	0.00017178	0.00002116	0.12
Wtd.Theil	1.00020254	0.00002110	0.00020254	0.00002114	0.12
Model (2)					
OLS	0.99998099	0.00002174	-0.00001901	0.00002174	0.00
MML	1.00001729	0.00001752	0.00001729	0.00001752	0.19
LAD	1.00012374	0.00002977	0.00012374	0.00002979	-0.37
WIN10	0.99999189	0.00002014	-0.00000811	0.00002014	0.07
WIN20	0.99998057	0.00002010	-0.00001943	0.00002010	0.08
LTS	0.99993479	0.00002601	-0.00006521	0.00002602	-0.20
Theil	0.99998200	0.00001936	-0.00001800	0.00001936	0.11
Wtd.Theil	1.00000060	0.00001936	0.00000060	0.00001936	0.11
Model (3)					
OLS	0.99983180	0.00010051	-0.00016820	0.00010054	0.00
MML	0.99974793	0.00002562	-0.00025207	0.00002568	0.74
LAD	0.99988639	0.00003723	-0.00011361	0.00003725	0.63
WIN10	0.99978793	0.00004109	-0.00021207	0.00004113	0.59
WIN20	0.99976426	0.00003708	-0.00023574	0.00003714	0.63
LTS	0.99960756	0.00004699	-0.00039244	0.00004714	0.53
Theil	0.99973816	0.00002835	-0.00026184	0.00002841	0.72
Wtd.Theil	0.99974078	0.00002802	-0.00025922	0.00002809	0.72
Model (4)					
OLS	0.99983269	0.00008471	-0.00016731	0.00008474	0.00
MML	0.99989712	0.00002574	-0.00010288	0.00002575	0.70
LAD	0.99982631	0.00003922	-0.00017369	0.00003925	0.54
WIN10	0.99988866	0.00004041	-0.00011134	0.00004042	0.52
WIN20	0.99986488	0.00003660	-0.00013512	0.00003662	0.57
LTS	0.99984807	0.00004442	-0.00015193	0.00004444	0.48
Theil	0.99988830	0.00002818	-0.00011170	0.00002819	0.67
Wtd.Theil	0.99989480	0.00002822	-0.00010520	0.00002823	0.67
Model (5)					
OLS	0.99990410	0.00002195	-0.00009590	0.00002196	0.00
MML	0.99996275	0.00001606	-0.00003725	0.00001606	0.27
LAD	0.99998736	0.00003093	-0.00001264	0.00003093	-0.41
WIN10	0.99992800	0.00002013	-0.00007200	0.00002014	0.08
WIN20	0.99994045	0.00002032	-0.00005955	0.00002033	0.07
LTS	1.00002730	0.00002876	0.00002730	0.00002876	-0.31
Theil	0.99991834	0.00001946	-0.00008166	0.00001947	0.11
Wtd.Theil	0.99992836	0.00001934	-0.00007164	0.00001934	0.12

Slope Estimator Performance: Tables 5.2, 5.4, 5.6 and 5.8 give summary results for θ_1 under the Generalised Logistic distribution with b=0.5, 1.0, 2.0, 4.0 and 6.0 for sample sizes n=10, 20, 50 and 100, confirming the fact that deviations from normality cause OLS estimators to be poor estimators.

For negatively skewed cases (b<1.0), all the estimators have negligible bias and MML method, which gives small weights to extreme obsevations, demonstrates the strongest performance gains with 15%-25% decreases in MSE as compared to OLS. For small samples (n=10, n=20), Winsorized least squares regression has the second smallest MSE values. However, as sample size increases (n=50, n=100) the second place belongs to nonparametric Theil methods based on medians of pairwise slopes.

For symmetric cases (b=1.0), the performance of the estimators are similar. However, the reduction in MSE is less than the negatively skewed cases and the second place is taken by the Winsorized least squares method which applies smoothing techniques to decrease the effect of outlying observations.

For positively skewed cases (b>1.0), again the MML methodology has the highest performance and similar to the previous results Theil based and Winsorized least squares methods compete with eachother.

For most of the cases the LTS and LAD method show poor results with negative RMSE values. This situation for LAD method is not a surprise, because as it is mentioned in Chapter 4, this estimation procedure is suitable when the error terms come from heavy-tailed distributions.

The simulation results of θ_1 for the alternative models are given in the Tables 5.10, 5.12, 5.14 and 5.16 for sample sizes n=10, 20, 50 and 100 and from these results we can observe that the reduction in MSE values for n=50 and n=100 are higher than those for n=10 and n=20. Performance gains especially for Model (3) and Model (4) are higher for these cases and similar to the previous results, MML methodology exhibits the strongest performance, with 14%-74% reduction in MSE. However, for these cases, Theil based estimators seem to be the second best leaving the third place to Winsorized least squares estimators. Although the LAD and LTS slope estimators have higher MSE than the OLS estimators for Model (1), (2) and (5), they improve for Model (3) and Model (4) yielding a 48%-63% reduction in MSE.

Y-intercept Performance: Tables 5.1, 5.3, 5.5, 5.7, 5.9, 5.11, 5.13, and 5.15 give summary results for $\theta_{0.}$ It is easily seen that estimators other than the MML are substantially biased resulting in high MSE values. Therefore, for these cases they are generally not competitive. However, when b=1.0 (i.e. when the Generalised Logistic distribution turns out to be a symmetric distribution), the bias decreases especially for large sample sizes. For these cases it is easily seen that LAD, LTS, Theil and weighted Theil estimation methods show unreliable performance compared to OLS with negative RMSE values. The MML procedure has the first rank with 4%-8% reduction in MSE that is followed by the Winsorized least squares regression. Therefore, we can conclude that MML estimators are self-bias correcting and LAD, Winsorized and LTS, Theil's and weighted Theil's regression methods give reliable results for the situations where error terms are coming from symmetric distributions.

CHAPTER 6

CONCLUSION

The usual assumption for a linear regression model is that error terms have a normal distribution, which leads OLS estimation procedure to give good results. However, in real life it is nearly impossible to find a data set that satisfies the normality assumption. Since under these situations OLS estimators result in loss of efficiency, alternative regression techniques are needed. In this thesis, we study some robust and nonparametric procedures for a simple linear regression model when the error terms are coming from a Generalised Logistic distribution.

In Chapter 1, we give brief information about the simple linear regression and in Chapter 2, we explain the estimation procedures we investigate briefly and review the literature by mentioning the important studies about these estimation techniques.

OLS regression is discussed in Chapter 3, while the iterations for alternative techniques, MML, LAD, 10% and 20% Winsorized least squares, 20% LTS, Theil and weighted Theil regression, are explained in Chapter 4 with applications to a real life data sets.

Chapter 5 covers the simulation study for selected values of b and n, and for plausible alternatives. In order to measure the quality of the parameter estimation, summary measures for y-intercept and slope estimators are obtained. Looking at the summary tables for slope estimator (for b=0.5, 1.0, 2.0, 4.0, 6.0

and sample sizes n=10, 20, 50, 100), we see that for all the cases (i.e. negatively skewed, symmetric or positively skewed), the MML method demonstrates the strongest performance gains with 1%-25% decreases in MSE as compared to OLS. That is to say, they have negligible bias and the smallest MSE. These estimators are not only highly efficient for small sample sizes but also explicit functions of the sample observations, therefore, easy to compute. The second best results are obtained from Winsorized least squares and weighted Theil methods with relative reductions in MSE of 1-14% and 1%-20%, respectively. As sample size increases Theil based methods improve and get the second place. However, LAD and LTS regressions give the poorest slope estimators with negative RMSE values.

The simulation results of θ_1 for the alternative models for sample sizes n=10, 20, 50 and 100 are similar to the previous results, again MML methodology exhibits the strongest performance leaving a competitive struggle between Theil based and Winsorized least squares estimators for the second place. It is observed that outlier and mixture model show greater performance gains compared to the other three models and also for these models LAD and LTS slope estimators improve resulting in positive RMSE values.

The summary results for y-intercept estimator need more consideration. Other than the MML method, all estimation procedures show large bias which prove their inadequacy to reflect the true values of the parameters. Therefore, under these conditions these estimation methods are not competitive. However, for the cases b=1.0 (symmetric distribution), the bias decreases leading to comparable MSE values and again the results demonstrate that MML methodology has the largest RMSE and the second best place belongs to the Winsorized least squares regression.

The poor performance of OLS estimators under the nonnormal error distributions confirms our need for alternative methods. Therefore, before analyzing the data, we should first check for the outliers and then construct the necessary tests whether to see the underlying assumptions are satisfied. After this, we should conduct the appropriate estimation techniques. This study shows that for simple linear regression model with the Generalised Logistic error terms, MML estimation technique has the strongest performance and most reliable results.

REFERENCES

Agullo, J. (2001). A new algorithm for computing the least trimmed squares regression estimator. *Computational Statistics and Data Analysis,* 36 (4), 425-439.

Barnett, V. D. (1966). Evaluation of the maximum likelihood estimator when the likelihood equation has multiple roots. *Biometrika,* 53, 151-165.

Birkes, D., and Dodge, Y. (1993). *Alternative Methods of Regression.* New York, NY: Wiley.

Brown, B. M. (1980). Median estimates in simple linear regression. *Austral. J. Statist.,* 22 (2),154-165.

Castillo, E., Hadi, A. S., and Lacruz, B. (2001). Regression diagnostics for the least absolute deviations and the minimax methods. *Communications in Statistics- Theory and Methods*, 30 (6), 1197-1225.

Cizek, P., and Visek, J. A. (2000). Least trimmed squares. *Quantifikation und Simulation Okonomisher Prozesse/ Humboldt Universitat zu Berlin with number 00-53.*

Daniel, C. and Wood, F. S. (1971) *Fitting Equations to Data*, Wiley-Interscience, New York.

Draper, N. R., and Smith, H. (1966). *Applied Regression Analysis.* New York, NY: Wiley.

Guo, J. (2000). Robust estimation for the coefficient of a first order autoregressive process. *Communications in Statistics- Theory and Methods,* 29 (1), 55-66.

Hoo, K., Tvarlapati K., Piovoso, M., and Hajare, R. (2002). A method of robust multivariate outlier replacement. *Computers and Chemical Engineering,* 26 (1), 17-39.

Hossjer, O. (1995). Exact computation of the least trimmed squares estimate in simple linear-regression. *Computational Statistics and Data Analysis,* 19 (3), 265-282.

Hussain, S. S., and Sprent, P. (1983). Nonparametric Regression. *Journal of the Royal Statistical Society, series A,* 146, 182-191.

Islam, M. Q., Tiku, M. L., and Yıldırım, F. (2001). Non-normal regression I. Skew distributions. *Commun. Stat.-Theory Meth.,* 30 (6), 993-1020.

Jaeckel, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics,* 43, 1449-1458.

Lawrence, K., and Shier, D. (1981). A comparison of least-squares and least absolute deviation regression-models for estimating Weibull parameters. *Communications in Statistics Part B-Simulation and Computation,* 10 (3), 315-326.

Lee, K. R., Kapadia, C. H., and Dwight, B.B. (1980). On estimating the scale parameter of Rayleigh distribution from censored samples. *Statist Hefte,* 21, 14-20.

Mathew, T., and Nordstrom, K. (1993). Least squares and least absolute deviation procedures in approximately linear models. *Statistics and Probability Letters,* 16 (2), 153-158.

Montgomery, D. C., and Peck, E. A. (1991). *Introduction to Linear Regression Analysis.* New York, NY: Wiley, 2nd edition.

Myers, R. H. (1986). *Classical and Modern Regression with Applications.* PWS-Kent, Boston.

Nevitt, T., and Tam, H. P. (1998). A comparison of robust and nonparametric estimators under the simple linear regression model. *Multiple Linear Regression Viewpoints,* 25, 54-69.

Puthenpura, S., and Sinha, N. K. (1986). Modified maximum likelihood method for the robust estimation of system parameters from very noisy data. *Automatica,* 22, 231-235.

Rawlings, J. O., Pantula, S. G., and Dickey, D. A. (1998). *Applied Regression Analysis: A Research Tool.* New York, NY: Springer, 2nd edition.

Rivest, L. (1994). Statistical properties of Winsorized means for skewed distributions. *Biometrika,* 81 (2), 373-383.

Rousseeuw, P. J., and Leroy, A. M. (1987). *Robust Regression and Outlier Detection.* New York, NY: Wiley.

Rousseeuw, P. J., and Van Zomeren, B. C., (1990). Unmasking multivariate outliers and leverage points. *Journal of American Statistical Association* 85, 633-639.

Scholz, F.W. (1978). Weighted median regression estimates. *The Annals of Statistics,* 6 (3), 603-609.

Sprent, P. (1993). *Applied Nonparametric Statistical Methods.* London; New York: Chapman and Hall.

Talwar, P. (1993). A simulation study of some nonparametric regression estimators. *Computational Statistics and Data Analysis,* 15 (3), 309-327.

Theil, H. (1950). A rank-invariant method of linear and polynomial regression analysis. *Indagationes Mathematicae,* 12, 85-91.

Tiku, M. L. (1967). Estimating the mean and the standard deviation from a censored normal sample. *Biometrika,* 54, 155-165.

Tiku, M. L. (1968). Estimating the parameters of normal and logistic distributions from censored samples. *Austral. J. Statist.,* 10, 64-74.

Tiku, M. L. (1980). Robustness of MML estimators based on censored samples and robust test statistics. *J. Stat. Plann. Inf.,* 4,123-143.

Tiku, M. L., and Suresh, R. P. (1992). A new method of estimation for location and scale parameters. *J. Stat. Plann. Inf.,* 30, 281-292.

Tiku, M. L., Islam, M. Q., and Selcuk, A. S. (2001). Non-normal regression II. Symmetric distributions. *Commun. Stat.-Theory Meth.,* 30 (6), 1021-1045.

Tiku, M. L., Tan, W. Y., and Balakrishnan, N. (1986). *Robust Inference,* Marcel Dekker, New York.

Vaughan, D. C. (1992). On the Tiku-Suresh method of estimation. *Commun. Stat.-Theory Meth.* 21 (2), 451-469.

Vaughan, D. C., and Tiku, M. L. (2000). Estimation and hypothesis testing for a non-normal bivariate distribution with applications. *J. Mathematical and Computer Modelling,* 32, 53-67.

Wilcox, R. (1998). Simulation results on extensions of the Theil-Sen regression estimator. *Communications in Statistics- Simulation and Computation,* 27 (4), 1117-1126.

Yale, C., and Forsythe, A. B. (1976). Winsorized regression. *Technometrics,* 18, 291-300.

Zaman, A., Rousseeuw, P. J., and Orhan M. (2001). Econometric applications of high breakdown robust regression techniques. *Economics Letters,* 71, 1-8.

APPENDIX

Visual Fortran Program for the Comparison of Robust and Nonparametric Estimators Under the Simple Linear Regression Model

```
******************************************************************
Program Written by  Oya CAN MUTAN, 2004, Ankara
******************************************************************

use numerical_libraries

real x(100),y(100),e(100),xsum,xbar,sum1,sum2,LSEteta1
real sumkareLSE1,LSE1(30000),sumLSE1,simLSE1bar,varpayLSE1,varLSE1
real LSE1kare(30000),ysum,ybar,LSEteta0,sumkareLSE0,LSE0(30000)
real simLSE0bar,varpayLSE0,LSE0kare(30000),sumLSE0,varLSE0
real biasLSE0,biasLSE1,mseLSE0,mseLSE1,nvarLSE0,nbiasLSE0,nmseLSE0
real nvarLSE1,nbiasLSE1,nmseLSE1, u1,ulg(100),uexp(100),expo

real un(100),b,teta(3),w(2,100),conco1,conco2,conco3,t(100)
real q(100),alfapay(100),alfapayda(100),alfa(100),bbeta(100),mm
real delta,sumy,cybar,sumx,cxbar,Kpay,Kpayda,mK,Dpay,D,sumBB,BB
real sumC,C,sigmapay(2),sigmapayda(2),sigma(2),tetazero(2)
real delta1(100),sumMML0,sumkareMML0,sumMML1,sumkareMML1
real MML0(30000),MML0kare(30000),MML1(30000),MML1kare(30000)
real simMML0bar,biasMML0,varpayMML0,varMML0,mseMML0,simMML1bar
real biasMML1,varpayMML1,varMML1,mseMML1,rmseMML0,rmseMML1
real nvarMML0,nbiasMML0,nmseMML0,nvarMML1,nbiasMML1,nmseMML1

real reskare(100),concores,concox,concoy,LTSxsum,LTSxbar
real LTSsum1,LTSsum2,LTSysum,LTSybar,LTSteta1,LTSteta0,sumLTS0
real sumkareLTS0,sumLTS1,sumkareLTS1,LTS0(30000),LTS0kare(30000)
real LTS1(30000),LTS1kare(30000),simLTS0bar,varpayLTS0,varLTS0
real simLTS1bar,varpayLTS1,varLTS1,biasLTS0,biasLTS1,mseLTS0
real mseLTS1,rmseLTS0,rmseLTS1,nbiasLTS0,nvarLTS0,nmseLTS0
real nbiasLTS1,nvarLTS1,nmseLTS1

real reswin(6,100),concoreswin,concoxwin,concoywin
real WINteta0(6),WINteta1(6),WINxsum,WINysum,WINxbar,WINybar
real WINsum1,WINsum2,sumWIN0,sumkareWIN0,sumWIN1,sumkareWIN1
real WIN0(30000),WIN0kare(30000),WIN1(30000),WIN1kare(30000)
real simWIN0bar,varpayWIN0,varWIN0,simWIN1bar,varpayWIN1,varWIN1
real biasWIN0,biasWIN1,mseWIN0,mseWIN1,rmseWIN0,rmseWIN1
real nbiasWIN0,nvarWIN0,nmseWIN0,nbiasWIN1,nvarWIN1,nmseWIN1

real reswin2(6,100),concoreswin2,concoxwin2,concoywin2
real WIN2teta0(6),WIN2teta1(6),WIN2xsum,WIN2ysum,WIN2xbar,WIN2ybar
real WIN2sum1,WIN2sum2,sumWIN20,sumkareWIN20,sumWIN21,sumkareWIN21
```

```
      real WIN20(30000),WIN20kare(30000),WIN21(30000),WIN21kare(30000)
      real simWIN20bar,varpayWIN20,varWIN20,simWIN21bar,varpayWIN21
      real varWIN21,biasWIN20,biasWIN21,mseWIN20,mseWIN21,rmseWIN20
      real rmseWIN21,nbiasWIN20,nvarWIN20,nmseWIN20,nbiasWIN21
      real nvarWIN21,nmseWIN21

      real concotb,tb(9900),ta(100),concota,THLteta1,THLteta0,sumTHL0
      real sumkareTHL0,sumTHL1,sumkareTHL1,THL0(30000),THL0kare(30000)
      real THL1(30000),THL1kare(30000),simTHL0bar,varpayTHL0,varTHL0
      real simTHL1bar,varpayTHL1,varTHL1,biasTHL0,biasTHL1,mseTHL0
      real mseTHL1,rmseTHL0,rmseTHL1,nbiasTHL0,nvarTHL0,nmseTHL0
      real nbiasTHL1,nvarTHL1,nmseTHL1

      real concowtb,wtb(9900),wta(100),concowta,wTHLteta1,wTHLteta0
      real bw1(100,100),unorb(4950),tw(100,100),unort(4950),concotw
      real sumwTHL0,sumkarewTHL0,sumwTHL1,sumkarewTHL1,wTHL0(30000)
      real wTHL0kare(30000),wTHL1(30000),wTHL1kare(30000),simwTHL0bar
      real varpaywTHL0,varwTHL0,simwTHL1bar,varpaywTHL1,varwTHL1
      real biaswTHL0,biaswTHL1,msewTHL0,msewTHL1,rmsewTHL0,rmsewTHL1
      real sumw,sumwtb,nbiaswTHL0,nvarwTHL0,nmsewTHL0,nbiaswTHL1
      real nvarwTHL1,nmsewTHL1,sumtw

      real sl(100),sx(100),concos,concosx,sumsx,sumT,slope(100)
      real intercept(100),LADteta0,LADteta1,sumLAD0,sumkareLAD0,sumLAD1
      real sumkareLAD1,LAD0(30000),LAD0kare(30000),LAD1(30000)
      real LAD1kare(30000),simLAD0bar,biasLAD0,varpayLAD0,varLAD0
      real mseLAD0,simLAD1bar,biasLAD1,varpayLAD1,varLAD1,mseLAD1
      real rmseLAD0,rmseLAD1,yzero,xzero,concoladx,concolady,nbiasLAD0
      real nvarLAD0,nmseLAD0,nbiasLAD1,nvarLAD1,nmseLAD1

      integer n,nn,scount,res,res2,l,l2,theil,kk,jj,ordertb,orderta,k
      integer orderwtb,orderLTS,orderWIN,orderWIN2,orderwta,m,wk,wj,wt
      integer orderaij,orderzy,orderzx,twk,twj,twt,orderb,tk,orders
      integer resul,r

c-----------------------------------------------------------------------
      specifying the path of the output file
c-----------------------------------------------------------------------

      open(unit=1,file='c:\oyaprogram\tezgenlog\tezGENLOG.txt')

      print*,'enter nn,n,b'
      read*,nn,n,b

c-----------------------------------------------------------------------
      generating design points x(i)=i
c-----------------------------------------------------------------------

      do m=1,n
      x(m)=m*1.0
      enddo

c-----------------------------------------------------------------------
      starting simulation
c-----------------------------------------------------------------------

      do 1000 s=1,nn
      scount=s

c-----------------------------------------------------------------------
      generating y(i) from generalised logistic
c-----------------------------------------------------------------------

      call rnun(n,un)
      do i=1,n
```

```
      e(i)=-alog((un(i)**(-1.0/b))-1.0)
      y(i)=x(i)+e(i)
      enddo

c-----------------------------------------------------------------------
 model 1: GL(b=2.0,sigma=1.0), assumed model GL(b=4.0,sigma=1.0)
 model 2: GL(b=6.0,sigma=1.0), assumed model GL(b=4.0,sigma=1.0)
c-----------------------------------------------------------------------

      b=4.0

c-----------------------------------------------------------------------
  model 3: outlier model,(n-r)*GL(b=4.0,sigma=1.0)+r*GL(b=4.0,sigma=2.0)
c-----------------------------------------------------------------------

      r=int(0.5+0.1*n)
      do i=1,r
      e(i)=4.0*e(i)
      y(i)=x(i)+e(i)
      enddo

c-----------------------------------------------------------------------
  model 4: mixture model,0.90*GL(b=4.0,sigma=1.0)+0.10*GL(b=4.0,sigma=2.0)
c-----------------------------------------------------------------------

      call rnun(n,ulg)
      do i=1,n
      if(ulg(i).gt.0.90) then
      e(i)=4.0*e(i)
      y(i)=x(i)+e(i)
      endif
      enddo

c-----------------------------------------------------------------------
  model 5: contamination model, 0.90*GL(b=4.0,sigma=1.0)+0.10*EXP(1)
c-----------------------------------------------------------------------

      call rnun(n,uexp)
      do i=1,n
      if (uexp(i).gt.0.90) then
      call rnexp(1,expo)
      e(i)=expo
      y(i)=x(i)+e(i)
      endif
      enddo

c-----------------------------------------------------------------------
   calculating Theil estimator of teta1, THLteta1=med(b)  (i<j)
c-----------------------------------------------------------------------

      twt=0
      twk=1

      do 110 i=1,n

      twj=1

111   if(i.ge.twj) then
      twj=twj+1
      go to 111
      endif

      if(twj.gt.n) then
      twt=1
      endif
```

```
      if(twt.eq.1) then
      go to 62
      endif
      tb(twk)=(y(twj)-y(i))/(x(twj)-x(i))

      twk=twk+1
      twj=twj+1

      if(twj.le.n) then
      go to 111
      endif

110   continue

62    ordertb=1

63    if(ordertb.eq.1) then

      ordertb=0

      do 64 i=1,n*(n-1)/2-1

      if(tb(i).gt.tb(i+1)) then

      concotb=tb(i)
      tb(i)=tb(i+1)
      tb(i+1)=concotb

      ordertb=1

      endif

64    continue

      go to 63

      endif

      THLteta1=tb((n*(n-1)/2+1)/2)

c-----------------------------------------------------------------------
   calculating weighted Theil estimator of teta1
c-----------------------------------------------------------------------

      tk=0

      sumw=0.0

      do i=1,n

      do j=1,n
      if (i.lt.j) then
      sumw=sumw+abs(x(j)-x(i))
      endif
      enddo

      enddo

      do i=1,n

      do j=1,n
      if (i.lt.j) then
      tk=tk+1
      tw(i,j)=abs(x(j)-x(i))/sumw
      unort(tk)=tw(i,j)
```

```
      bw1(i,j)=(y(j)-y(i))/(x(j)-x(i))
      unorb(tk)=bw1(i,j)
      endif
      enddo
      enddo


      orderb=1
133   if(orderb.eq.1) then
      orderb=0

      do 134 i=1,tk-1

      if(unorb(i).gt.unorb(i+1)) then

      concotb=unorb(i)
      unorb(i)=unorb(i+1)
      unorb(i+1)=concotb

      concotw=unort(i)
      unort(i)=unort(i+1)
      unort(i+1)=concotw

      orderb=1

      endif

134   continue

      go to 133

      endif


      sumtw=0.0

      do i=1,tk

      sumtw=sumtw+unort(i)

      if (sumtw.gt.0.5) then
      wTHLteta1=unorb(i)
      go to 65

      else if(sumtw.eq.0.5) then
      wTHLteta1=(unorb(i-1)+unorb(i))/2.0
      go to 65

      endif

      enddo

c-----------------------------------------------------------------------
      calculating y-intercepts for Theil & weighted Theil method
c-----------------------------------------------------------------------

65    do i=1,n
      ta(i)=y(i)-THLteta1*x(i)
      wta(i)=y(i)-wTHLteta1*x(i)
      enddo

      orderta=1
66    if(orderta.eq.1) then
      orderta=0
```

```
      do 67 i=1,n-1

      if(ta(i).gt.ta(i+1)) then

      concota=ta(i)
      ta(i)=ta(i+1)
      ta(i+1)=concota

      orderta=1

      endif

67    continue

      go to 66

      endif


      THLteta0=(ta(n/2)+ta(n/2+1))/2.0


      orderwta=1
70    if(orderwta.eq.1) then
      orderwta=0

      do 71 i=1,n-1

      if(wta(i).gt.wta(i+1)) then

      concowta=wta(i)
      wta(i)=wta(i+1)
      wta(i+1)=concowta

      orderwta=1

      endif

71    continue

      go to 70

      endif

      wTHLteta0=(wta(n/2)+wta(n/2+1))/2.0

c-------------------------------------------------------------------------
      calculating LAD estimators
c-------------------------------------------------------------------------

      j=1
      k=1

1     yzero=y(k)
      xzero=x(k)

      sumT=0.0

      do i=1,n

      sx(i)=abs(x(i)-xzero)
      sumT=sumT+sx(i)
      sl(i)=(y(i)-yzero)/(x(i)-xzero)

      if(i.eq.k) then
```

```
      sl(i)=10000
      endif

      enddo
      orders=1

135   if(orders.eq.1) then

      orders=0

      do 136 i=1,n-1

      if(sl(i).gt.sl(i+1)) then

      concos=sl(i)
      sl(i)=sl(i+1)
      sl(i+1)=concos

      concosx=sx(i)
      sx(i)=sx(i+1)
      sx(i+1)=concosx

      concoladx=x(i)
      x(i)=x(i+1)
      x(i+1)=concoladx

      conco lady=y(i)
      y(i)=y(i+1)
      y(i+1)=concolady

      orders=1

      endif

136   continue

      go to 135

      endif

      sumsx=0.0

      do i=1,n

      sumsx=sumsx+sx(i)

      if (sumsx.gt.(sumT/2.0)) then

      slope(j)=sl(i)
      intercept(j)=yzero-slope(j)*xzero

      k=i

      if(j.ge.2) then

      if (slope(j).eq.slope(j-1)) then

      LADteta1=slope(j)
      LADteta0=intercept(j)

      go to 10
      endif

      endif
```

```
      j=j+1
      if (j.le.n) then
      go to 1
      endif
      endif

      enddo

c------------------------------------------------------------------------
      calculating LS estimators
c------------------------------------------------------------------------

10    xsum=0.0
      ysum=0.0

      do 2 i=1,n
      xsum=xsum+x(i)
      ysum=ysum+y(i)
2     continue

      xbar=xsum/(1.0*n)
      ybar=ysum/(1.0*n)

c------------------------------------------------------------------------
      calculating LSE of teta1
c------------------------------------------------------------------------

      sum1=0.0

      do i=1,n
      sum1=sum1+(x(i)-xbar)*y(i)
      enddo

      sum2=0.0

      do i=1,n
      sum2=sum2+(x(i)-xbar)**2.0
      enddo

      LSEteta1=sum1/sum2

c------------------------------------------------------------------------
      calculating LSE of teta0
c------------------------------------------------------------------------

      LSEteta0=ybar-LSEteta1*xbar

c------------------------------------------------------------------------
      calculating LTS estimators (totally 20% )
c------------------------------------------------------------------------

      do i=1,n
      reskare(i)=(y(i)-LSEteta0-LSEteta1*x(i))**2.0
      enddo

c------------------------------------------------------------------------
      ordering residuals and calculating concominants
c------------------------------------------------------------------------

      orderLTS=1
50    if (orderLTS.eq.1) then
      orderLTS=0

      do 31 i=1,n-1
```

```fortran
      if (reskare(i).gt.reskare(i+1)) then
      concores=reskare(i)
      reskare(i)=reskare(i+1)
      reskare(i+1)=concores

      concox=x(i)
      x(i)=x(i+1)
      x(i+1)=concox

      concoy=y(i)
      y(i)=y(i+1)
      y(i+1)=concoy

      orderLTS=1

      endif

31    continue

      go to 50

      endif

      LTSxsum=0.0
      LTSysum=0.0

      do i=1,(n-nint(0.2*n))

      LTSxsum=LTSxsum+x(i)
      LTSysum=LTSysum+y(i)

      enddo

      LTSxbar=LTSxsum/(1.0*n-nint(0.2*n))
      LTSybar=LTSysum/(1.0*n-nint(0.2*n))

c----------------------------------------------------------------------
      calculating LTS estimator of teta1
c----------------------------------------------------------------------

      LTSsum1=0.0

      do i=1,(n-nint(0.2*n))
      LTSsum1=LTSsum1+(x(i)-LTSxbar)*y(i)
      enddo

      LTSsum2=0.0

      do i=1,(n-nint(0.2*n))
      LTSsum2=LTSsum2+(x(i)-LTSxbar)**2.0
      enddo

      LTSteta1=LTSsum1/LTSsum2

c----------------------------------------------------------------------
      calculating LTS estimator of teta0
c----------------------------------------------------------------------

      LTSteta0=LTSybar-LTSteta1*LTSxbar

c----------------------------------------------------------------------
      calculating MMLE's
c----------------------------------------------------------------------
```

```
c------------------------------------------------------------------------
      ordering w(i)'s and calculating concomitants
c------------------------------------------------------------------------

      teta(1)=LSEteta1

      do 6 j=1,2

      do i=1,n
      w(j,i)=y(i)-teta(j)*x(i)
      enddo

      resul=1
5     if (resul.eq.1.0) then
      resul=0

      do 8 i=1,n-1

      if (w(j,i).gt.w(j,i+1)) then

      conco1=w(j,i)
      w(j,i)=w(j,i+1)
      w(j,i+1)=conco1

      conco2=x(i)
      x(i)=x(i+1)
      x(i+1)=conco2

      conco3=y(i)
      y(i)=y(i+1)
      y(i+1)=conco3

      resul=1.0

      endif

8     continue

      go to 5

      endif

c------------------------------------------------------------------------
      calculating q(i),t(i),alfa(i),beta(i),m,delta(i)
c------------------------------------------------------------------------

      do i=1,n
      q(i)=(1.0*i)/(1.0*n+1.0)
      enddo

      do i=1,n
      t(i)=-alog((q(i)**(-1.0/b))-1.0)
      enddo

      do i=1,n
      alfapay(i)=1.0+exp(t(i))+t(i)*exp(t(i))
      enddo

      do i=1,n
      alfapayda(i)=(1.0+exp(t(i)))**2.0
      enddo

      do i=1,n
      alfa(i)=alfapay(i)/alfapayda(i)
      enddo
```

```
      do i=1,n
      bbeta(i)=exp(t(i))/alfapayda(i)
      enddo

      mm=0.0

      do i=1,n
      mm=mm+bbeta(i)
      enddo

      do i=1,n
      delta1(i)=alfa(i)-(b+1.0)**(-1.0)
      enddo

      delta=0.0

      do i=1,n
      delta=delta+delta1(i)
      enddo

c-----------------------------------------------------------------------
      calculating ybar[.],xbar[.],K,D,B,C
c-----------------------------------------------------------------------

      sumy=0.0

      do i=1,n
      sumy=sumy+bbeta(i)*y(i)
      enddo

      cybar=sumy/mm

      sumx=0.0

      do i=1,n
      sumx=sumx+bbeta(i)*x(i)
      enddo

      cxbar=sumx/mm

      Kpay=0.0

      do i=1,n
      Kpay=Kpay+bbeta(i)*(x(i)-cxbar)*y(i)
      enddo

      Kpayda=0.0

      do i=1,n
      Kpayda=Kpayda+bbeta(i)*(x(i)-cxbar)*(x(i)-cxbar)
      enddo

      mK=Kpay/Kpayda

      Dpay=0.0

      do i=1,n
      Dpay=Dpay+delta1(i)*(x(i)-cxbar)
      enddo

      D=Dpay/Kpayda

      sumBB=0.0

      do i=1,n
```

```
      sumBB=sumBB+delta1(i)*(y(i)-cybar-mK*(x(i)-cxbar))
      enddo

      BB=(b+1.0)*sumBB
      sumC=0.0

      do i=1,n
      sumC=sumC+bbeta(i)*((y(i)-cybar-mK*(x(i)-cxbar))**2.0)
      enddo

      C=(b+1.0)*sumC

      sigmapay(j)=-BB+sqrt(BB*BB+4.0*n*C)
      sigmapayda(j)=2.0*sqrt(1.0*n*(1.0*n-2.0))

      sigma(j)=sigmapay(j)/sigmapayda(j)

      teta(j+1)=mK-D*sigma(j)

      tetazero(j)=cybar-teta(j+1)*cxbar-(delta/mm)*sigma(j)

6     continue

c-------------------------------------------------------------------------
     calculating 10% winsorized estimators
c-------------------------------------------------------------------------

      WINteta0(1)=LSEteta0
      WINteta1(1)=LSEteta1

      do 39 j=1,5

      do i=1,n
      reswin(j,i)=abs(y(i)-WINteta0(j)-WINteta1(j)*x(i))
      enddo

c-------------------------------------------------------------------------
      ordering residuals and calculating concominants
c-------------------------------------------------------------------------

      orderWIN=1

51    if (orderWIN.eq.1) then

      orderWIN=0

      do 41 i=1,n-1

      if (reswin(j,i).gt.reswin(j,i+1)) then

      concoreswin=reswin(j,i)
      reswin(j,i)=reswin(j,i+1)
      reswin(j,i+1)=concoreswin

      concoxwin=x(i)
      x(i)=x(i+1)
      x(i+1)=concoxwin

      concoywin=y(i)
      y(i)=y(i+1)
      y(i+1)=concoywin

      orderWIN=1
      endif
```

```
41     continue

       go to 51

       endif

       do 42 k=1,nint(0.1*n)

       res=1

       l=1

52     if(res.eq.1) then

       if (((y(n-k+1)-WINteta0(j)-WINteta1(j)*x(n-k+1))*(y(n-k+1-l)-
      &WINteta0(j)-WINteta1(j)*x(n-k+1-l))).gt.0.0) then

       y(n-k+1)=WINteta0(j)+WINteta1(j)*x(n-k+1)+y(n-k+1-l)-WINteta0(j)-
     & WINteta1(j)*x(n-k+1-l)

       res=0

       endif

       l=l+1

       if (l.le.(n-1)) then
       go to 52
       endif

       endif

       l=l+1

       if (l.le.(n-1)) then
       go to 52
       endif

42     continue

       WINxsum=0.0
       WINysum=0.0

       do i=1,n
       WINxsum=WINxsum+x(i)
       WINysum=WINysum+y(i)
       enddo

       WINxbar=WINxsum/(1.0*n)
       WINybar=WINysum/(1.0*n)

c-------------------------------------------------------------------------
       calculating WIN estimator of teta1
c-------------------------------------------------------------------------

       WINsum1=0.0
       WINsum2=0.0

       do i=1,n
       WINsum1=WINsum1+(x(i)-WINxbar)*y(i)
       enddo

       do i=1,n
       WINsum2=WINsum2+(x(i)-WINxbar)**2.0
       enddo
```

```
      WINteta1(j+1)=WINsum1/WINsum2

c-----------------------------------------------------------------------
      calculating WIN estimator of teta0
c-----------------------------------------------------------------------

      WINteta0(j+1)=WINybar-WINteta1(j+1)*WINxbar

39    continue

c-----------------------------------------------------------------------
     calculating 20% winsorized estimators
c-----------------------------------------------------------------------

      WIN2teta0(1)=LSEteta0
      WIN2teta1(1)=LSEteta1

      do 75 j=1,5

      do i=1,n
      reswin2(j,i)=abs(y(i)-WIN2teta0(j)-WIN2teta1(j)*x(i))
      enddo

c-----------------------------------------------------------------------
      ordering residuals and calculating concominants
c-----------------------------------------------------------------------

      orderWIN2=1

77    if (orderWIN2.eq.1) then

      orderWIN2=0

      do 78 i=1,n-1

      if (reswin2(j,i).gt.reswin2(j,i+1)) then

      concoreswin2=reswin2(j,i)
      reswin2(j,i)=reswin2(j,i+1)
      reswin2(j,i+1)=concoreswin2

      concoxwin2=x(i)
      x(i)=x(i+1)
      x(i+1)=concoxwin2

      concoywin2=y(i)
      y(i)=y(i+1)
      y(i+1)=concoywin2

      orderWIN2=1

      endif

78    continue

      go to 77

      endif

      do 79 k=1,nint(0.2*n)

      res2=1
      l2=1
```

```
80    if(res2.eq.1) then

      if (((y(n-k+1)-WIN2teta0(j)-WIN2teta1(j)*x(n-k+1))*(y(n-k+1-l2)-
     &WIN2teta0(j)-WIN2teta1(j)*x(n-k+1-l2))).gt.0.0) then

      y(n-k+1)=WIN2teta0(j)+WIN2teta1(j)*x(n-k+1)+y(n-k+1-l2)-
     &WIN2teta0(j)-WIN2teta1(j)*x(n-k+1-l2)

      res2=0

      endif

      l2=l2+1

      if (l2.le.(n-1)) then
      go to 80
      endif

      endif

      l2=l2+1

      if (l2.le.(n-1)) then
      go to 80
      endif

79    continue

      WIN2xsum=0.0
      WIN2ysum=0.0

      do i=1,n
      WIN2xsum=WIN2xsum+x(i)
      WIN2ysum=WIN2ysum+y(i)
      enddo

      WIN2xbar=WIN2xsum/(1.0*n)
      WIN2ybar=WIN2ysum/(1.0*n)

c-------------------------------------------------------------------------
      calculating WIN estimator of teta1
c-------------------------------------------------------------------------

      WIN2sum1=0.0
      WIN2sum2=0.0

      do i=1,n
      WIN2sum1=WIN2sum1+(x(i)-WIN2xbar)*y(i)
      enddo

      do i=1,n
      WIN2sum2=WIN2sum2+(x(i)-WIN2xbar)**2.0
      enddo

      WIN2teta1(j+1)=WIN2sum1/WIN2sum2

c-------------------------------------------------------------------------
      calculating WIN estimator of teta0
c-------------------------------------------------------------------------

      WIN2teta0(j+1)=WIN2ybar-WIN2teta1(j+1)*WIN2xbar

75    continue
      LSE0(scount)=LSEteta0
      LSE1(scount)=LSEteta1
```

```
      MML0(scount)=tetazero(2)
      MML1(scount)=teta(3)

      LTS0(scount)=LTSteta0
      LTS1(scount)=LTSteta1

      WIN0(scount)=WINteta0(6)
      WIN1(scount)=WINteta1(6)

      WIN20(scount)=WIN2teta0(6)
      WIN21(scount)=WIN2teta1(6)

      THL0(scount)=THLteta0
      THL1(scount)=THLteta1

      wTHL0(scount)=wTHLteta0
      wTHL1(scount)=wTHLteta1

      LAD0(scount)=LADteta0
      LAD1(scount)=LADteta1

1000  continue

c------------------------------------------------------------------------
      calculating simulated means & bias
c------------------------------------------------------------------------

      sumLSE0=0.0
      sumLSE1=0.0

      sumMML0=0.0
      sumMML1=0.0

      sumLTS0=0.0
      sumLTS1=0.0

      sumWIN0=0.0
      sumWIN1=0.0

      sumWIN20=0.0
      sumWIN21=0.0

      sumTHL0=0.0
      sumTHL1=0.0

      sumwTHL0=0.0
      sumwTHL1=0.0

      sumLAD0=0.0
      sumLAD1=0.0

      do i=1,nn

      sumLSE0=sumLSE0+LSE0(i)
      sumLSE1=sumLSE1+LSE1(i)

      sumMML0=sumMML0+MML0(i)
      sumMML1=sumMML1+MML1(i)

      sumLTS0=sumLTS0+LTS0(i)
      sumLTS1=sumLTS1+LTS1(i)

      sumWIN0=sumWIN0+WIN0(i)
      sumWIN1=sumWIN1+WIN1(i)
```

```
      sumWIN20=sumWIN20+WIN20(i)
      sumWIN21=sumWIN21+WIN21(i)

      sumTHL0=sumTHL0+THL0(i)
      sumTHL1=sumTHL1+THL1(i)

      sumwTHL0=sumwTHL0+wTHL0(i)
      sumwTHL1=sumwTHL1+wTHL1(i)

      sumLAD0=sumLAD0+LAD0(i)
      sumLAD1=sumLAD1+LAD1(i)

      enddo

      simLSE0bar=sumLSE0/(1.0*nn)
      biasLSE0=simLSE0bar-0.0
      simLSE1bar=sumLSE1/(1.0*nn)
      biasLSE1=simLSE1bar-1.0

      simMML0bar=sumMML0/(1.0*nn)
      biasMML0=simMML0bar-0.0
      simMML1bar=sumMML1/(1.0*nn)
      biasMML1=simMML1bar-1.0

      simLTS0bar=sumLTS0/(1.0*nn)
      biasLTS0=simLTS0bar-0.0
      simLTS1bar=sumLTS1/(1.0*nn)
      biasLTS1=simLTS1bar-1.0

      simWIN0bar=sumWIN0/(1.0*nn)
      biasWIN0=simWIN0bar-0.0
      simWIN1bar=sumWIN1/(1.0*nn)
      biasWIN1=simWIN1bar-1.0

      simWIN20bar=sumWIN20/(1.0*nn)
      biasWIN20=simWIN20bar-0.0
      simWIN21bar=sumWIN21/(1.0*nn)
      biasWIN21=simWIN21bar-1.0

      simTHL0bar=sumTHL0/(1.0*nn)
      biasTHL0=simTHL0bar-0.0
      simTHL1bar=sumTHL1/(1.0*nn)
      biasTHL1=simTHL1bar-1.0

      simwTHL0bar=sumwTHL0/(1.0*nn)
      biaswTHL0=simwTHL0bar-0.0
      simwTHL1bar=sumwTHL1/(1.0*nn)
      biaswTHL1=simwTHL1bar-1.0

      simLAD0bar=sumLAD0/(1.0*nn)
      biasLAD0=simLAD0bar-0.0
      simLAD1bar=sumLAD1/(1.0*nn)
      biasLAD1=simLAD1bar-1.0

c-----------------------------------------------------------------------
      calculating simulated variances
c-----------------------------------------------------------------------

      varpayLSE0=0.0
      varpayLSE1=0.0

      varpayMML0=0.0
      varpayMML1=0.0
```

```
varpayLTS0=0.0
varpayLTS1=0.0

varpayWIN0=0.0
varpayWIN1=0.0

varpayWIN20=0.0
varpayWIN21=0.0

varpayTHL0=0.0
varpayTHL1=0.0

varpaywTHL0=0.0
varpaywTHL1=0.0

varpayLAD0=0.0
varpayLAD1=0.0

do i=1,nn

varpayLSE0=varpayLSE0+(LSE0(i)-simLSE0bar)**2.0
varpayLSE1=varpayLSE1+(LSE1(i)-simLSE1bar)**2.0

varpayMML0=varpayMML0+(MML0(i)-simMML0bar)**2.0
varpayMML1=varpayMML1+(MML1(i)-simMML1bar)**2.0

varpayLTS0=varpayLTS0+(LTS0(i)-simLTS0bar)**2.0
varpayLTS1=varpayLTS1+(LTS1(i)-simLTS1bar)**2.0

varpayWIN0=varpayWIN0+(WIN0(i)-simWIN0bar)**2.0
varpayWIN1=varpayWIN1+(WIN1(i)-simWIN1bar)**2.0

varpayWIN20=varpayWIN20+(WIN20(i)-simWIN20bar)**2.0
varpayWIN21=varpayWIN21+(WIN21(i)-simWIN21bar)**2.0

varpayTHL0=varpayTHL0+(THL0(i)-simTHL0bar)**2.0
varpayTHL1=varpayTHL1+(THL1(i)-simTHL1bar)**2.0

varpaywTHL0=varpaywTHL0+(wTHL0(i)-simwTHL0bar)**2.0
varpaywTHL1=varpaywTHL1+(wTHL1(i)-simwTHL1bar)**2.0

varpayLAD0=varpayLAD0+(LAD0(i)-simLAD0bar)**2.0
varpayLAD1=varpayLAD1+(LAD1(i)-simLAD1bar)**2.0

enddo

varLSE0=varpayLSE0/(1.0*nn-1.0)
varLSE1=varpayLSE1/(1.0*nn-1.0)

varMML0=varpayMML0/(1.0*nn-1.0)
varMML1=varpayMML1/(1.0*nn-1.0)

nvarMML0=1.0*n*varMML0
nvarMML1=1.0*n*varMML1

varLTS0=varpayLTS0/(1.0*nn-1.0)
varLTS1=varpayLTS1/(1.0*nn-1.0)

varWIN0=varpayWIN0/(1.0*nn-1.0)
varWIN1=varpayWIN1/(1.0*nn-1.0)

varWIN20=varpayWIN20/(1.0*nn-1.0)
varWIN21=varpayWIN21/(1.0*nn-1.0)

varTHL0=varpayTHL0/(1.0*nn-1.0)
```

```
      varTHL1=varpayTHL1/(1.0*nn-1.0)

      varwTHL0=varpaywTHL0/(1.0*nn-1.0)
      varwTHL1=varpaywTHL1/(1.0*nn-1.0)

      varLAD0=varpayLAD0/(1.0*nn-1.0)
      varLAD1=varpayLAD1/(1.0*nn-1.0)

c-----------------------------------------------------------------------
      calculating mean square error
c-----------------------------------------------------------------------

      mseLSE0=varLSE0+biasLSE0**2.0
      mseLSE1=varLSE1+biasLSE1**2.0

      mseMML0=varMML0+biasMML0**2.0
      mseMML1=varMML1+biasMML1**2.0

      mseLTS0=varLTS0+biasLTS0**2.0
      mseLTS1=varLTS1+biasLTS1**2.0

      mseWIN0=varWIN0+biasWIN0**2.0
      mseWIN1=varWIN1+biasWIN1**2.0

      mseWIN20=varWIN20+biasWIN20**2.0
      mseWIN21=varWIN21+biasWIN21**2.0

      mseTHL0=varTHL0+biasTHL0**2.0
      mseTHL1=varTHL1+biasTHL1**2.0

      msewTHL0=varwTHL0+biaswTHL0**2.0
      msewTHL1=varwTHL1+biaswTHL1**2.0

      mseLAD0=varLAD0+biasLAD0**2.0
      mseLAD1=varLAD1+biasLAD1**2.0

c-----------------------------------------------------------------------
     calculating relative mean square error rmse=(mseOLS-mseTETA)/mseOLS
c-----------------------------------------------------------------------

      rmseLSE0=(mseLSE0-mseLSE0)/mseLSE0
      rmseMML0=(mseLSE0-mseMML0)/mseLSE0
      rmseLTS0=(mseLSE0-mseLTS0)/mseLSE0
      rmseWIN0=(mseLSE0-mseWIN0)/mseLSE0
      rmseWIN20=(mseLSE0-mseWIN20)/mseLSE0
      rmseTHL0=(mseLSE0-mseTHL0)/mseLSE0
      rmsewTHL0=(mseLSE0-msewTHL0)/mseLSE0
      rmseLAD0=(mseLSE0-mseLAD0)/mseLSE0

      rmseLSE1=(mseLSE1-mseLSE1)/mseLSE1
      rmseMML1=(mseLSE1-mseMML1)/mseLSE1
      rmseLTS1=(mseLSE1-mseLTS1)/mseLSE1
      rmseWIN1=(mseLSE1-mseWIN1)/mseLSE1
      rmseWIN21=(mseLSE1-mseWIN21)/mseLSE1
      rmseTHL1=(mseLSE1-mseTHL1)/mseLSE1
      rmsewTHL1=(mseLSE1-msewTHL1)/mseLSE1
      rmseLAD1=(mseLSE1-mseLAD1)/mseLSE1

c-----------------------------------------------------------------------
      printing mean,variance,bias,mse & rmse of teta0
c-----------------------------------------------------------------------

508   format(a3,1x,a5,7x,a4,10x,a8,10x,a4,13x,a3,13x,a4,9x)
      write(1,508)'for','teta0','mean','variance','bias','MSE',
     &'RMSE'
```

```
510    format(a3,7x,5(f12.8,4x))
       write(1,510)'OLS',simLSE0bar,varLSE0,biasLSE0,mseLSE0,rmseLSE0

522    format(a3,7x,5(f12.8,4x))
       write(1,522)'MML',simMML0bar,varMML0,biasMML0,mseMML0,rmseMML0

520    format(a3,7x,5(f12.8,4x))
       write(1,520)'LAD',simLAD0bar,varLAD0,biasLAD0,mseLAD0,rmseLAD0

511    format(a5,5x,5(f12.8,4x))
       write(1,511)'WIN10',simWIN0bar,varWIN0,biasWIN0,mseWIN0,rmseWIN0

512    format(a5,5x,5(f12.8,4x))
       write(1,512)'WIN20',simWIN20bar,varWIN20,biasWIN20,mseWIN20,
      &rmseWIN20

513    format(a3,7x,5(f12.8,4x))
       write(1,513)'TLS',simLTS0bar,varLTS0,biasLTS0,mseLTS0,rmseLTS0

516    format(a5,5x,5(f12.8,4x))
       write(1,516)'Theil',simTHL0bar,varTHL0,biasTHL0,mseTHL0,
      &rmseTHL0

517    format(a4,a5,1x,5(f12.8,4x))
       write(1,517)'Wtd.','Theil',simwTHL0bar,varwTHL0,biaswTHL0,
      &msewTHL0,rmsewTHL0

c-----------------------------------------------------------------------
       printing mean,variance,bias,mse & rmse of teta1
c-----------------------------------------------------------------------

500    format(a3,1x,a5,7x,a4,10x,a8,10x,a4,13x,a3,13x,a4,9x)
       write(1,500)'for','teta1','mean','variance','bias','MSE',
     &'RMSE'

501   format(a3,7x,5(f12.8,4x))
      write(1,501)'OLS',simLSE1bar,varLSE1,biasLSE1,mseLSE1,rmseLSE1

523   format(a3,7x,5(f12.8,4x))
      write(1,523)'MML',simMML1bar,varMML1,biasMML1,mseMML1,rmseMML1

521   format(a3,7x,5(f12.8,4x))
      write(1,521)'LAD',simLAD1bar,varLAD1,biasLAD1,mseLAD1,rmseLAD1

502   format(a5,5x,5(f12.8,4x))
      write(1,502)'WIN10',simWIN1bar,varWIN1,biasWIN1,mseWIN1,rmseWIN1

503   format(a5,5x,5(f12.8,4x))
      write(1,503)'WIN20',simWIN21bar,varWIN21,biasWIN21,mseWIN21,
     &rmseWIN21

504   format(a3,7x,5(f12.8,4x))
      write(1,504)'TLS',simLTS1bar,varLTS1,biasLTS1,mseLTS1,rmseLTS1

505   format(a5,5x,5(f12.8,4x))
       write(1,505)'Theil',simTHL1bar,varTHL1,biasTHL1,mseTHL1,
     &rmseTHL1

506   format(a4,a5,1x,5(f12.8,4x))
      write(1,506)'Wtd.','Theil',simwTHL1bar,varwTHL1,biaswTHL1,
     &msewTHL1,rmsewTHL1

       stop
       end
```