

ROBUST LINEAR REGRESSION

by

XUE BAI

B.S., Mathematics and Applied Mathematics, China, 2010

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2012

Approved by:

Major Professor

Weixin Yao

Copyright

Xue Bai

2012

Abstract

In practice, when applying a statistical method it often occurs that some observations deviate from the usual model assumptions. Least-squares (LS) estimators are very sensitive to outliers. Even one single atypical value may have a large effect on the regression parameter estimates. The goal of robust regression is to develop methods that are resistant to the possibility that one or several unknown outliers may occur anywhere in the data. In this paper, we review various robust regression methods including: M-estimate, LMS estimate, LTS estimate, S-estimate, τ -estimate, MM-estimate, GM-estimate, and REWLS estimate. Finally, we compare these robust estimates based on their robustness and efficiency through a simulation study. A real data set application is also provided to compare the robust estimates with traditional least squares estimator.

Key words: Linear regression model; outliers; robust regression.

Table of Contents

Table of Contents	iv
List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Least Squares estimate	1
1.2 Robust Method	2
1.2.1 Background	2
1.2.2 Measuring Robustness	3
2 Review of Robust Regression Methods	5
2.1 M-estimate	5
2.2 Least Median of Squares estimate	9
2.3 Least Trimmed Squares estimate	9
2.4 S-estimate	11
2.5 τ -estimate	12
2.6 M M-estimate	12
2.7 GM-estimate	14
2.7.1 Mallows GM-estimate	14
2.7.2 Schweppe GM-estimate	15
2.7.3 One-step GM-estimate	15
2.8 Robust and Efficient Weighted Least Squares estimate	16
3 Comparing Various Estimators	18
3.1 Simulation Study	18
3.2 Example	21
A R Code	30

List of Figures

2.1	Figure (a) shows the objective, ψ , and weight functions for the least-squares(top) and LAD (bottom) estimators. Figure (b) shows the objective, ψ , and weight functions for the Huber (top) and bisquare (bottom) estimators. The tuning constants for these graphs are $k=1.345$ for the Huber estimator and $k=4.685$ for the bisquare. (One way to think about this scaling is that the standard deviation of the errors, σ , is taken as 1.)	8
2.2	Least squares estimate (solid line) and M-estimate with Huber function (dashed line) for a dataset contains 20 observations two of which are high leverage outliers.	10
3.1	Plot of MSE of slope estimates vs. different cases for LMS, LTS, S, MM, and REWLS, for model 1 when $n = 20$	20
3.2	Plot of MSE of intercept estimates vs. different cases for LMS, LTS, S, MM, and REWLS, for model 1 when $n = 20$	21
3.3	Fit lines for Cigarettes data	26

List of Tables

3.1	Breakdown Points and Asymptotic Efficiencies of Various Regression Estimators	19
3.2	MSE of Point Estimates for Model 1 with $n = 20$	22
3.3	MSE of Point Estimates for Model 1 with $n = 100$	23
3.4	MSE of Point Estimates for Model 2 with $n = 20$	24
3.5	MSE of Point Estimates for Model 2 with $n = 100$	25
3.6	Cigarettes data	27
3.7	Regression estimates for Cigarettes data	27

Acknowledgments

First and foremost, I would like to express my appreciation to my major professor, Dr. Weixin Yao, for all his encouragement, guidance and suggestions.

I would also like to thank Dr. Leigh Murray and Dr. Abigail Jager for their willingness to serve on my committee and for their valuable insight.

My gratefulness extends to everyone who supported me in any respect during the completion of the report.

Chapter 1

Introduction

1.1 Least Squares estimate

The most basic method to estimate the parameters of linear regression models is the *Least Squares method* (LS method). The idea of least-squares analysis was independently formulated by Gauss (1777-1855) and Legendre (1752-1833), though it was first proposed in 1805 by Legendre (for more details, see Stigler, 1981). Since it was the only method of estimation that could be effectively computed before the advent of electronic computers, the LS method found immediate and lasting success.

Suppose the linear regression model is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \quad (1.1)$$

where \mathbf{y} is a $n \times 1$ response variable vector, \mathbf{X} is a $n \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$ unknown parameter vector and \mathbf{e} is the random error vector. The LS estimate of $\boldsymbol{\beta}$ is the $\hat{\boldsymbol{\beta}}$ which can minimize

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

Let

$$Q(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

We can get

$$Q(\boldsymbol{\beta}) = \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}.$$

Differentiating with respect to $\boldsymbol{\beta}$ yields

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}. \quad (1.2)$$

The matrix equation (1.2) is called the 'normal equations'. If \mathbf{X} has *full rank* then the solution is unique and is given by

$$\hat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

If \mathbf{X} is not of full rank, then we have what is called *collinearity*. In this situation, we will have an infinite number of solutions, which all yield the same predicted values and hence the same residuals.

The least squares estimate is optimal among the class of linear unbiased estimates and it is also the most efficient unbiased estimate under the assumption that errors in regression models are normally distributed ($e_i \sim \text{IIDN}(0, \sigma^2)$). However, in reality, errors do not usually exactly follow a normal distribution. Outliers often exist in a data set or errors may follow another distribution (e.g. t-distribution which has a heavier tail than the normal distribution). Here outliers are observations which do not follow the pattern of the other observations. In either case, the LS estimator will diverge strongly from the real value, no matter how big the sample size is.

Since the LS estimate is so sensitive to outliers, the easy and popular way to improve the estimate is by modifying or deleting outliers and applying LS to the modified data. Classical methods for outlier detection are based on initial LS fit and using numerical or graphical procedures (or both) called *regression diagnostics* to detect influential observations. These include the familiar Q-Q plots of residuals, plots of residuals vs. fitted values, Cook's distance, leverages, "leave-one-out" approach and so on. Since all of these methods are based on the initial LS fit, the residuals, standard deviations, and leverages may be largely influenced by outliers. Therefore, the above methods can be fooled by the combined action of several outliers, an effect that is referred to as *masking*, so these methods may even fail to recognize a single outlier.

Because of the above mentioned problems with the LS estimate, we want to develop new procedures that give a good fit to the unusual data without being perturbed by a small proportion of outliers and that do not require deciding and removing outliers. This leads to *robust regression*.

1.2 Robust Method

1.2.1 Background

Robust linear regression is designed to circumvent some limitations of traditional parametric methods. Ordinary least squares played an important role in the estimation field since they have favorable properties if their underlying assumptions are true. Unfortunately, those assumptions are often not met in practice. When people realized this, they started to seek methods which could remedy this problem. The idea of robust methods developed at the beginning of nineteenth century, with the rapid development of electronic technology. Robust statistics have attracted increasing attention from the 1960s. Now this is a very popular research area in statistics and a large number of articles have been published. There is still considerable work that needs to be done in this research area. This paper will review some popular robust regression methods and discuss their properties.

Robust regression tries to seek a model which represents the information in the majority of the data. Usually we use the properties of efficiency, the breakdown point (Donoho, 1983), and the influence function (Hampel, 1974) to measure the performance of robust techniques. The breakdown point and influence function will be described in the next section. The efficiency tells us how well the robust method works compared to LSE when data exactly follow a normal distribution. Because LSE is the best estimation method when data are normal, we want the robust estimator to perform as closely to LSE as possible. Thus, high efficiency is desired for robust estimation.

Historically, robust regression techniques mainly deal with three classes of problems:

- 1) Outliers only in the response direction (*y* – direction)
- 2) High leverage outliers (outliers in both the *x* – space and *y* – direction)
- 3) Distribution with heavier tail than normal distribution

Many methods have been developed for these problems. In the last chapter of this paper, we will discuss the properties of some robust methods based on these three kinds of problems by simulation studies.

1.2.2 Measuring Robustness

The goal of robust methods is to develop estimates which have ‘good’ behavior in an approximately normal model. The most common method to measure robustness is the *breakdown point*. The breakdown point (BP) is the largest proportion of the contaminations that the data can contain before the estimate fails. Thus the higher the BP of an estimator, the more robust it is. A common practical definition is the BP for a finite sample (FBP). The finite sample version of this concept is given by Donoho and Huber (1983) and is defined as the following.

Let $\mathbf{Z} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \dots, n\}$ and the corresponding $\hat{\boldsymbol{\beta}} = \mathbf{T}(\mathbf{Z})$. Then the breakdown-point of \mathbf{T} at \mathbf{Z} is defined by

$$\varepsilon^*(\mathbf{T}, \mathbf{Z}) = \min\{m/n : \sup_{\mathbf{Z}^*} \|\mathbf{T}(\mathbf{Z}^*) - \mathbf{T}(\mathbf{Z})\| = \infty\},$$

where $\|\cdot\|$ denotes the Euclidean norm and the supremum is taken over all choices of \mathbf{Z}^* consisting of $(n - m)$ points from \mathbf{Z} and m arbitrary points.

It is obvious that the BP of a sample mean and LSE is $\frac{1}{n}$, since even one outlier will lead to a big change in the estimation. For a similar reason, the BP of the sample median is $\frac{1}{2}$, which is the highest value of BP. Intuitively, a BP cannot exceed $\frac{1}{2}$ because if more than half of the data are outliers, it is impossible to distinguish between the “good” and “bad” distributions. Therefore, the maximum BP is 0.5 which is the goal of robust estimation. Although the sample median can achieve the best BP value, its efficiency is very low.

Another popular measurement of robustness is the *influence function* (Hampel, 1974). Let $\hat{\beta}$ be the estimate of β based on the original data and $\hat{\beta}_0$ be the estimate based on the data which has removed all outliers. Then we call $\hat{\beta} - \hat{\beta}_0$ the *sensitivity curve* of $\hat{\beta}$. The influence function (IF) is an asymptotic version of the sensitivity curve. When the sample contains a small fraction ε of identical outliers, it is defined as

$$IF_{\hat{\beta}}(x_0, F) = \lim_{\varepsilon \rightarrow 0^+} \frac{\hat{\beta}_{\infty}((1 - \varepsilon)F + \varepsilon\delta_{x_0}) - \hat{\beta}_{\infty}(F)}{\varepsilon},$$

where x_0 is the outlier, δ_{x_0} is a point mass at x_0 , and $\hat{\beta}_{\infty}(F)$ is the *asymptotic value* of the estimate at F .

The IF is a measure of the rate at which β responds to a small amount of contamination at x_0 . Loosely speaking, it tells us how much a single outlier affects the estimate. For a robust estimator, we want to ensure the IF does not go to infinity as x becomes arbitrarily large. Therefore, a bounded influence function is desired.

Chapter 2

Review of Robust Regression

Methods

2.1 M-estimate

The most common general robust method is M-estimates, introduced by Huber (1973). The M in M-estimates stands for “maximum likelihood type”. That is because M-estimation is a generalization of maximum likelihood estimates (MLE). Suppose the regression model is

$$y_i = \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i, \quad (2.1)$$

where ε_i has a density $\frac{1}{\sigma} f(\frac{\varepsilon}{\sigma})$ and σ is a scale parameter. Then y_i 's have density functions $\frac{1}{\sigma} f(\frac{y_i - \mathbf{x}'_i \boldsymbol{\beta}}{\sigma})$ and the *likelihood function* for $\boldsymbol{\beta}$ and σ is

$$L(\boldsymbol{\beta}, \sigma) = \prod_{i=1}^n \frac{1}{\sigma} f\left(\frac{y_i - \mathbf{x}'_i \boldsymbol{\beta}}{\sigma}\right) = \frac{1}{\sigma^n} \prod_{i=1}^n f\left(\frac{y_i - \mathbf{x}'_i \boldsymbol{\beta}}{\sigma}\right). \quad (2.2)$$

Based on this, the log likelihood is given by:

$$l(\boldsymbol{\beta}, \sigma) = -n \log \sigma + \sum_{i=1}^n \log f\left(\frac{y_i - \mathbf{x}'_i \boldsymbol{\beta}}{\sigma}\right).$$

Setting $\rho(x) = -\log f(x)$, and letting $e_i(\boldsymbol{\beta}) = y_i - \mathbf{x}'_i \boldsymbol{\beta}$, we have

$$l(\boldsymbol{\beta}, \sigma) = -[n \log \sigma + \sum_{i=1}^n \rho\left(\frac{e_i(\boldsymbol{\beta})}{\sigma}\right)].$$

To maximize this equation is equivalent to minimizing

$$\log \sigma + \frac{1}{n} \sum_{i=1}^n \rho\left(\frac{e_i(\boldsymbol{\beta})}{\sigma}\right). \quad (2.3)$$

Assuming σ is a fixed value, the M-estimator minimizes the *objective function*

$$\sum_{i=1}^n \rho(r_i) = \sum_{i=1}^n \rho\left(\frac{e_i(\boldsymbol{\beta})}{\sigma}\right), \quad (2.4)$$

where $r_i = \frac{e_i(\boldsymbol{\beta})}{\sigma}$ are called *standardized residuals*. Let $\psi(x) = \rho'(x) = -f'(x)/f(x)$, differentiating (2.4) with respect to $\boldsymbol{\beta}$, assuming σ is fixed, and setting the partial derivatives to 0, we get the normal equations

$$\sum_{i=1}^n \psi\left(\frac{e_i(\boldsymbol{\beta})}{\sigma}\right) \mathbf{x}_i = \mathbf{0}. \quad (2.5)$$

To solve (2.5) we define the *weight function*

$$W(x) = \begin{cases} \psi(x)/x, & \text{if } x \neq 0; \\ \psi'(0), & \text{if } x=0. \end{cases} \quad (2.6)$$

and let $w_i = W(r_i)$. Then equations (2.5) can be written as

$$\sum_{i=1}^n w_i (y_i - \mathbf{x}_i' \boldsymbol{\beta}) \mathbf{x}_i = \mathbf{0}. \quad (2.7)$$

Note that if w_i is known, the above equation generates the weighted least squares estimate. For robust estimate, usually $\psi(\cdot)$ is bounded. Therefore, the weights w_i s will be small if the standardized residuals r_i s are large and thus downweight the effects of outliers. The solution of (2.7) can be found by iterating between w_i and $\boldsymbol{\beta}$:

1. Set an initial estimates $\boldsymbol{\beta}^{(0)}$, for example LSEs.
2. Calculate standardized residuals $r_i^{(k-1)}$ and weights $w_i^{(k-1)} = W(r_i^{(k-1)})$ at each iteration k from the previous iteration.
3. Update $\boldsymbol{\beta}$ by

$$\boldsymbol{\beta}^k = [\mathbf{X}' \mathbf{W}^{(k-1)} \mathbf{X}]^{-1} \mathbf{X}' \mathbf{W}^{(k-1)} \mathbf{y},$$

where $\mathbf{W}^{(k-1)} = \text{diag}\{w_i^{(k-1)}\}$ is the current weight matrix.

Repeat step 2 and step 3 until the estimated coefficients converge.

In equation (2.4), a reasonable ρ should satisfy the following properties:

- $\rho(e) \geq 0$ (nonnegative);
- $\rho(e) = \rho(-e)$ (symmetric);
- $\rho(0) = 0$;
- $\rho(e_i) \geq \rho(e_j)$ for $|e_i| > |e_j|$ (nondecreasing function of $|x|$).

Several choices of ρ have been proposed. If $\rho(x) = \frac{1}{2}x^2$, then the solution of the normal equations is the LSE which we introduced in chapter 1. If $\rho(x) = |x|$, then the objective function becomes:

$$\sum_{i=1}^n |e_i(\hat{\beta})| = \min.$$

This is known as the L_1 estimate, which is also called the Least Absolute Deviation (LAD) estimate or median regression estimate. Another two popular choices of ρ are the *Huber function* and the Tukey *bisquare* (or *biweight*) function. For the Huber function,

$$\rho_H(e) = \begin{cases} \frac{1}{2}e^2, & \text{for } |e| \leq k; \\ k|e| - \frac{1}{2}k^2, & \text{for } |e| > k. \end{cases}$$

The corresponding weight function is

$$w_H(e) = \begin{cases} 1, & \text{for } |e| \leq k; \\ k/|e|, & \text{for } |e| > k. \end{cases},$$

where k is called a *tuning constant*. Smaller values of k produce more resistance to outliers, but comes at the price of loss in efficiency under the normal distribution. Usually, the tuning constant is picked to give reasonably high efficiency in the normal case, for example, $k = 1.345\sigma$ produces 95% efficiency and can still offer some protection against outliers. For the Tukey bisquare function,

$$\rho_B(e) = \begin{cases} \frac{k^2}{6} \{1 - [1 - (\frac{e}{k})^2]^3\}, & \text{for } |e| \leq k; \\ k^2/6, & \text{for } |e| > k. \end{cases}$$

The corresponding weight function is

$$w_B(e) = \begin{cases} [1 - (\frac{e}{k})^2]^2, & \text{for } |e| \leq k; \\ 0, & \text{for } |e| > k. \end{cases}$$

Generally, $k = 4.685\sigma$ is used to produce 95% efficiency .

Figure 2.1 compares the $\rho(\cdot)$, $\psi(\cdot)$ and weight functions $w(\cdot)$ for the above four M-estimators: the familiar least squares estimate; the least absolute deviation estimate; the Huber estimate; and the Tukey bisquare estimate. The objective functions for least squares, LAD, and Huber increase without bound as e departs from 0, but the least squares objective function increases more rapidly. In contrast, the objective function for Tukey bisquare eventually levels off (for $|e| > k$). For weight functions, least squares assigns equal weight to each observation; the weights for the Huber estimate decline when $|e| > k$; and the weights for the LAD and Tukey bisquare estimators decline as soon as e departs from 0, but LAD declines more rapidly.

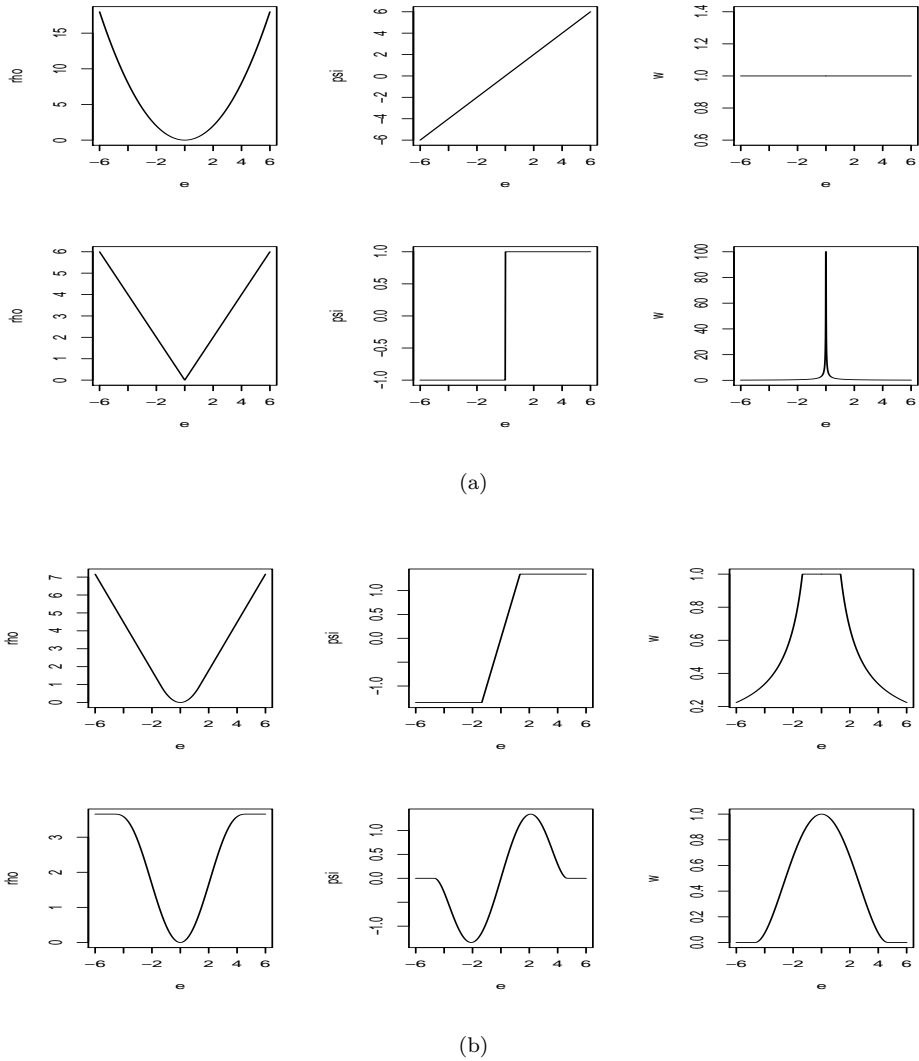


Figure 2.1: Figure (a) shows the objective, ψ , and weight functions for the least-squares(top) and LAD (bottom) estimators. Figure (b) shows the objective, ψ , and weight functions for the Huber (top) and bisquare (bottom) estimators. The tuning constants for these graphs are $k=1.345$ for the Huber estimator and $k=4.685$ for the bisquare. (One way to think about this scaling is that the standard deviation of the errors, σ , is taken as 1.)

The above calculations are based on the assumption that σ is known. However, in reality, σ is usually unknown. In this situation, we can compute it simultaneously by adding a scale M-estimating equation to equation (2.5). Differentiating (2.3) with respect to σ can get $\frac{1}{n} \sum_{i=1}^n \rho_{scale}(\frac{e_i(\boldsymbol{\beta})}{\hat{\sigma}}) = 1$, where $\rho_{scale}(t) = t\psi(t)$. Thus, in general, an *M-estimate of scale* is the estimate which satisfies an equation of the form

$$\frac{1}{n} \sum_{i=1}^n \rho\left(\frac{e_i(\boldsymbol{\beta})}{\hat{\sigma}}\right) = \delta,$$

where ρ is a ρ -function and δ is a positive constant.

M-estimators with Huber function or Tukey bisquare function are robust to outliers in the response variable with high efficiency. However, M-estimators are just as vulnerable as least squares estimates to high leverage outliers. In fact, the BP (breakdown point) of M-estimates is $1/n \rightarrow 0$ (Rousseeuw and Yohai 1984). Figure 2.2 illustrates an effect of high leverage outliers on the least squares estimator and M-estimators. It shows that both methods are very sensitive to high leverage outliers.

2.2 Least Median of Squares estimate

Since M-estimators are not robust to high-leverage outliers, we want to find some methods that can have high BP. Siegel (1982) defined the *least median of squares* (LMS) estimator as

$$\min_{\boldsymbol{\beta}} \text{med}_i \{e_i^2\}, \tag{2.8}$$

which replaced the sum by the median in the LSE. This proposal is essentially based on an idea by Hampel (1975).

The advantage of LMS estimate is that it is very robust to outliers in both the y direction and the leverage points and it has been shown that LMS has the highest possible BP of 0.5. Unfortunately, it has a very low efficiency and it can be unstable. Moreover, due to its slow convergence rate of $n^{-\frac{1}{3}}$, LMS estimate does not have a well-defined influence function. Because of these properties, LMS estimate is usually used as the initial estimate of the residuals for other more efficient methods such as MM-estimators (Yohai, 1987).

2.3 Least Trimmed Squares estimate

Another regression estimator that has BP of nearly 50 percent is the *least trimmed squares* (LTS) estimator proposed by Rousseeuw and Yohai (1984). This estimator chooses the regression coefficients $\boldsymbol{\beta}$ to minimize the sum of the smallest h of the squared residuals and is defined as:

$$\hat{\boldsymbol{\beta}}_{LTS} = \arg \min_{\boldsymbol{\beta}} \sum_{i=1}^h e_{(i)}^2(\boldsymbol{\beta}), \tag{2.9}$$

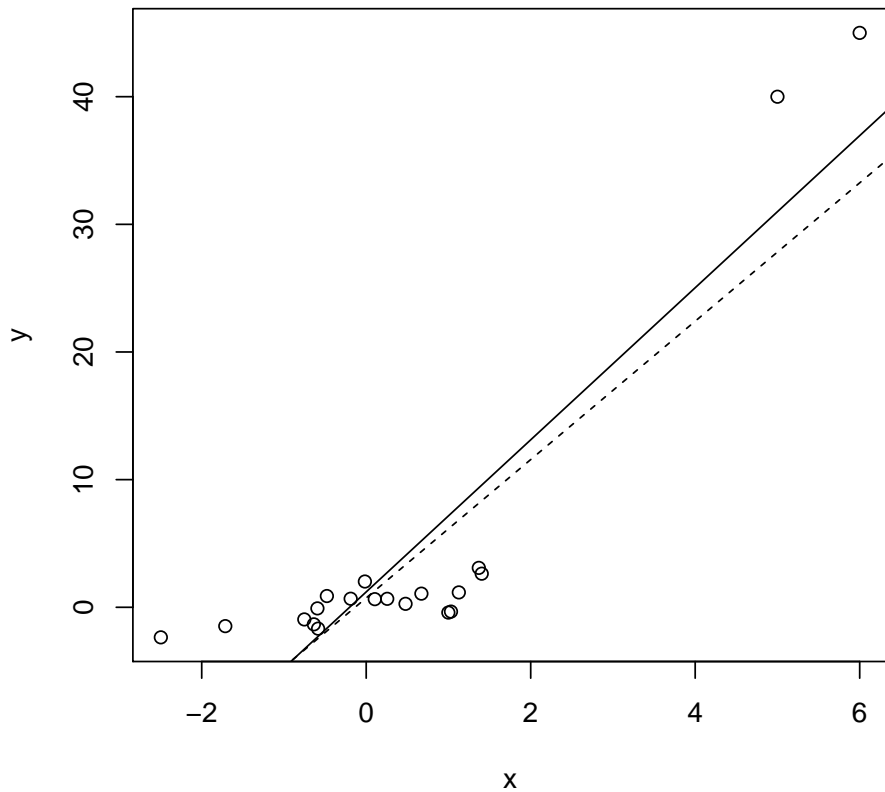


Figure 2.2: Least squares estimate (solid line) and M-estimate with Huber function (dashed line) for a dataset contains 20 observations two of which are high leverage outliers.

where $e_{(i)}^2(\boldsymbol{\beta})$ represents the i -th ordered squared residuals $e_{(1)}^2(\boldsymbol{\beta}) \leq \dots \leq e_{(n)}^2(\boldsymbol{\beta})$ and h is called the trimming constant which has to satisfy $\frac{n}{2} < h \leq n$. This constant h determines the BP of the LTS estimator. Typically, $h = \lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$ can attain the maximum $BP = (\lfloor (n-p)/2 \rfloor + 1)/n$, where $\lfloor \cdot \rfloor$ means rounding down to the next smallest integer. When $h = n$, LTS is exactly equivalent to LS estimator whose BP is 0. Like LMS, LTS has a high BP but low efficiency. Although its convergence rate of $n^{-\frac{1}{2}}$ (Rousseeuw 1983) makes it asymptotical normal which is better than LMS, it still suffers a very low efficiency of only 7%. The reason that LTS estimator is discussed in this paper is that LTS is often suggested as the starting point for more efficient procedures.

2.4 S-estimate

To find a simple high-breakdown regression estimator which shares the flexibility and nice asymptotic properties of M-estimator, Rousseeuw and Yohai (1984) introduced the S-estimator. They called it S-estimator because it is derived from the M-scale estimate equation:

$$\frac{1}{n} \sum_{i=1}^n \rho\left(\frac{e_i(\boldsymbol{\beta})}{\hat{\sigma}}\right) = \delta. \quad (2.10)$$

In M-estimates, when σ is unknown, we use this equation with equation (2.5) to get the scale parameter and regression parameter simultaneously. Let $\delta = E_{\Phi}[\rho(x)]$, where Φ represents the standard normal, and let

$$d(\mathbf{e}) = \#\{i : 1 \leq i \leq n, e_i = 0\}/n.$$

When $d(\mathbf{e}) < 1 - \delta/a$ (where a is the upper bounded of ρ and $a \in (0, \infty)$), equation (2.10) has a unique positive solution; if $d(\mathbf{e}) = 1 - \delta/a$, equation (2.10) may have infinite solutions which include $\sigma = 0$; and if $d(\mathbf{e}) > 1 - \delta/a$, then no solution for equation (2.10) exists. To avoid indeterminacies, we combine the last two situations together and define whenever $d(\mathbf{e}) \geq 1 - \delta/a$, $\sigma(\mathbf{e}) = 0$.

To define S-estimator we let ρ satisfy

(A1):

(i) symmetric, continuously differentiable and $\rho(0) = 0$;

(ii) there exists $c > 0$ such that ρ_{scale} is strictly increasing on $[0, c]$ and constant on $[c, \infty)$.

For each vector $\boldsymbol{\beta}$, using (2.10) we can calculate the dispersion of residuals $\hat{\sigma}(e_1(\boldsymbol{\beta}), \dots, e_n(\boldsymbol{\beta}))$, where ρ satisfies (A1). Then the S-estimator $\hat{\boldsymbol{\beta}}$ is defined by

$$\arg \min_{\boldsymbol{\beta}} \hat{\sigma}(e_1(\boldsymbol{\beta}), \dots, e_n(\boldsymbol{\beta})). \quad (2.11)$$

The breakdown point of S-estimator can be made to obtain 0.5 with an appropriate ρ -function. A popular choice of ρ -function is the bisquare (or biweight) function:

$$\rho(x) = \begin{cases} \frac{x^2}{2} - \frac{x^4}{2k^2} + \frac{x^6}{6k^4}, & \text{for } |x| \leq k; \\ \frac{k^2}{6}, & \text{for } |x| > k. \end{cases} \quad (2.12)$$

When $k = 1.547$, BP is equal to 0.5 but at the cost of low efficiency which is only 28.7%. Hossjer (1992) proved that there exists a trade-off between efficiency and robustness. In other words, S-estimator cannot achieve simultaneously high breakdown point and high efficiency under the normal model. To make up this, we chose the S-estimator with high breakdown point as the initial estimator for the one-step M-estimator (Bickel 1975), so that the resulting one-step M-estimator has high efficiency and also inherits the 0.5 BP from the first stage (more details in Chapter 2.7).

2.5 τ -estimate

As shown above, some estimates might have a high breakdown point, but at the cost of low efficiency under normal errors. To solve this problem, Yohai and Zamar (1988) defined a broader class of scale estimates, called τ -estimate, and singled out a subclass that can achieve a high BP and high efficiency at the same time.

Let ρ and ρ_1 be two bounded continuous ρ -function, and let $\hat{\sigma}(\mathbf{e})$ be a robust M-scale estimate based on ρ . Then let $\mathbf{e} = (e_1, \dots, e_n)$ and define the scale τ as

$$\tau^2(\mathbf{e}) = \hat{\sigma}^2(\mathbf{e}) \frac{1}{n} \sum_{i=1}^n \rho_1\left(\frac{e_i}{\hat{\sigma}(\mathbf{e})}\right). \quad (2.13)$$

Then the regression τ -estimate $\hat{\boldsymbol{\beta}}$ is defined by

$$\arg \min_{\boldsymbol{\beta}} \tau(\mathbf{e}(\boldsymbol{\beta})). \quad (2.14)$$

Yohai and Zamar (1988) showed that the BP depends only on ρ . Therefore, we choose an appropriate ρ satisfying all the assumptions so that the τ estimator has a high BP, and at the same time, choose a ρ_1 so that the τ estimator has arbitrarily high efficiency under the normal model. It is also shown that by adequately choosing ρ and ρ_1 , the estimate can attain the maximum BP (0.5) for regression estimates and arbitrarily efficient at the normal distribution (close to 1). For example, when the bisquare family of ρ function (2.12) is used, if we take $k = 1.56$ and $E_{\Phi}(\rho) = 0.203$ for ρ and $k_1 = 6.08$ for ρ_1 , the resulting τ estimator has simultaneously a breakdown point of 0.5 and an efficiency of 0.95 under normal distribution.

The computing algorithm of this estimate is a modification of the iterative weighted least squares algorithm for M estimates (see Yohai and Zamar, 1988). And in that paper, they also showed that a τ estimate is asymptotical equivalent to an M estimate with a ψ function given by a linear combination of ρ' and ρ'_1 with coefficients depending on the data.

2.6 M M-estimate

Another class of robust estimates which have high breakdown point and high efficiency under normal error is MM-estimates. Yohai (1987) introduced the MM-estimates for robust regression. This estimate is defined in three stages:

STAGE 1. Compute an initial consistent robust estimate $\hat{\boldsymbol{\beta}}_0$ with high BP, possibly 0.5, but not necessarily efficient.

STAGE 2. Compute the M-scale $\hat{\sigma}$ of the residuals $e_i(\hat{\boldsymbol{\beta}}_0)$ using equation (2.10), using a function ρ_0 satisfying (A1) and choosing a constant δ such that $\delta/a = 0.5$, where $a = \sup \rho_0(e)$. Thus, the asymptotic BP of $\hat{\sigma}$ is 0.5 (proved by Huber, 1981).

STAGE 3. Let ρ_1 be another ρ -function satisfying (A1) and such that

$$\sup \rho_1(e) = \sup \rho_0(e) = a, \quad (2.15)$$

$$\rho_1(e) \leq \rho_0(e). \quad (2.16)$$

Let $\psi_1 = \rho_1'$, $L(\boldsymbol{\beta}) = \sum_{i=1}^n \rho_1(\frac{e_i(\boldsymbol{\beta})}{\hat{\sigma}})$, and $\rho_1(0/0) = 0$. Then the MM-estimate $\hat{\boldsymbol{\beta}}_1$ is defined as any solution to the equation

$$\sum_{i=1}^n \psi_1(\frac{e_i(\boldsymbol{\beta})}{\hat{\sigma}}) \mathbf{x}_i = \mathbf{0}, \quad (2.17)$$

that also satisfies

$$L(\hat{\boldsymbol{\beta}}_1) \leq L(\hat{\boldsymbol{\beta}}_0). \quad (2.18)$$

Yohai (1987) showed that any value of $\boldsymbol{\beta}$ which satisfies (2.17) and (2.18), e.g., a local minimum, will have the same efficiency as the global minimum and its BP is not less than that of $\hat{\boldsymbol{\beta}}_0$. Thus, although the absolute minimum of $L(\boldsymbol{\beta})$ exists, it is not necessary to find it.

In the first stage, the robust initial estimate $\hat{\boldsymbol{\beta}}_0$ should satisfy regression, scale and affine equivalent and also have a high BP. LMS-, LTS-, and S-estimates are possible candidates. For stage 2, one way of choosing ρ_0 and ρ_1 is as follows. Let ρ be a function satisfying (A1) and let $\rho_0(e) = \rho(e/k_0)$ and $\rho_1(e) = \rho(e/k_1)$. In order to satisfy (2.16), we must have $0 < k_0 < k_1$. The value of k_0 should be chosen such that $\delta/a = 0.5$ holds. It has also been proven that the asymptotic variance of MM-estimate depends only on k_1 : the larger k_1 , the higher the efficiency at the normal distribution. Therefore, similarly to τ -estimate, MM-estimate can attain high efficiency by choosing an appropriate k_1 without affecting its breakdown point, which depends only on the choice of k_0 . For example, let ρ be the ρ -function of bisquare family (2.12) and $k_0 = 1.56$ which can hold the equations in stage 2, the corresponding $\delta = 0.0833$ and $k_1 = 4.68$ which gives efficiency 0.95 for normal errors. However, Maronna, Martin, and Yohai (2006) showed that there is a basic trade-off between normal efficiency and bias under contamination and Yohai (1987) also indicated that MM-estimates with larger values of k_1 are more sensitive to outliers than the estimates corresponding to smaller values of k_1 . It is therefore important to choose the efficiency to balance these. Maronna, Martin, and Yohai (2006) recommended the efficiency of 0.85 which gives a small bias while retaining a sufficiently high efficiency. And for the above example, this require $k_1 = 3.44$ which is a smaller value compare to 4.86.

The numerical computation of MM-estimate is a modified version of the IRWLS (iterated weighted least-squares) algorithm used for computing M-estimate. First we define the weight function for $\mathbf{t} \in R^p$:

$$w_i(\mathbf{t}) = \frac{\psi_1(e_i(\mathbf{t})/\hat{\sigma})}{e_i(\mathbf{t})/\hat{\sigma}}. \quad (2.19)$$

Also define

$$\mathbf{g}(\mathbf{t}) = \frac{1}{\hat{\sigma}^2} \sum_{i=1}^n w_i(\mathbf{t}) r_i(\mathbf{t}) \mathbf{x}_i, \quad (2.20)$$

and

$$\mathbf{M}(\mathbf{t}) = \frac{1}{\hat{\sigma}^2} \sum_{i=1}^n w_i(\mathbf{t}) \mathbf{x}_i \mathbf{x}_i', \quad (2.21)$$

where $-\mathbf{g}(\mathbf{t})$ is the gradient of $L(\mathbf{t})$. To ensure (2.18) holds, we modify IRWLS as follows: take $0 < \delta < 1$, and find an integer k such that

$$L(\mathbf{t}^{(j)} + \Delta(\mathbf{t}^{(j)})/2^k) \leq L(\mathbf{t}^{(j)}) - \delta(\Delta(\mathbf{t}^{(j)})/2^k)'g(\mathbf{t}^{(j)}), \quad (2.22)$$

where $\Delta(\mathbf{t}) = \mathbf{M}^{-1}(\mathbf{t})\mathbf{g}(\mathbf{t})$. Let $k_{1,j}$ be the minimum of such k 's, $0 \leq k \leq k_{1,j}$, and let $k_{2,j}$ be the value of k which gives the minimum of $L(\mathbf{t}^{(j)} + \Delta(\mathbf{t}^{(j)})/2^k)$. Then the recursion step is defined by

$$\mathbf{t}^{(j+1)} = \mathbf{t}^{(j)} + (1/2^{k_{2,j}})\Delta(\mathbf{t}^{(j)}), \quad (2.23)$$

starting with $\mathbf{t}^{(0)} = \hat{\boldsymbol{\beta}}_0$. Thus any limit point of the sequence $\mathbf{t}^{(j)}$ is an MM-estimate.

2.7 GM-estimate

The M-estimate is the most popular robust estimate but it has a low BP due to the failure to account for high leverage outliers. In response to this problem, bounded influence generalized M estimate (GM estimate) were proposed to produce stable results when there are outliers in the explanatory variables. Examples of these proposals include Mallows (1975), Hampel (1978), Krasker (1980), and Krasker and Welsch (1982). The goal of GM-estimate is to create weights that consider the outliers both in the y-direction and the leverage points. Using a standard M-estimate to deal with the vertical outliers, the weight functions can downweight leverage points so that observations with high leverage receive less weight than observations with small leverage. The general GM class of estimators is defined by

$$\sum_{i=1}^n w_i(\mathbf{x}_i) \psi\left\{\frac{e_i(\hat{\boldsymbol{\beta}})}{v(\mathbf{x}_i)\hat{\sigma}}\right\} \mathbf{x}_i = 0, \quad (2.24)$$

where ψ is the score function (as in the case of M-estimate).

2.7.1 Mallows GM-estimate

The first GM-estimate was proposed by Mallows (1975). For Mallows GM-estimate, the v_i in equation (2.24) is equal to 1 and $w_i = \sqrt{1 - \hat{h}_i}$ where \hat{h}_i is the hat value. Hat values range from 0 to 1, so weight function downweights the high leverage points. However, there are possible leverage points whose responses fall in line with the pattern in the bulk of the data. In these cases Mallows GM-estimate cannot distinguish these “good” leverage points and will down weight them which result in a loss of efficiency.

2.7.2 Schweppe GM-estimate

Another GM-estimate is called Schweppe GM-estimate. This method adjusted the leverage weights according to the size of the residual e_i by using $v_i = w_i$, where w_i is the weight function which is the same as Mallows GM-estimate and equal to $\sqrt{1 - h_i}$ (see Handschin et al. 1975). However, since the weight function of this estimate only depends on x values without considering how the corresponding y values fit with the pattern of the bulk of the data, efficiency is still hindered (Krasker and Welsch 1982). Moreover, Carroll and Welsh (1988) suggested that the Schweppe estimate is not consistent when the errors are asymmetric.

The breakdown points for the above two GM-estimates, although better than for regular M-estimate, are at most $1/(p + 1)$, where p is the number of predictor variables (Maronna, Bustos and Yohai 1979). Thus, as dimensionality increases, their BP tends to 0.

2.7.3 One-step GM-estimate

We usually refer to the property of high BP as global stability and that of bounded influence as local stability. To combine these two stabilities as well as a degree of efficiency under the Gauss-Markov assumptions, one-step GM estimate was proposed by several authors (Bickel, 1975; Jureckova and Portnoy, 1987; Giltinan, Carroll, and Ruppert, 1986). The strategy of this estimate is as follows: Start with a high breakdown point estimator such as LTS or LMS and perform one iteration towards solution of the GM-estimate equations. And it takes the form

$$\hat{\beta} = \hat{\beta}_0 + H_0^{-1}g_0, \quad (2.25)$$

where $\hat{\beta}_0$ is a high breakdown preliminary estimate with BP at least m/n (usually we take $m = [(n - p)/2] + 1$), $g_0 = \hat{\sigma}_0 \sum_{i=1}^n \Psi(e_i/\hat{\sigma}_0)w_i x_i$. There are two viable choices for H_0 :

- Newton-Raphson: $H_0 = \sum_{i=1}^n w_i x_i x_i' \psi^{(1)}(e_i/\hat{\sigma}_0)$;
- Scoring: $H_0 = n^{-1} \sum_{i=1}^n \psi^{(1)}(e_i/\hat{\sigma}_0) \sum_{j=1}^n w_j x_j x_j'$.

These two methods are asymptotically equivalent if the errors are independent, identically and symmetrically distributed. For w_i , we use Mallows weights:

$$w_i = \min\left[1, \frac{b}{(x_i - m_x)' C_x^{-1} (x_i - m_x)}\right]^{\alpha/2}$$

and set b equal to the $(1 - r)$ quartile of the chi-squared distribution with $df = p - 1$, where $r = 0.1$ or 0.5 . In the formula, when $\alpha = 0$, we call it one-step Huber estimate which is discussed by Bickel (1975) and Jureckova and Portnoy (1987); when $\alpha = 1$, we usually use it for GM estimate; and when $\alpha = 2$, Giltinan, Carroll and Ruppert (1986) used it to force a bounded change of variance.

Simpson, Ruppert, and Carroll (1992) showed that under reasonably general conditions, one-step Mallows estimates inherit the breakdown properties of the preliminary estimates of the regression parameters and

the multivariate location and scale estimates of the design x 's. However, the estimated standard errors of this estimate may change radically with deletion of a single observation.

2.8 Robust and Efficient Weighted Least Squares estimate

In the model (1.1), we assume the error terms $\{e_i\}$ are *iid* unobservable random variables with unknown distribution $F_0(\frac{\cdot}{\sigma})$ for some scale parameter $\sigma > 0$. F_0 is symmetric about 0.

Let $\hat{\beta}_0$ and $\hat{\sigma}_0$ be the initial robust estimators of regression and scale, respectively. If $\hat{\sigma}_0 > 0$, the *standardized residuals* defined in Chapter 2.1 are:

$$r_i = \frac{y_i - \mathbf{x}_i' \hat{\beta}_0}{\hat{\sigma}_0}. \quad (2.26)$$

Then, a large $|r_i|$ implies that (\mathbf{x}_i, y_i) is an outlier. The idea of weighting is: we set a cutoff point say t_0 , if $|r_i| \leq t_0$ we will keep that point, but if $|r_i| > t_0$ we will treat it as an outlier and remove it by weighting it 0. As we know that this weighting step improves the efficiency under normal errors and also maintains the breakdown point of the initial estimator. However, He and Portnoy (1992) showed that, it cannot be asymptotically efficient. To obtain the full asymptotic efficiency under normal errors without less the breakdown point of the initial estimator, Gervini and Yohai (2002) suggested using the adaptive cutoff values which lead to a new class of estimators called *robust and efficient weighted least squares estimators* (REWLSes). Instead of setting a particular fixed value to t_0 , this method adaptively calculates t_n from the data.

To define the adaptive cutoff values, first let the *empirical distribution function* of the standardized absolute residuals be

$$F_n^+(t) = \frac{1}{n} \sum_{i=1}^n I(|r_i| \leq t). \quad (2.27)$$

Suppose the distribution function of the absolute errors under the actual model is $F_0^+(t)$. If $F_n^+(t) < F_0^+(t)$ under a large t , then we can say there are outliers in the sample. However, in reality, we will never know the actual distribution of the errors, so a hypothetical $F = \Phi$ must be used instead of F_0 . Secondly, we define a measure of the proportion of outliers in the sample:

$$d_n = \max_{i > i_0} \left\{ F^+(|r_{(i)}|) - \frac{i}{n} \right\}^+, \quad (2.28)$$

where $\{\cdot\}^+$ denotes the positive part, F^+ denotes the distribution of $|\mathbf{X}|$ when $\mathbf{X} \sim F$. Let $|r_{(1)}| \leq \dots \leq |r_{(n)}|$ be the order statistics of the standardized absolute residuals. Then $i_0 = \max\{i : |r_{(i)}| < \eta\}$. Here η is some large quartile of F^+ (Rousseeuw and Leroy (1987) used $\eta = 2.5$). Thus those $[nd_n]$ observations (here $[nd_n]$ is the largest integer less than or equal to nd_n) with largest standardized absolute residuals are eliminated by using $t_n = |r_{(i_n)}|$ with $i_n = n - [nd_n]$. Observing that $i_n > i_0$ and $t_n > \eta$, another way to define the

adaptive cut-off value is:

$$t_n = \min\{t : F_n^+(t) \geq 1 - d_n\}, \quad (2.29)$$

which means the same thing as above. With this t_n , we define weights of the form $w_i = w(|r_i|/t_n)$. When $w(u) = I(u < 1)$ we called it the *hard-rejection weight* which is the most common weight function. However, in general, we will only require $w(u)$ to satisfy the following three properties:

1) $w(0) = 1$.

2)

$$\begin{cases} w(u) > 0, & \text{if } 0 < u < 1; \\ w(u) = 0, & \text{if } u \geq 1. \end{cases}$$

3) $u \in [0, \infty)$, $w(u) \in [0, 1]$, $w(u)$ is nonincreasing, right continuous and continuous in a neighborhood of 0.

The property 2) ensures that observations with large residuals are completely eliminated. Let $\mathbf{W} = \text{diag}(w_1, \dots, w_n)$ and $\mathbf{Y} = (y_1, \dots, y_n)'$, REWLSE is defined as:

$$\hat{\beta}_{REWLSE} = \hat{\beta}_1 = \begin{cases} (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}, & \text{if } \hat{\sigma}_0 > 0; \\ \hat{\beta}_0, & \text{if } \hat{\sigma}_0 = 0. \end{cases}$$

Gervini and Yohai (2002) showed that t_n remains bounded in the presence of outliers which implies that $\hat{\beta}_1$ keeps the finite sample and asymptotic breakdown points of $\hat{\beta}_0$. They proved that the BP of REWLSE satisfies $\varepsilon_n^*(\hat{\beta}_1, \mathbf{Z}) \geq \varepsilon_n^*(\hat{\beta}_0, \mathbf{Z}) - 1/n$. On the other hand, the REWLS estimates are asymptotically equivalent to the LS estimates and hence asymptotically efficient under the normal errors. That is because when F_0 is of unbounded support but of lighter tails than F , its cutoff value will approach infinity under the model and then $w(|r_i|/t_n) \rightarrow 1$. The same happens if F_0 is of bounded support with lighter tails than F but $w(u)$ should be the hard-rejection function.

Chapter 3

Comparing Various Estimators

Table 3.1 summarizes the robustness and efficiency attributes of most of the estimators we have discussed in Chapter 2. The breakdown point, whether or not the estimator has a bounded influence function, and the approximate asymptotic efficiency of the estimator relative to the LSE are reported. When compared in terms of breakdown point, it is obvious that M-estimate and LAD have the BP as low as LSE which indicates a single discrepant observation can render these estimates useless. Sometimes the LAD and M-estimate perform no better than LSE. Although GM-estimate has a higher BP, however, when the number of parameters increase, its BP can be very small. On the other hand, when looking at the high BP estimates we should also be cautious of the low efficiency. Here, LMS, LTS, and S-estimates have asymptotic efficiency less than 0.4. When used in combination with more resistant estimators, τ -estimate, MM-estimate, one-step GM-estimate, and REWLS-estimate can attain a nearly optimal efficiency and maximum breakdown point at the same time.

3.1 Simulation Study

According to the various types of outliers introduced in Chapter 1, we now compare different methods and report the mean squared errors (MSE) of the parameter estimates for each estimation method. We explore eight different regression estimates: LSE, M-estimate using Huber weights (M_H), M-estimate using Tukey weights (M_T), LMS, LTS, S-estimate, MM-estimate (using bisquare weights and $k_1 = 4.68$), and REWLS. We use two models to compare the performance of these eight methods:

Model 1:

$$Y = X + \varepsilon,$$

where $X \sim N(0, 1)$;

Table 3.1: Breakdown Points and Asymptotic Efficiencies of Various Regression Estimators

	Estimator	Breakdown Point	Bounded	Asymptotic Efficiency
High BP	LMS	0.5	Yes	0
	LTS	0.5	Yes	0.07
	S-estimates	0.5	Yes	0.29
	τ -estimates	0.5	Yes	0.95
	MM-estimates	0.5	Yes	0.85
	GM-estimates(one-step)	0.5	Yes	0.95
	REWLS-estimates	0.5	No	1.00
Low BP	GM-estimates(Mallows,Schweppe)	$1/(p+1)$	Yes	0.95
	M-estimates	$1/n$	No	0.95
	LAD	$1/n$	No	0.64
	LSE	$1/n$	No	1.00

Model 2:

$$Y = X_1 + X_2 + X_3 + \varepsilon,$$

where $X_i \sim N(0, 1)$, $i = 1, 2, 3$ and X_i s are independent.

We consider the following six cases for the error density of ε :

Case I: $\varepsilon \sim N(0, 1)$ - standard normal distribution.

Case II: $\varepsilon \sim t_1$ - t-distribution with degrees of freedom 1 (Cauchy distribution).

Case III: $\varepsilon \sim t_3$ - t-distribution with degrees of freedom 3.

Case IV: $\varepsilon \sim N(0, 1)$ with 10% identical outliers in y direction (where we let the first 10% of y 's equal to 30).

Case V: $\varepsilon \sim N(0, 1)$ with 10% identical high leverage outliers (where we let the first 10% of x 's equal to 10 and their correspond y 's equal to 50).

Case VI: $\varepsilon \sim 0.95N(0, 1) + 0.05N(0, 10^2)$ - contaminated normal mixture.

For each of those we use R to simulate 200 samples and get the mean squared errors of the parameter estimates for each estimation method. Table 3.2 and 3.3 show the mean squared errors (MSE) for model 1 and sample size $n=20$ and $n=100$, respectively; Table 3.4 and 3.5 show the mean squared errors (MSE) for model 2 and sample size $n=20$ and $n=100$, respectively. Based on these four tables, we can see that MM-estimates and REWLS estimates have the overall best performance throughout most cases and they are consistent for different sample sizes. For Case I, LSE has the smallest MSE which is reasonable since under normal errors LSE is the best estimate; M-estimates, MM-estimate, and REWLS estimate have similar MSE to LSE, due to their high efficiency property; LMS, LTS, and S estimate have relative larger MSE

due to their low efficiency. For Case II, it can be seen that LSE has much larger MSE than other robust estimators; M-estimates, MM-estimate and REWLS estimate have similar MSE to S-estimate. For Case III, M-estimate, MM-estimate, and REWLS work better than other estimates. From Case IV, we can see that when the data contain outliers in the y-direction, LSE is much worse than any other robust estimates; MM-estimates, REWLS, and M_T are better than other robust estimators. For Case V, since there are high leverage outliers, similar to LSE, both M_T and M_H perform poorly; MM-estimate and REWLS work better than other robust estimates for this case. Finally for Case VI, M-estimates, MM-estimate, and REWLS estimates have smaller MSE than others.

In summary, LSE only works well when there are no outliers since it is very sensitive to outliers. M-estimates (M_H and M_T) work well if the outliers are in y direction but is also sensitive to the high leverage outliers. To better compare the performance of LMS, LTS, S, MM, and REWLS, Figure 3.1 and Figure 3.2 show the plot of their MSE versus each case for slope and intercept parameters, respectively, for model 1 when $n = 20$. Since the lines for LTS and LMS estimates are above the other lines, we conclude that S-estimate, MM-estimate, and REWLS are better estimates than LTS and LMS. In addition, it seems that REWLS has the overall best performance and MM-estimate has overall better performance than S-estimate. Plot for intercept tells a similar story.

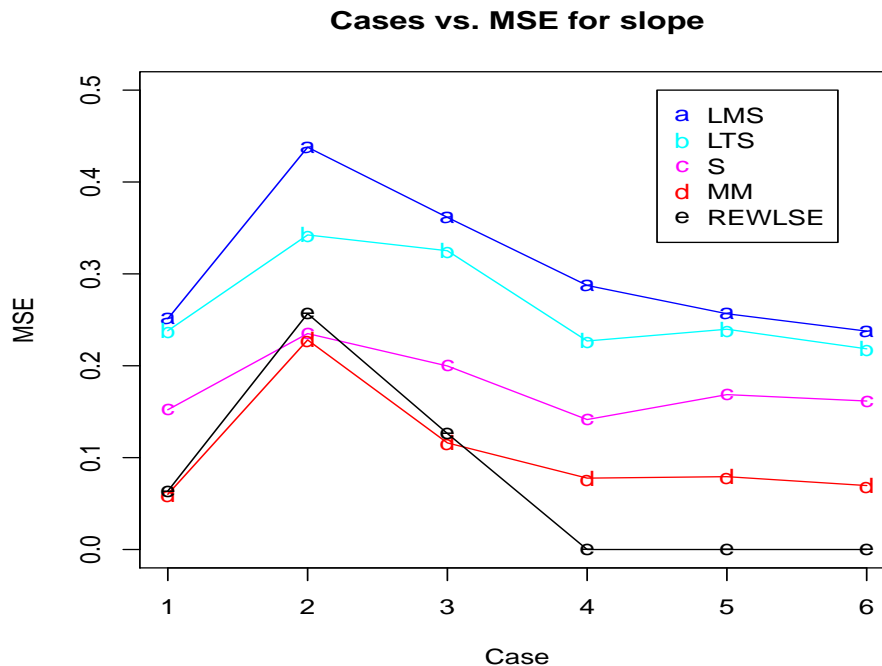


Figure 3.1: Plot of MSE of slope estimates vs. different cases for LMS, LTS, S, MM, and REWLS, for model 1 when $n = 20$.

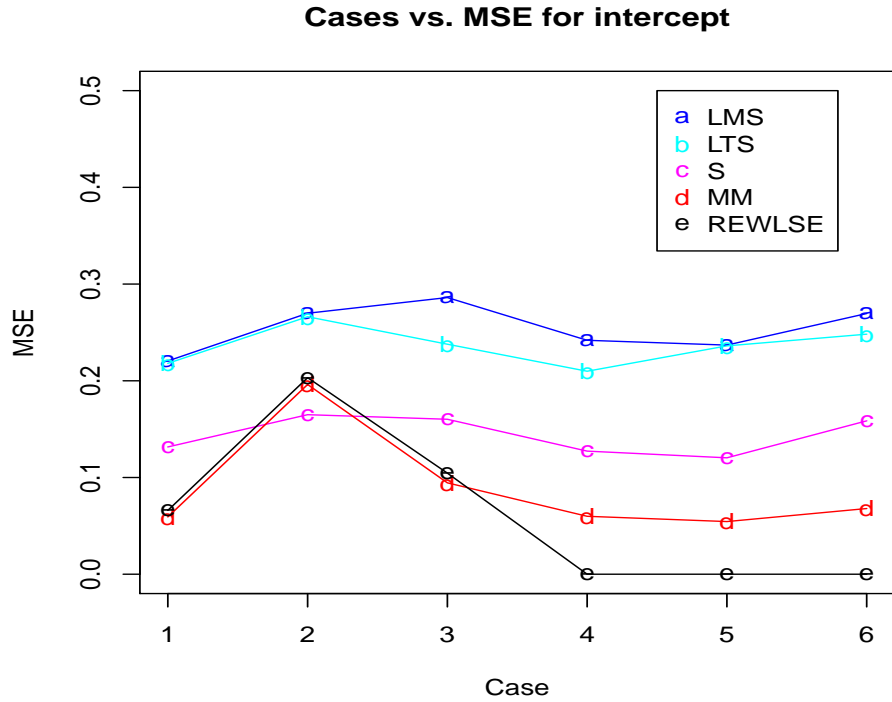


Figure 3.2: Plot of MSE of intercept estimates vs. different cases for LMS, LTS, S, MM, and REWLS, for model 1 when $n = 20$

3.2 Example

Table 3.6 shows a famous data set found in Freedman et al. (1991). The data set contains per capita consumption of cigarettes in various countries in 1930 and the death rates (number of deaths per million people) from lung cancer for 1950. We let death rates per million people be the dependent variable y and let the consumption of cigarette per capita be the independent variable x . The previous study indicates that the data for USA is an outlier. Figure 3.3 is a scatter plot of the data. From the plot, we can see that USA is an outlier with high leverage. Here, let's compare LSE with MM-estimate and REWLS. Figure 3.3 shows the data and the lines fit by these three estimate. The LSE line does not fit the bulk of the data, being a compromise between USA data and the rest, while the fitted lines for the other two estimates almost overlap and give a better representation of the majority of the data.

Table 3.7 also gives the estimated parameters for these three methods with the complete data and with the points of USA deleted. Comparing these two cases, For LSE the intercept estimate changes from 67.56 (whole data set) to 9.14 (without outlier) and the slope estimate changes from 0.23 (whole data set) to

Table 3.2: *MSE of Point Estimates for Model 1 with $n = 20$*

TRUE	LSE	M_H	M_T	LMS	LTS	S	MM	REWLSE
Case I: $\varepsilon \sim N(0,1)$								
$\beta_0 : 0$	0.0497	0.0532	0.0551	0.2485	0.2342	0.1372	0.0564	0.0645
$\beta_1 : 1$	0.0556	0.0597	0.0606	0.2553	0.2328	0.1679	0.0643	0.0733
Case II: $\varepsilon \sim t_1$								
$\beta_0 : 0$	1003.8360	0.2545	0.2146	0.3215	0.2872	0.1447	0.1824	0.1990
$\beta_1 : 1$	1374.0645	0.4103	0.3209	0.3659	0.3496	0.1843	0.2996	0.3164
Case III: $\varepsilon \sim t_3$								
$\beta_0 : 0$	0.1692	0.0884	0.0890	0.3289	0.3076	0.1637	0.0856	0.0982
$\beta_1 : 1$	0.1766	0.1041	0.1027	0.4317	0.3905	0.2041	0.1027	0.1189
Case IV: $\varepsilon \sim N(0,1)$ with outliers in y direction								
$\beta_0 : 0$	9.3051	0.1082	0.0697	0.2752	0.2460	0.1430	0.0671	0.0667
$\beta_1 : 1$	5.5747	0.1083	0.0762	0.2608	0.2029	0.1552	0.0746	0.0801
Case V: $\varepsilon \sim N(0,1)$ with high leverage outliers								
$\beta_0 : 0$	0.8045	0.8711	0.8857	0.2161	0.1984	0.1256	0.0581	0.0598
$\beta_1 : 1$	13.4258	13.7499	13.8487	0.3377	0.3019	0.1695	0.0749	0.0749
Case VI: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,10^2)$								
$\beta_0 : 0$	0.3338	0.0610	0.0528	0.2105	0.2135	0.1228	0.0523	0.0538
$\beta_1 : 1$	0.4304	0.0808	0.0644	0.3149	0.2908	0.1519	0.0636	0.0691

0.37 (without outlier). Thus, it is clear that the outlier strongly influences LSE. For MM-estimate, after deleting the outlier, the intercept estimate change slightly but slope estimate remains almost the same. For REWLSE, both intercept and slope estimates remain unchanged after deleting the outlier. In addition, note that REWLSE for the whole data gives almost the same result as LSE without the outlier.

Table 3.3: *MSE of Point Estimates for Model 1 with $n = 100$*

TRUE	LSE	M_H	M_T	LMS	LTS	S	MM	REWLSE
	Case I: $\varepsilon \sim N(0,1)$							
$\beta_0 : 0$	0.0113	0.0126	0.0125	0.0755	0.0767	0.0347	0.0125	0.0131
$\beta_1 : 1$	0.0096	0.0102	0.0103	0.0693	0.0705	0.0312	0.0103	0.0112
	Case II: $\varepsilon \sim t_1$							
$\beta_0 : 0$	40.8454	0.0416	0.0310	0.0550	0.0392	0.0201	0.0323	0.0354
$\beta_1 : 1$	39.5950	0.0469	0.0387	0.0607	0.0476	0.0274	0.0402	0.0447
	Case III: $\varepsilon \sim t_3$							
$\beta_0 : 0$	0.0283	0.0154	0.0153	0.0596	0.0659	0.0231	0.0153	0.0170
$\beta_1 : 1$	0.0255	0.0157	0.0164	0.0652	0.0752	0.0356	0.0163	0.0185
	Case IV: $\varepsilon \sim N(0,1)$ with outliers in y direction							
$\beta_0 : 0$	8.9470	0.0465	0.0107	0.0674	0.0658	0.0283	0.0106	0.0108
$\beta_1 : 1$	0.7643	0.0146	0.0120	0.0611	0.0704	0.0338	0.0119	0.0120
	Case V: $\varepsilon \sim N(0,1)$ with high leverage outliers							
$\beta_0 : 0$	0.2840	0.2999	0.2983	0.0575	0.0595	0.0234	0.0107	0.0106
$\beta_1 : 1$	13.2298	13.5907	13.7210	0.0624	0.0790	0.0310	0.0127	0.0131
	Case VI: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,10^2)$							
$\beta_0 : 0$	0.0650	0.0119	0.0107	0.0732	0.0737	0.0296	0.0107	0.0110
$\beta_1 : 1$	0.0596	0.0126	0.0123	0.0696	0.0775	0.0353	0.0122	0.0134

Table 3.4: *MSE of Point Estimates for Model 2 with $n = 20$*

TRUE	LSE	M_H	M_T	LMS	LTS	S	MM	REWLSE
Case I: $\varepsilon \sim N(0,1)$								
$\beta_0 : 0$	0.0610	0.0659	0.0744	0.3472	0.2424	0.1738	0.0679	0.0800
$\beta_1 : 1$	0.0588	0.0664	0.0752	0.4066	0.3247	0.2299	0.0709	0.1051
$\beta_2 : 1$	0.0620	0.0653	0.0725	0.3557	0.2724	0.2018	0.0716	0.0880
$\beta_3 : 1$	0.0698	0.0719	0.0758	0.3444	0.2657	0.1904	0.0751	0.0999
Case II: $\varepsilon \sim t_1$								
$\beta_0 : 0$	248.0170	0.3492	0.2579	0.7935	0.4657	0.3615	0.2630	0.2957
$\beta_1 : 1$	209.8339	0.4503	0.3713	1.2482	0.9701	0.4355	0.3784	0.4443
$\beta_2 : 1$	93.1344	0.4089	0.2936	1.0517	0.6203	0.5086	0.2965	0.3365
$\beta_3 : 1$	374.7307	0.4387	0.3206	1.0829	0.7704	0.4717	0.3123	0.4023
Case III: $\varepsilon \sim t_3$								
$\beta_0 : 0$	0.1745	0.1125	0.1168	0.3799	0.3040	0.2326	0.1177	0.1210
$\beta_1 : 1$	0.1998	0.1332	0.1364	0.4402	0.3404	0.2539	0.1311	0.1485
$\beta_2 : 1$	0.1704	0.1203	0.1272	0.4868	0.3831	0.2118	0.1242	0.1461
$\beta_3 : 1$	0.2018	0.1520	0.1732	0.5687	0.4964	0.3145	0.1649	0.2049
Case IV: $\varepsilon \sim N(0,1)$ with outliers in y direction								
$\beta_0 : 0$	9.9455	0.1442	0.0706	0.3127	0.2334	0.1759	0.0680	0.0713
$\beta_1 : 1$	5.1353	0.1015	0.0636	0.3638	0.2769	0.1508	0.0617	0.0654
$\beta_2 : 1$	5.1578	0.1245	0.0730	0.4647	0.2796	0.1759	0.0690	0.0722
$\beta_3 : 1$	6.0662	0.1273	0.0612	0.3922	0.2733	0.1797	0.0597	0.0654
Case V: $\varepsilon \sim N(0,1)$ with high leverage outliers								
$\beta_0 : 0$	1.0096	1.0733	1.1334	0.3339	0.2491	0.1716	0.0821	0.0840
$\beta_1 : 1$	13.6630	14.0715	14.1688	0.4698	0.3126	0.2500	0.1467	0.1031
$\beta_2 : 1$	0.9201	0.9684	1.0108	0.4088	0.2681	0.2064	0.0899	0.1088
$\beta_3 : 1$	0.8538	0.9316	0.9937	0.4411	0.3373	0.2077	0.0709	0.0957
Case VI: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,10^2)$								
$\beta_0 : 0$	0.3245	0.0853	0.0837	0.2820	0.2433	0.1873	0.0785	0.0924
$\beta_1 : 1$	0.3391	0.1026	0.1001	0.4609	0.2875	0.2328	0.0996	0.1047
$\beta_2 : 1$	0.3039	0.0898	0.0938	0.4077	0.3053	0.1887	0.0900	0.1170
$\beta_3 : 1$	0.2618	0.0846	0.0941	0.4560	0.3023	0.2054	0.0900	0.1007

Table 3.5: *MSE of Point Estimates for Model 2 with $n = 100$*

TRUE	LSE	M_H	M_T	LMS	LTS	S	MM	REWLSE
Case I: $\varepsilon \sim N(0,1)$								
$\beta_0 : 0$	0.0097	0.0108	0.0109	0.0743	0.0690	0.0359	0.0108	0.0119
$\beta_1 : 1$	0.0111	0.0120	0.0121	0.0736	0.0778	0.0399	0.0119	0.0130
$\beta_2 : 1$	0.0100	0.0106	0.0107	0.0713	0.0715	0.0404	0.0107	0.0114
$\beta_3 : 1$	0.0110	0.0116	0.0118	0.0662	0.0712	0.0388	0.0118	0.0121
Case II: $\varepsilon \sim t_1$								
$\beta_0 : 0$	36.7303	0.0388	0.0287	0.0681	0.0590	0.0317	0.0289	0.0326
$\beta_1 : 1$	31.6433	0.0499	0.0351	0.0624	0.0618	0.0262	0.0367	0.0372
$\beta_2 : 1$	41.4547	0.0422	0.0337	0.0788	0.0613	0.0321	0.0344	0.0369
$\beta_3 : 1$	29.7017	0.0476	0.0317	0.0714	0.0506	0.0320	0.0332	0.0362
Case III: $\varepsilon \sim t_3$								
$\beta_0 : 0$	0.0294	0.0145	0.0159	0.0713	0.0655	0.0330	0.0158	0.0179
$\beta_1 : 1$	0.0464	0.0198	0.0180	0.0651	0.0674	0.0368	0.0181	0.0195
$\beta_2 : 1$	0.0375	0.0183	0.0181	0.0727	0.0733	0.0352	0.0181	0.0195
$\beta_3 : 1$	0.0365	0.0176	0.0167	0.0646	0.0736	0.0344	0.0167	0.0175
Case IV: $\varepsilon \sim N(0,1)$ with outliers in y direction								
$\beta_0 : 0$	9.1058	0.0560	0.0118	0.0631	0.0579	0.0322	0.0118	0.0120
$\beta_1 : 1$	0.8544	0.0186	0.0137	0.0738	0.0814	0.0377	0.0136	0.0143
$\beta_2 : 1$	0.9538	0.0189	0.0141	0.0672	0.0717	0.0379	0.0140	0.0146
$\beta_3 : 1$	0.8953	0.0193	0.0121	0.0652	0.0696	0.0363	0.0120	0.0123
Case V: $\varepsilon \sim N(0,1)$ with high leverage outliers								
$\beta_0 : 0$	0.2673	0.2869	0.2901	0.0632	0.0596	0.0300	0.0114	0.0114
$\beta_1 : 1$	13.2587	13.6355	13.6754	0.0590	0.0658	0.0305	0.0123	0.0127
$\beta_2 : 1$	0.1817	0.1889	0.1922	0.0660	0.0727	0.0344	0.0139	0.0144
$\beta_3 : 1$	0.1546	0.1607	0.1643	0.0668	0.0710	0.0344	0.0107	0.0108
Case VI: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,10^2)$								
$\beta_0 : 0$	0.0591	0.0109	0.0100	0.0656	0.0625	0.0281	0.0100	0.0109
$\beta_1 : 1$	0.0492	0.0122	0.0112	0.0558	0.0643	0.0349	0.0110	0.0115
$\beta_2 : 1$	0.0640	0.0123	0.0110	0.0635	0.0683	0.0337	0.0109	0.0118
$\beta_3 : 1$	0.0696	0.0135	0.0122	0.0573	0.0608	0.0333	0.0122	0.0128

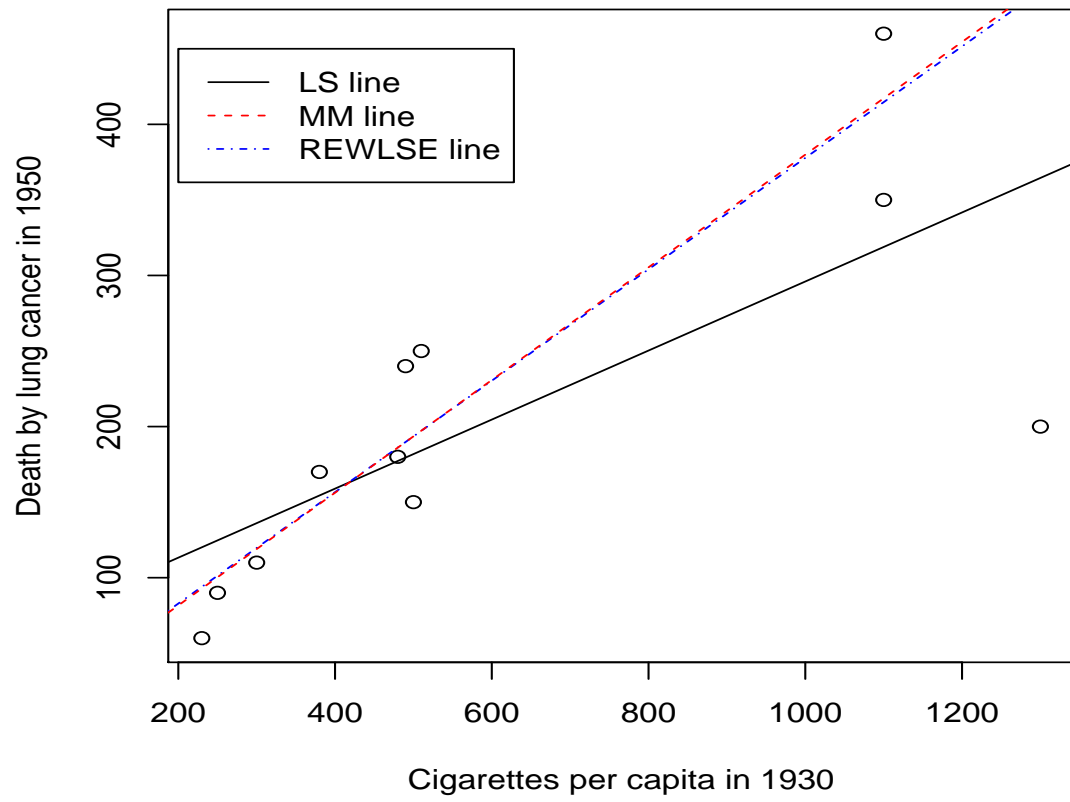


Figure 3.3: *Fit lines for Cigarettes data*

Table 3.6: *Cigarettes data*

Country	Cigarette per capita	Deaths p. mill.
Australia	480	180
Canada	500	150
Denmark	380	170
Finland	1100	350
GreatBritain	1100	460
Iceland	230	060
Netherlands	490	240
Norway	250	090
Sweden	300	110
Switzerland	510	250
USA	1300	200

Table 3.7: *Regression estimates for Cigarettes data*

estimators	Intercept	Intercept (-USA)	Slope	Slope (-USA)
LS	67.5609	9.1393	0.2284	0.3687
MM	7.0639	5.9414	0.3729	0.3753
REWLSE	9.1393	9.1393	0.3686	0.3686

Bibliography

- [1] Bickel, P. J. (1975), One-step Huber Estimates in the Linear Model. *Journal of American Statistical Association*, 70, 428-434.
- [2] Carroll, R. J., and Welsh, A. H. (1988), A Note on Asymmetry and Robustness in Linear Regression. *The American Statistician*, 42, 285-287.
- [3] Donoho, D. L. and Huber, P. J. (1983), The Notation of Break-down Point, *in A Festschrift for E. L. Lehmann, Wadsworth*
- [4] Freedman, W. L., Wilson, C. D., and Madore, B. F. (1991), *ApJ*, 372, 455
- [5] Gervini, D. and Yohai, V. J. (2002), A Class of Robust and Fully Efficient Regression Estimators. *The Annals of Statistics*, 30, 583-616.
- [6] Giltinan, D. M., Carroll, R. J., and Ruppert, D. (1986), Some New Estimation Methods for Weighted Regression When There Are Possible Outliers. *Technometrics*, 28, 219-230.
- [7] Hampel, F. R. (1974), The Influence Curve and Its Role in Robust Estimation. *The Annals of Statistics*, 69, 383-393.
- [8] Hampel, F. R. (1975), Beyond Location Parameters: Robust Concepts and Methods. *Bernoulli (International Statistical Institute)*, 46, 375-382.
- [9] Hampel, F. R. (1978), Optimally Bounding the Gross-Error-Sensitivity and the Influence of Position in Factor Space. *American Statistical Association*, pp. 59-64.
- [10] Handschin, E., Kohlas, J., Fiechter, A., and Schweppe, F. (1975), Bad Data Analysis for Power System State Estimation. *IEEE Transactions on Power Apparatus and Systems*, 2, 329-337.
- [11] He, X., and Portnoy, S. (1992), Reweighted LS Estimators Converge at the Same Rate as the Initial Estimator. *The Annals of Statistics*, 20, 2161-2167.
- [12] Hossjer, O. (1992). On the Optimality of S-estimators. *Statistics and Probability Letters*, 14, 413-419.
- [13] Huber, P. J. (1973), Robust Regression: Asymptotics, Conjectures and Monte Carlo. *Annals of Mathematical Statistics*, 1, 799-821.

- [14] Huber, P. J. (1981), Robust Statistics. *Wiley, New York*.
- [15] Jureckova, J., and Portnoy, S. (1987), Asymptotics for One-Step M-Estimators in Regression With Application to Combining Efficiency and High Breakdown Point. *Communications in Statistics, Theory, and Methods*, 16(8), 2187-2199.
- [16] Krasker, W. S. (1980), Estimation in Linear Regression Models With Disparate Data Points. *Econometrica*, 48, 1333-1346.
- [17] Krasker, W. S., and Welsch, R. E. (1982), Efficient Bounded-Influence Regression Estimation. *Journal of American Statistical Association*, 77, 595-604.
- [18] Mallows, C. L. (1975), On Some Topics in Robustness. unpublished memorandum, Bell Tel. Laboratories, Murray Hill.
- [19] Maronna, R. A., Bustos, O. H., and Yohai, V.J. (1979), Bias- and Efficiency-Robustness of General M-Estimators for Regression With Random Carriers. *Smoothing Techniques for Curve Estimation*, eds. T. Gasser and M. Rosenblatt, New York: Springer-Verlag, pp. 91-116.
- [20] Maronna, R. A., Martin, R. D. and Yohai, V. J. (2006), Robust Statistics. *John Wiley*.
- [21] Rousseeuw, P. J.(1983), Multivariate Estimation with High Breakdown Point. Resaerch Report No. 192, Center for Statistics and Operations research, VUB Brussels.
- [22] Rousseeuw, P. J. and Leroy, A. (1987), Robust Regression and Outlier Detection. *Wiley, New York*.
- [23] Rousseeuw, P. J. and Yohai, V. J. (1984), Robust Regression by Means of S-estimators. *Robust and Nonlinear Time series*, J. Franke, W. Härdle and R. D. Martin (eds.),Lectures Notes in Statistics 26, 256-272, New York: Springer.
- [24] Siegel, A. F. (1982), Robust Regression Using Repeated Medians. *Biometrika*, 69, 242-244.
- [25] Simpson, D. G., Ruppert, D., and Carroll, R.J. (1992), On One-step GM Estimates and Stability of Inferences in Linear Regression. *Journal of American Statistical Association*, 87, 439-450.
- [26] Stigler, S. M. (1981), Gauss and the invention of least squares. *Annals of Statistics*, 9, 465-474.
- [27] Yohai, V. J. (1987), High Breakdown-point and High Efficiency Robust Estimates for Regression. *The Annals of Statistics*, 15, 642-656.
- [28] Yohai, V. J. and Zamar, R. H. (1988), High Breakdown-point Estimates of Regression by Means of the Minimization of an Efficient Scale. *Journal of American Statistical Association*, 83, 406-413.

Appendix A

R Code

```
#####  
compare methods among M estimators  
#####  
par(mfrow=c(2,3))  
e=seq(-6,6,0.01)  
# LSE  
rho=e^2/2  
psi=e  
w=e/e  
plot(e,rho,type="l")  
plot(e,psi,type="l")  
plot(e,w,type="l")  
#LAD  
rho=abs(e)  
psi=sign(e)  
w=1/rho  
plot(e,rho,type="l")  
plot(e,psi,type="l")  
plot(e,w,type="l")  
# Huber  
k1=1.345  
rho=(abs(e)<=k1)*e^2/2+(abs(e)>k1)*(k1*abs(e)-k1^2/2)
```

```

plot(e,rho,type="l")
psi=(abs(e)<=k1)*e+(abs(e)>k1)*sign(e)*k1
plot(e,psi,type="l")
w=(abs(e)<=k1)+(abs(e)>k1)*k1/abs(e)
plot(e,w,type="l")
#Bisquare
k2=4.685
a=1-(e/k2)^2
b=1-a^3
c=k2^2/6
rho=(abs(e)<=k2)*b*c+(abs(e)>k2)*k2^2/6
plot(e,rho,type="l")
psi=(abs(e)<=k2)*e*a^2+(abs(e)>k2)*0
w=(abs(e)<=k2)*a^2+(abs(e)>k2)*0
plot(e,psi,type="l")
plot(e,w,type="l")

```

```

#####
the effect of leverage point on LSE and M-estimate
#####

```

```

library(MASS)
library(robust)
n=20
p=2
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rnorm(n),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
z[1:2]=c(5,6)
y[1:2]=c(40,45)
ls<-lm(y~z)

```

```

m<-rlm(y~z)
plot(z,y,xlab="x",ylab="y")
#plot(z,y,type="n",cex=.5)
#points(z,y,cex=.5,pch=20)
abline(ls,lty = 1)
abline(m,lty = 2)

#####
##compare various method with model y=x+e,n=20
#####

library(MASS)
library(robust)
n=20
p=2
mycontrol=lmRob.control(weight=c("bisquare","optimal"))
col.1=rep(0,200,nrow=200)
col.2=rep(1,200,nrow=200)
beta.true=cbind(col.1,col.2)
beta.orig=matrix(rep(0,400),nrow=200)
beta.ls=beta.orig
beta.m1=beta.orig
beta.m2=beta.orig
beta.lms=beta.orig
beta.lts=beta.orig
beta.s=beta.orig
beta.mm=beta.orig
beta.rewlse=beta.orig
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
#e=matrix(rnorm(n),nrow=n) # case1 normal errors
#e=matrix(rt(n,df=1),nrow=n) # case2 Cauchy distribution (t distribution with df=1)
#e=matrix(rt(n,df=3),nrow=n) # case3 t distribution with df=3

```

```

#y[1:2]=c(30,30) # case4 outlier contamination in y direction
#z[1:2]=c(10,10) # case5 outlier contamination in x and y directions
#y[1:2]=c(50,50) # case5 outlier contamination in x and y directions
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
rbind(mse.ls,mse.m1,mse.m2,mse.lms,mse.lts,mse.s,mse.mm,mse.rewlse)

#####

```



```

## case6 outliers contaminated by 95% standard normal and 5% normal with sd=10
#####
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
e0=cbind(rnorm(n), rnorm(n,mean=0,sd=10))
p1=0.95
p2=0.05
ix=sample(c(1,2),n,prob=c(p1,p2),replace=TRUE)
e=matrix(e0[,1]*(ix==1)+e0[,2]*(ix==2),nrow=n)
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)

```

```

mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
rbind(mse.ls,mse.m1,mse.m2,mse.lms,mse.lts,mse.s,mse.mm,mse.rewlse)

```

```

#####
##compare various method with model y=x1+x2+x3+e,n=20
#####
library(MASS)
library(robust)
##compare various method with model y=x1+x2+x3+e,n=20
n=20
p=4
mycontrol=lmRob.control(weight=c("bisquare","optimal"))
col.1=rep(0,200,nrow=200)
col.2=rep(1,200,nrow=200)
col.3=rep(1,200,nrow=200)
col.4=rep(1,200,nrow=200)
beta.true=cbind(col.1,col.2,col.3,col.4)
beta.orig=matrix(rep(0,800),nrow=200)
beta.ls=beta.orig
beta.m1=beta.orig
beta.m2=beta.orig
beta.lms=beta.orig
beta.lts=beta.orig
beta.s=beta.orig
beta.mm=beta.orig
beta.rewlse=beta.orig
for (i in 1:200){
  z=matrix(rnorm((p-1)*n),nrow=n)
  col1=rep(1,n,nrow=n)
  x=cbind(col1,z)

```

```

e=matrix(rnorm(n),nrow=n) # case1 normal errors
#e=matrix(rt(n,df=1),nrow=n) # case2 Cauchy distribution (t distribution with df=1)
#e=matrix(rt(n,df=3),nrow=n) # case3 t distribution with df=3
#y[1:2]=c(30,30) # case4 outlier contamination in y direction
#z[1:2]=c(10,10) # case5 outlier contamination in x and y directions
#y[1:2]=c(50,50) # case5 outlier contamination in x and y directions
a=c(0,1,1,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)

```

```
rbind(mse.ls,mse.m1,mse.m2,mse.lms,mse.lts,mse.s,mse.mm,mse.rewlse)
```

```
#####  
## case6 outliers contaminated by 95% standard normal and 5% normal with sd=10  
#####  
for (i in 1:200){  
  z=matrix(rnorm((p-1)*n),nrow=n)  
  col1=rep(1,n,nrow=n)  
  x=cbind(col1,z)  
  a=c(0,1,1,1)  
  beta=matrix(a,nrow=p)  
  y=x%*%beta+e  
  e0=cbind(rnorm(n), rnorm(n,mean=0,sd=10))  
  p1=0.95  
  p2=0.05  
  ix=sample(c(1,2),n,prob=c(p1,p2),replace=TRUE)  
  e=matrix(e0[,1]*(ix==1)+e0[,2]*(ix==2),nrow=n)  
  ls<-lm(y~z)  
  m1<-rlm(y~z)#psi=Huber  
  m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey  
  lms<-lmsreg(y~z)  
  lts<-ltsreg(y~z)  
  s<-lqs(y~z,method="S")  
  mm<-rlm(y~z,method="MM")  
  rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)  
  beta.ls[i,]=c(ls$coef)  
  beta.m1[i,]=c(m1$coef)  
  beta.m2[i,]=c(m2$coef)  
  beta.lms[i,]=c(lms$coef)  
  beta.lts[i,]=c(lts$coef)  
  beta.s[i,]=c(s$coef)  
  beta.mm[i,]=c(mm$coef)  
  beta.rewlse[i,]=c(rewlse$coef)
```

```

}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
rbind(mse.ls,mse.m1,mse.m2,mse.lms,mse.lts,mse.s,mse.mm,mse.rewlse)

```

```

#####
compare mse vs. cases
#####
library(MASS)
library(robust)
n=20
p=2
beta=matrix(rep(0,12),nrow=6)
B.ls=beta
B.m1=beta
B.m2=beta
B.lms=beta
B.lts=beta
B.s=beta
B.mm=beta
B.rewlse=beta
mycontrol=lmRob.control(weight=c("bisquare","optimal"))
col.1=rep(0,200,nrow=200)
col.2=rep(1,200,nrow=200)
beta.true=cbind(col.1,col.2)
beta.orig=matrix(rep(0,400),nrow=200)
beta.ls=beta.orig
beta.m1=beta.orig

```

```

beta.m2=beta.orig
beta.lms=beta.orig
beta.lts=beta.orig
beta.s=beta.orig
beta.mm=beta.orig
beta.rewlse=beta.orig
## case1 normal errors
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rnorm(n),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)

```

```

mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=1
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse
## case2 Cauchy distribution (t distribution with df=1)
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rt(n,df=1),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)

```

```

beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=2
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse
## case3 t distribution with df=3
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rt(n,df=3),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
ls<-lm(y~z)

```



```

m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=3
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse
## case4 outlier contamination in y direction

```

```

for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rnorm(n),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
#z[1:2]=c(0,0)
y[1:2]=c(30,30)
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)

```

```

mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=4
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse
## case5 outlier contamination in x and y directions
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
e=matrix(rnorm(n),nrow=n)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
z[1:2]=c(10,10)
y[1:2]=c(50,50)
ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)

```

```

beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=5
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse
## case6 outliers contaminated by 95% standard normal and 5% normal with sd=10
for (i in 1:200){
z=matrix(rnorm((p-1)*n),nrow=n)
col1=rep(1,n,nrow=n)
x=cbind(col1,z)
a=c(0,1)
beta=matrix(a,nrow=p)
y=x%*%beta+e
e0=cbind(rnorm(n), rnorm(n,mean=0,sd=10))
p1=0.95
p2=0.05
ix=sample(c(1,2),n,prob=c(p1,p2),replace=TRUE)
e=matrix(e0[,1]*(ix==1)+e0[,2]*(ix==2),nrow=n)

```

```

ls<-lm(y~z)
m1<-rlm(y~z)#psi=Huber
m2<-rlm(y~z,psi=psi.bisquare) #psi=Tukey
lms<-lmsreg(y~z)
lts<-ltsreg(y~z)
s<-lqs(y~z,method="S")
mm<-rlm(y~z,method="MM")
rewlse<-lmRob(y~z,control=mycontrol,final.alg=adaptive)
beta.ls[i,]=c(ls$coef)
beta.m1[i,]=c(m1$coef)
beta.m2[i,]=c(m2$coef)
beta.lms[i,]=c(lms$coef)
beta.lts[i,]=c(lts$coef)
beta.s[i,]=c(s$coef)
beta.mm[i,]=c(mm$coef)
beta.rewlse[i,]=c(rewlse$coef)
}
mse.ls=apply((beta.ls-beta.true)^2,2,mean)
mse.m1=apply((beta.m1-beta.true)^2,2,mean)
mse.m2=apply((beta.m2-beta.true)^2,2,mean)
mse.lms=apply((beta.lms-beta.true)^2,2,mean)
mse.lts=apply((beta.lts-beta.true)^2,2,mean)
mse.s=apply((beta.s-beta.true)^2,2,mean)
mse.mm=apply((beta.mm-beta.true)^2,2,mean)
mse.rewlse=apply((beta.rewlse-beta.true)^2,2,mean)
j=6
B.ls[j,]=mse.ls
B.m1[j,]=mse.m1
B.m2[j,]=mse.m2
B.lms[j,]=mse.lms
B.lts[j,]=mse.lts
B.s[j,]=mse.s
B.mm[j,]=mse.mm
B.rewlse[j,]=mse.rewlse

```

```

i=c(1,2,3,4,5,6)
#MSE for slopes
t=seq(0,0.5,0.1)
y=10*t+1
plot(y,t,xlab="Case",ylab="MSE",main="Cases vs. MSE for slope",type="n")
points(i,B.lms[i,2],col=4,type="o",pch="a")
points(i,B.lts[i,2],col=5,type="o",pch="b")
points(i,B.s[i,2],col=6,type="o",pch="c")
points(i,B.mm[i,2],col=10,type="o",pch="d")
points(i,B.rewlse[i,2],col=9,type="o",pch="e")
legend(4.5,0.5,c("LMS","LTS","S","MM","REWLSE"),col=c(4,5,6,10,9),pch=c("a","b","c","d","e"))
#MSE for intercepts
t=seq(0,0.5,0.1)
y=10*t+1
plot(y,t,xlab="Case",ylab="MSE",main="Cases vs. MSE for intercept",type="n")
points(i,B.lms[i,1],col=4,type="o",pch="a")
points(i,B.lts[i,1],col=5,type="o",pch="b")
points(i,B.s[i,1],col=6,type="o",pch="c")
points(i,B.mm[i,1],col=10,type="o",pch="d")
points(i,B.rewlse[i,1],col=9,type="o",pch="e")
legend(4.5,0.5,c("LMS","LTS","S","MM","REWLSE"),col=c(4,5,6,10,9),pch=c("a","b","c","d","e"))

#####
Effect of one outlier on LSE
#####
da=read.table("C:\\Users\\Xue Bai\\Desktop\\simulation\\example\\d4.txt",header=T,row.names=1)
B.ls=matrix(rep(0,22),nrow=11)
Y=da[,3]
X=da[,2]
ls0<-lm(Y~X)
for (i in 1:11){
Y[11]=220-20*i
ls<-lm(Y~X)
B.ls[i,]=abs(c(ls$coef)-c(ls0$coef))
}

```

```

}
i=c(1,2,3,4,5,6,7,8,9,10,11)
par(mfrow=c(1,2))
plot(20*i,B.ls[i,2],xlab="dist. from 200 to new number of deaths", ylab="dist. from original LSE to new
plot(20*i,B.ls[i,1],xlab="dist. from 200 to new number of deaths", ylab="dist. from original LSE to new

#####
Fit lines for Cigarettes data
#####
da=read.table("C:\\Users\\Xue Bai\\Desktop\\simulation\\example\\d4.txt",header=T)
attach(da)
library(MASS)
library(robust)
mycontrol=lmRob.control(weight=c("bisquare","optimal"))
ls<-lm(Deaths~Cigarette)
mm<-rlm(Deaths~Cigarette,method="MM")
rewlse<-lmRob(Deaths~Cigarette,control=mycontrol,final.alg=adaptive)
plot(Deaths~Cigarette,ylab="Death by lung cancer in 1950",xlab="Cigarettes per capita in 1930")
abline(ls,col = 1)
abline(mm,col =2,lty=2)
abline(rewlse,col=4,lty=3)
legend(200,450,c("LS line","MM line","REWLSE line"),col=c(1,2,4),lty=c(1,2,3))

```