ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

Gülsen KIRAL

A COMPARISON OF THE RECENT ALGORITHMS FOR THE IDENTIFICATION OF OUTLIERS IN DATA

MATEMATİK ANABİLİM DALI

ADANA, 2003

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

A COMPARISON OF THE RECENT ALGORITHMS FOR THE IDENTIFICATION OF OUTLIERS IN DATA

GÜLSEN KIRAL

DOKTORA TEZİ

MATEMATİK ANABİLİM DALI

Bu tez 20../03/.2003 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından Oybirliği/~~Oy Çokluğu~~ İle Kabul Edilmiştir.

İmza:.....	İmza:.....	İmza:.....
Doç. Dr. Nedret BİLLOR	Prof. Dr. Fikri AKDENİZ	Prof. Dr. H. Altan ÇABUK
DANIŞMAN	ÜYE	ÜYE
İmza:.....	İmza:.....	
Prof. Dr. Müjgan TEZ	Prof. Dr. Refik BURGUT	
ÜYE	ÜYE	

Bu tez Enstitümüz Matematik Anabilim Dalında hazırlanmıştır.

Kod No: 728

Prof. Dr. Fikri AKDENİZ
Enstitü Müdürü
İmza ve Mühür

Bu çalışma, Çukurova Üniversitesi Rektörlüğü Araştırma Fonu Tarafından Desteklenmştir.
Proje No: FBE-2000.D.53

CONTENTS

ABSTRACT

Ph.D. THESIS

A COMPARISON OF THE RECENT ALGORITHMS FOR THE IDENTIFICATION OF OUTLIERS IN DATA

GÜLSEN KIRAL

DEPARTMENT OF MATHEMATICS
INSTITUTE OF NATURAL AND APPLIED SCIENCES
ÇUKUROVA UNIVERSITY

Supervisor: Assoc. Prof. Dr. Nedret BİLLOR
Year: 2003, Pages: 155
Jury: Assoc. Prof. Dr. Nedret BİLLOR
Prof. Dr. Fikri AKDENİZ
Prof. Dr. H. Altan ÇABUK
Prof. Dr. Müjgan TEZ
Prof. Dr. Refik BURGUT

In this thesis we examine outlier detection methods in multivariate and regression data and present an extensive literature on them. Since there has been a fast growing interest in the outlier detection methods for regression data for ten years we mainly focus on the outlier detection methods in regression data in this thesis. We conduct an extensive simulation study to assess the performances of the multiple outlier detection methods for regression data, that are either most recently published or most frequently cited in the literature. Furthermore in the context of multivariate data we propose a new outlier detection algorithm in principal component analysis called *BACON robust principle component analysis*. We also carry out a simulation study to assess the performance of the proposed method and use some data sets to evaluate the applicability of the method.

Key Words: influential observations, detection of outliers, multivariate data, principle component analysis, regression.

ÖZ

DOKTORA TEZİ

VERİLERDE SAPAN DEĞERLERİN SAPTANMASINA İLİŞKİN YENİ GELİŞTİRİLMİŞ ALGORİTMALARIN KARŞILAŞTIRILMASI

GÜLSEN KIRAL

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

Danışman: Doç. Dr. Nedret BİLLOR
Yıl: 2003, Sayfa: 155
Jüri: Doç. Dr. Nedret BİLLOR
Prof. Dr. Fikri AKDENİZ
Prof. Dr. H. Altan ÇABUK
Prof. Dr. Müjgan TEZ
Prof. Dr. Refik BURGUT

Bu tezde, çok değiskenli ve regresyon tipi veri kümelerinde sapan değerlerin belirlenmesine yönelik yöntemler geniş bir şekilde incelenmiştir. Son on yıldır regresyon tipi veri kümelerinde sapan değerlerin belirlenmesine yönelik yöntemler üzerinde çok yoğun çalışmalar olduğundan bu çalışmada, sadece regresyon tipi veri kümeleri için son yıllarda önerilen veya literatürde yaygın olarak kullanılan teknikler üzerine yoğunlaşılmıştır. Bu yöntemlerin karşılaştırılması amacıyla detaylı bir simülasyon çalışması yapılmış ve yöntemlerin performansları saptanmıştır. Ayrıca çok değişkenli veri kümeleri analizi icin yeni bir *dayanıklı BACON temel bileşenler analiz* algoritması önerilerek; bu yöntemin performansı hem veri kumeleri hem de simülasyon çalışması üzerinde gösterilmiştir.

Anahtar Kelimeler: etkili gözlem, sapan değerlerin saptanması, çok değişkenli veri, temel bileşenler analizi.

ACKNOWLEDGEMENTS

First of all; I would like to thank to my committee chairs Assoc. Prof. Dr. Nedret Billor, Prof. Dr. Fikri Akdeniz and Prof. Dr. Altan Çabuk for serving on my committee. The quality of my research was greatly enhanced by their insight and guidance.

Additionally; I would also like to thank James Winsnowski and Mehmet Balcılar for their support during my simulation study.

I would like to thank the Billors family for their help, interests, encouragement, suggestions, remarks, patience, money support, hospitality and time over the period of my study in USA.

I would also like to thank my husband, my daughter, my mother-in-law and also my parents for their help, patience and support during my study.

Finally, I thank all the faculty, postgraduate students and staff in the Department of Mathematics in Çukurova University and the Department of Statistics and Actuarial Science in Iowa University for creating a pleasant working environment and help.

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

In order to extract valuable and accurate information from the available data in hand one has to assume that the data are homogeneous. However it is well known fact that in real life data contain outliers or subgroups which means that data are not homogeneous.

Outliers are minority of observations that do not conform to the pattern (model) suggested by the homogeneous majority of the observations. All classical statistical methods such as regression based methods, principal components analysis can be badly affected by outliers.

If the data are not homogeneous then the conclusions extracted from the data are likely to be incorrect (Barnett and Lewis, 1994).

There are two approaches concerning outlier problem:

1. Robust estimation methods, and
2. Outlier detection methods.

Robust methods are constructed specifically to be resistant to outliers. This is done by down weighting suspected or potential outliers. Robust methods are generally very effective for the detection of outliers, but unfortunately are computationally expensive and not useful in practice.

Outlier detection methods are proposed to detect outliers and to leave the decision to the analyst what to do with outliers. For example data analyst can continue the analysis on the rest of the observations after omitting outliers. In robust language this is equivalent to give zero weight to outliers and full weight to the rest of the observations.

The problem in the methods for the detection of outliers is that these methods are based on the classical estimators (e.g. arithmetic mean and covariance matrix for multivariate data, ordinary least square estimator for regression data) which are severely affected by outliers. Therefore they are not efficient for detecting outliers.

In the last decade, numerous studies have been intensified on a new approach which is a combination of the two approaches given above. This new approach is

easy to compute, and at the same time aims at finding reliable methods for detecting outliers.

In the literature many methods have been proposed for identifying multiple outliers in data. The performance of these methods has not been investigated in details. There is a little work done on the detailed investigations and comparisons of the recent methods concerning outlier detection and obtaining resistant (robust) estimators in multivariate and regression data (Hadi et al., 1996, Kianifard and Swallow, 1990, Winsnowski et al., 2001). For last four years the studies on new methods which aim at eliminating the problems given above increased noticeably.

In this thesis I will examine the recent outlier detection methods in multivariate and regression data sets. Since there has been a fast growing interest in the outlier detection methods for regression data for ten years I mainly focus on the outlier detection methods in regression data in this study. I investigate the performances of the outlier detection methods in regression data that are either most recently published or most frequently referred in the literature by constructing an extensive simulation study.

Chapter 2 introduces the methodology, which I come across in this thesis. Chapter 3 contains previously proposed multiple outlier detection methods with details in multivariate and regression data. A new proposed method, BACON robust principle component analysis, which is used for identifying multiple outliers in multivariate data, is given in Chapter 4. In this chapter I also display the performance of this method on real and simulated data sets. Chapter 5 consists of an application of the outlier detection methods proposed for regression data to the well-known data sets in the statistical literature. An extensive simulation study to assess the performances of the outlier detection procedures in regression data is given in Chapter 6.

2. METHODOLOGY

In this chapter I will give a detailed methodology for regression data which I will be using the most in this thesis.

2.1. Linear Regression

Regression analysis has become one of the most widely used statistical tools for analyzing data. It is appealing because it provides a conceptually simple method for investigating functional relationships among variables. The standard approach in regression analysis is to take data, fit a model, and then evaluate the fit using statistics such as *t, F,* and *R-squared.*

Applications of regression analysis exist in almost every field. In economics, the response variable might be a family's consumption expenditure and the explanatory variables might be the family's income, number of children in the family, and other factors that would affect the family's consumption patterns. In political science, the response variable might be a state's level of welfare spending and the explanatory variables measures of public opinion and institutional variables that would cause the state to have higher or lower levels of welfare spending. In sociology, the response variable might be a measure of the social status of various occupations and the explanatory variables characteristics of the occupations (pay, qualifications, etc.). In psychology, the response variable might be individual's racial tolerance as measured on a standard scale and with indicators of social background as explanatory variables. In education, the response variable might be a student's score on an achievement test and the explanatory variables characteristics of the student's family, teachers, or school.

The relationship between response and explanatory variables is explained by a linear regression model. In a linear regression model, the response variable is assumed to be a linear function of one or more explanatory variables plus an error introduced to account for all other factors:

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \varepsilon_i \qquad \text{for i} = 1, 2, ..., \text{n}. \tag{2.1}$$

In the above regression equation, y_i is the response variable, x_{i1}, ..., x_{ik} are the explanatory variables, and ε_i is the error term. The matrix form of equation (2.1) is given by

$$y = X\beta + \varepsilon, \tag{2.2}$$

where $y = \begin{bmatrix} y_1 \\ ... \\ y_n \end{bmatrix}_{nx1}$, $X = \begin{bmatrix} 1 & x_{11} & ... & x_{k1} \\ ... & ... & ... & ... \\ 1 & x_{1n} & ... & x_{kn} \end{bmatrix}_{nxp}$ and $\varepsilon = \begin{bmatrix} \varepsilon_1 \\ ... \\ \varepsilon_n \end{bmatrix}_{nx1}$, where $p=k+1$.

The goal of regression analysis is to obtain estimates of the unknown regression parameters $\beta_0, \beta_1, ..., \beta_k$ which indicate how a change in one of the explanatory variables affects the values taken by the response variable.

There are various methods to estimate regression parameters in a linear regression model. The most commonly used technique to find the best estimates of β is the method of ordinary least squares (OLS). The least squares estimates vector of β is the value $\hat{\beta}$, which minimizes $\varepsilon'\varepsilon$. It can be determined by differentiating $\varepsilon'\varepsilon$ with respect to β and setting the resultant matrix equation equal to zero, at the same time replacing β by $\hat{\beta}$. This provides the normal equations

$$(X'X)\hat{\beta} = X'y.$$

Solution of the normal equations can be written as

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

By using this, the vectors of fitted values and residuals are obtained as

$$\hat{y} = X\hat{\beta}$$

and

$$e = y - \hat{y} = (I - H)y,$$

respectively, where $H = (h_{ii}) = X(X'X)^{-1} X'$.

The major assumptions in the study of regression analysis are:

1. **Linearity Assumption:** For each value of X there is an array of possible y values which is normally distributed about the regression line, i.e. each observed response value y_i can be written as a linear function of the *i*th row of X, x_i', that is

$$y_i = x_i' \beta + \varepsilon, \quad i=1, 2,..., n.$$

2. Computational Assumption: In order to find a unique estimator of β it is necessary that $(X'X)^{-1}$ exists, or equivalently *rank(X) = p.*

3. Distributional Assumptions: The statistical analyses based on least squares (e.g., the t-tests, the F-test, etc.) assume that

a. the error term (ε_i) does not depend on the explanatory variable (x_i), for $i=1,2,...,n$,

b. the values of X columns are assumed fixed and to be linearly independent of each other and measured without errors,

c. the error terms are assumed to be independently and identically distributed normal random variables each with 0 mean and σ^2 constant variance, that is $\varepsilon \sim N(0, \sigma^2 I)$.

d. The response variables are assumed to be normal each with $X\beta$ mean and σ^2 constant variance, that is $y \sim N(X\beta, \sigma^2 I)$.

4. The implicit Assumption: All observations are equally reliable and should have an equal role in determining the least squares results and in influencing conclusions.

2.2. Outliers

An outlier is an observation that lies outside the overall pattern of the other observations. If a data point stands apart from the main body of points in a plot, it should be investigated to see what may have caused this. Perhaps a number has been incorrectly recorded. Or the experimenter who collected the data may remember some special circumstance for that data point. So one has to decide what to do with these outliers. There are two possibilities.

- One possibility is that the outlier was due to chance. In this case, you should keep the value in your analyses. The value comes from the same population as the other values, so should be included.

- The other possibility is that the outlier was due to a mistake-bad pipetting, voltage spike, holes in filters, etc. Since including an erroneous value will give invalid results, you should remove it. In other words, the value comes from a different population than the other and is misleading.

The problem, of course, is that you can never be sure which of these possibilities is correct. Clearly, no mathematical calculation will tell you for sure whether the outlier comes from the same or different population. But statistical calculations can answer this question: If the values really were all sampled from a Gaussian distribution, what is the chance that you'd find one value as far from the others as you observed? If this probability is small, then you will conclude that the outlier is likely to be an erroneous value, and you have justification to exclude it from your analyses.

Statisticians have devised several methods for detecting outliers. All the methods first quantify how far the outlier is from the other values. This can be the difference between the outlier and the mean of all points, the difference between the outlier and the mean of the remaining values, or the difference between the outlier and the next closest value. Next, standardize this value by dividing by some measure of scatter, such as the standard deviation of all values, the standard deviation of the remaining values, or the range of the data. Finally, compute a p-value answering this question: If all the values were really sampled from a Gaussian population, what is the chance of randomly obtaining an outlier so far from the other values? If the p-value is small, you conclude that the deviation of the outlier from the other values is statistically significant.

In regression model, there are three types of outliers:

- outliers in x-space
- outliers in y- space
- Regression or residual outliers

Outliers in the x-space are points lying far away from the rest when examining the x-values only. This means we do not use knowledge about the relationship between x and y. These are considered high-leverage points because they are influential and pull the regression surface toward them. The extreme observation(s) in the x-space can be measured by using the leverage value (h_{ii}) where h_{ii} is the diagonal elements of hat matrix, $H = X(X'X)^{-1}X'$.

Outliers in the y-space are points that are outlying in the response variable because of distant values from the responses of the clean cases. Detection of outliers in y is a univariate problem that can be handled with the usual univariate tests.

An observation is called the *xy-space outlier* if it is outlier in both the x-space and the y-space.

Regression or residual outliers are those that present a different relationship between x and y, or in other words, samples that do not fit the model. This type of observation's Student residual is large in magnitude compared to other observations in data set.

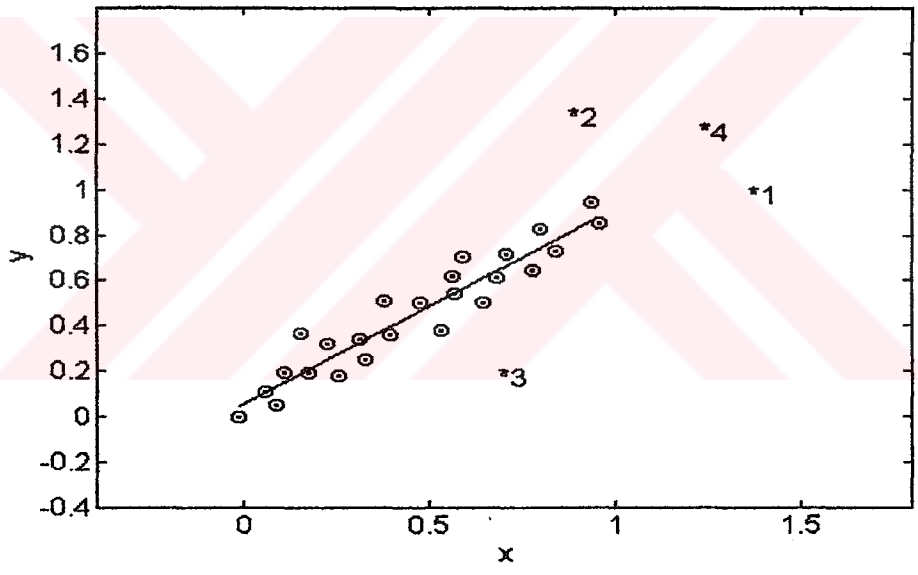

Figure 2.1. Illustration of the different types of outliers

We can see an illustration of the different types of outliers in Figure 2.1. In this figure (*1) is *x* and regression outlier, (*2) is *y* and regression outlier, (*3) is a regression outlier and (*4) is the xy-space outlier.

2.3. Detecting Outliers

The identification of outliers is often thought of as a means to eliminate observations from a data to avoid disturbance or further analyses. But outliers may as

well be the interesting observations in themselves, because they can give us hints about certain structures in the data or about special events during the sampling period. Therefore, appropriate methods for the detection of outliers are needed. Numerious influence diagnostics and graphical methods have been developed for the detection of outliers. I can give some well known influence diagnostics to detect outliers in a linear regression model. Hat matrix diagonals

$$h_{ii} = x_i'(X'X)^{-1}x_i \qquad i = 1,2,...,n,$$

indicate how outlying observations are in the x-space in the relation to the centroid (Hoaglin and Welsch, 1978). $DFFITS_i$ measures the effect of the *ith* observation on the predicted values (Belsley et al., 1980), which is given by

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{i(i)}}{\sqrt{s^2_{(i)}h_{ii}}} \qquad i = 1,2,...,n.$$

(*Note that the subscript (i) indicates that observation i is omitted from the calculation. For example;* $s^2_{(i)}$ *corresponds to the estimated variance after eliminating the ith observation from a data set*). $COVRATIO_i$ gives information on the influence of the *ith* observation on the precision of estimation for the regression coefficients (Andrews and Pregibon, 1978), which is given by

$$COVRATIO_i = \frac{\left|(X'_{(i)}X_{(i)})^{-1}s^2_{(i)}\right|}{\left|(X'X)^{-1}s^2\right|} \qquad i = 1,2,...,n.$$

Cook's distance (D_i) meausures the change in the regression coefficients that would occur if the *ith* case was omitted (Cook, 1977), which is given by

$$D_i = \frac{(\hat{\beta} - \hat{\beta}_{(i)})'X'X(\hat{\beta} - \hat{\beta}_{(i)})}{p.s^2} \qquad i = 1,2,...,n.$$

All of the above methods known as "single-row diagnostics" because they measure the effects of the *ith* observation as if it were deleted from the data set. They work well when data set contains only a single outlying point. However, they have been shown to fail in the presence of multiple outliers.

Most of the single-row deletion diagnostics can be extended in order to assess the changes caused by the deletion of more than one observation at the same time. These generalizations are called multiple-row diagnostics. Belsley et al. (1980) generalized many of the common single-row deletion diagnostics for multiple-rows. But without knowing the reasonable candidate set of cases to assess for joint influence, using the multiple-row diagnostics is computationally impractical because one has to analyze all possible combination of cases in the analysis. Because of this, multiple-row diagnostics are less popular than their single-row counterparts. For minimizing the computational cost Barnett and Ling (1992) propose a new approach, which upon developing cut off value on common diagnostics. However, develop appropriate cut off value seems unsatisfactory because in practice one will never really know it.

Beside of the computational problem there are some unclear points on multiple-row diagnostics. For example, do measures provide information that is similar to their single-row version? Specifically, is their behavior consistent with their single-row versions and are the interpretations of these diagnostics the same? Or, do any of these measures work better than others, and if so, in what type of situations?

To overcome the limitations of these approaches robust and combining methods have been proposed, which will be given in Chapter 3.

3. PREVIOUSLY PROPOSED MULTIPLE OUTLIER DETECTION METHODS

3.1. Introduction

Numerous classical and robust methods have been proposed in the literature for outlier detection in multivariate and regression data. Generally, all multiple outlier detection methods have a similar strategy. First, identify a candidate subset of observations that contains potential outliers and then test whether they are outlying significantly or not. Multiple outlier detection methods are examined in two broad categories, which are defined by Hadi and Simonoff (1993): direct methods and indirect methods. Direct methods are techniques that use classical estimators in order to obtain candidate outliers. Most of these methods first eliminate outliers from a data set, and then the analyst comfortably uses the classical statistical analysis on this clean data. Alternatively, indirect methods use robust estimators and accommodate outliers in their analysis. There are also many graphical and informal methods for detecting outliers in the statistical literature. However, I will not give these in this section since they are not objectively comparable to more formal procedures. In the following sections some of the graphical procedures are used as a part of confirmatory analysis for detecting outlying observations visually. As I go along I will mention these and make use of them.

There is a vast amount of literature on identification of outlying observations in a data, and many suggestions are given on what to do with these outliers. In this section only recently proposed outlier detection methods in multivariate and regression data are given. Detailed summaries of many other multiple outlier detection procedures can be found in Hawkins (1980), Hadi and Simonoff (1993), Barnett and Lewis (1994), and Sebert (1996) and Winskowski et al. (2001).

Identification of outliers in a multivariate data attracted many researchers for years; therefore many direct and indirect methods are developed for this type of data. In this thesis, although identification of outliers in a multivariate data is considered I will give more weight to the outlier detection methods in regression data since identification

of outliers for regression data requires more complicated methods, which should be capable to detect outliers in *x*-space, *y*-space or *xy*-space. Therefore using the *combined* techniques for the identification of outlying observations in regression data is useful for identification of outliers. In Chapter 6, the effectiveness and the power of these methods is compared by conducting an extensive simulation study.

3.2. Direct Methods

The earliest direct methods for univariate and multivariate data are based on a distance or a metric, which tells the analysts how far away the observations from the center of the data. For univariate data, there are many procedures to identify or to test outliers (for detailed information, see Hawkins, 1980 and Barnett and Lewis, 1994). There is no difficulty on identifying outliers in univariate data. However when we have multivariate data, then identifying outliers (single or multiple) gets more complicated because of two serious problems that arise in higher dimensions: swamping and masking. Masking occurs when the data contains outliers but fail to detect them and swamping occurs when we wrongly declare some of the non-outlying points as outliers.

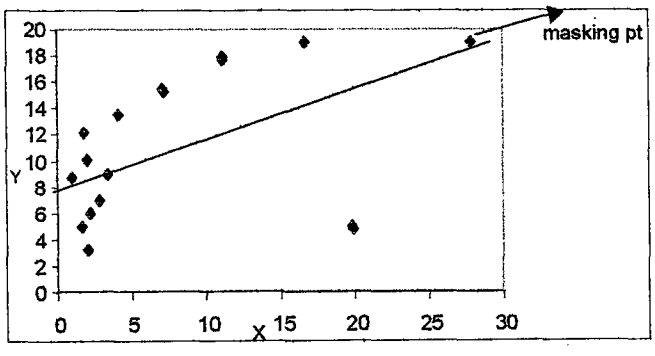

Figure 3.1. Illustration of masking problem

For identifying multiple outliers in a multivariate data a suitable metric is chosen to measure the distances between each observation x_i and the center of the observations, this distance is given by

$$d_i(C,V)=f(x_i - C, V), \qquad \text{for } i=1,...,n,$$

where C is the center (location parameter) and V is the covariance matrix (scale parameter). All of the procedures proposed for identification of multiple outliers in a multivariate data so far are based on this idea.

The earliest direct methods in regression data use the single-case diagnostics (*Prescott 1975, Tietjen, et. al 1973*) in a forward or backward stepping manner for identifying and testing observations. In forward-stepping search, the analysts determine the observation, which has the most extreme residual (or distance) first and, provided this observation is declared to be an outlier, next determine the one with the second-most-extreme residual (or distance), and so on. Backward-stepping search works similarly. The entire set of observations is initially considered and the outliers are sequentially removed by using some criteria such as the largest absolute value of transformed residual. It is well known, however, that in the presence of multiple outliers, these procedures cannot be effective because of masking and swamping problems, and combinatorical problems.

In this section I examine direct methods for two types of data: ***multivariate*** and ***regression data***. Since our purpose in this study is to focus on the methods defined in regression data, I will give a brief description for the methods defined for multivariate data.

3.2.1. Direct Methods in Multivariate Data

Outlier detection for univariate data is studied very widely by many researchers in statistical literature since 1950. Until 1980, there has been a little work on outlier

detection methods for multivariate data which were mainly on graphical and informal procedures. Pioneers for these procedures are Healy (1968), Andrews (1972), Gnanadesikan and Kettenring (1972), and Rohlf (1975). However these methods cannot be used as outlier-detection tests since they only provide informal information on the detection of outliers.

Outliers in a multivariate data matrix X_{nxp}, which is assumed to be multivariate elliptically distributed, can be hard to detect, especially when the dimension p exceeds 2, because we can no longer rely on visual perception. A classical method for detecting multivariate outliers is to compute the $d_i(C,V)=f(x_i-C,V)$, $i=1,...,n$ distances for each point x_i. The most commonly used form of $d_i(C,V)$ is given by

$$d_i(C,V)=\sqrt{(x_i - C)'V^{-1}(x_i - C)}, \quad i=1,...,n, \qquad (3.3)$$

where C is the location and V is the scale parameter.

The classical choices of C and V are the arithmetic mean ($\bar{x}$) and the sample covariance matrix (s) of the data matrix X, respectively. In which case equation (3.3) becomes

$$MD_i=di(\bar{x}, s)= \sqrt{(x_i - \bar{x})'s^{-1}(x_i - \bar{x})}, \quad i=1,...,n, \qquad (3.4)$$

which is called the Mahalanobis Distance (MD_i). A large value of MD_i (i.e if $MD_i > \chi_{p-1,\alpha}$, where $\chi_{p-1,\alpha}$ is square root of the 1-α quantiles of the chi-squared distribution with p-1 degrees of freedom) may indicate that the corresponding observation is an outlier. But this measure may fail to detect outliers because it depends on the sample mean and the covariance matrix, which are known to be sensitive to the presence of outliers. One way to solve this problem is to replace the mean and covariance matrix by more robust or resistance estimators of the location and scale.

There are many robust estimators for C and V, which will be given in the context of indirect methods.

Wilks (1963) proposes an outlier test, which is based on the change of the determinant of the sample scatter matrix after some observations are eliminated from the sample. Schawager and Margolin (1982) constructed a locally best invariant test for testing outliers in the multivariate mean-shift model, and the test statistics is found to be the same as Mardia's (1970) multivariate sample kurtosis. Note that as most outlier tests are based on an underlying normality assumption, they have a close relationship with tests for normality, as they test for a particular type of deviation from a normal distribution. Although the multivariate sample kurtosis is locally best invariant test for normality, it does not indicate which observations are outlying.

Bacon-Shone and Fung (1987) suggest a graphical approach based on Wilk's (1963) statistics. The method is found to be useful in the detection of outliers in univariate and multivariate data. In this method the following ratio is calculated by

$$R_{(t)} = |s_{(t)}| / |s|,$$

where $|s|$ and $|s_{(t)}|$ are the determinants of the sample scatter matrix over n and n-m observations, respectively, where $s = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})'$, $s_{(t)} = \sum_{i=1}^{n} (x_i - \bar{x}_{(t)})(x_i - \bar{x}_{(t)})'$. Here $t=(t_1,t_2,...,t_m)^T$ is an m-vector of indices of selected cases, where $1 \le t_i \le n$, and (t) mean that m observations, indexed by t, are being omitted. The smaller value of $R_{(t)}$ the greater decrease in the sum of squared volumes generated after elimination of these m observations, in other words, the more outlying these observations are. If $\min_t R_{(t)}$ is very small compared with the other quantities in the set $\{R_{(t)}$, all possible $t\}$, it is thought highly likely that these m observations are outliers. The plot of the observed $\{R_{(t)}\}$ versus the expected quantities gives valuable information in the understanding of the nature of

outliers in the sample. However, the expected quantiles are very difficult to obtain because $\{R_{(i)}\}$ are correlated. So they instead, calculate the expected quantiles, assuming a form of conditional independence in $\{R_{(i)}\}$, and, use this approximation for the purpose of plotting.

Simonoff (1991) presents a method based on clustering, which is defined by using combination of a robust scaling and backward stepping. In his method, after constructing the single linkage-clustering tree for a given data, clusters are ordered as most extreme to least extreme by order of joining. At each joining identify the cases in the smaller cluster first. If the cluster being examined splits further, repeat the above steps. Later as each case is identified, calculate Mahalanobis distances as a measure for the test statistics. If these Mahalanobis distances are higher than cutoff value then remove these cases from the data. After that test statistics are re-computed and again the next most extreme observation is tested. If a test statistic value of an observation is beyond the cutoff value then this observation and all of the previously removed observations are declared to be outliers and procedure stops. But important questions left to resolve include how to determine appropriate cutoff values in general, and whether other clustering approaches might be more effective in particular situations.

Caroni and Prescott (1992) suggest one outlier test for detecting of outliers in multivariate data, defined as a generalization of Wilks's test, and determined appropriate critical values in their study.

After 1992 a general set up has been emerged for the detection of outliers in multivariate data. This general method has two stages; the first is to find the basic subset, which is assumed free from outliers, and the second is to test for the outliers. In order to determine whether the observations are consistent with the basic subset, several forms of C and V are defined. I will give some forms of C and V in this section, and the robust versions of C and V will be given in the following section.

Hadi (1992a, 1994) proposes a procedure for detection of multiple outliers in multivariate data. The main idea of this method is to form a basic subset of about half of the data which is presumably free from multivariate outliers, then to add observations

that are consistent with the basic subset until it includes all the data values not nominated as outliers. If all the observations are added to the basic subset, the data is declared to be free from outliers, otherwise the observations that are not consistent with the basic subset are declared as multivariate outliers.

To determine whether the observations are consistent with the basic subset, Hadi (1992a, 1994) suggest using the distances

$$d_i(\bar{x}_b, s_b) = \sqrt{(x_i - \bar{x}_b)' s_b^{-1} (x_i - \bar{x}_b)} \qquad i=1,\ldots,n \qquad (3.5)$$

where $\bar{x}_b$ and s_b are the mean and covariance matrix of the basic subset, respectively. If the basic subset is free from outliers, then $\bar{x}_b$ and s_b will be reliable estimates for C and V in the general form (3.3), and then $d_i(\bar{x}_b, s_b)$ would be effective in the detection of the outliers.

If a data comes from a p-variate normal distribution, then $d_i^2(\bar{x}_b, s_b)$ follows approximately a χ^2 distribution with $p-1$ degrees of freedom, which is used as a cutoff value for detecting outliers.

Because the basic subset of "clean" values increases by one at each step Hadi's method requires at most $n-p$ covariance matrix computation and inversions. Furthermore, it needs ordering of the observations at each step of a process that can have $n-p$ steps. Although $n-p$ represents significant reduction in computing expense when compared to earlier methods, these methods can be prohibitively expensive to compute for large data.

Atkinson and Mulira (1993) describe the fast forward algorithm for finding multivariate outliers. The algorithm starts with a small, randomly selected subset of observations intended to be outlier free. This is incremented in such a way that outliers are unlikely to be included as long as the subset remains outlier free. The process is repeated several times from random starting points. This method can be viewed as a

search for a global minimum, the chances of success being increased by repeated random starts. The pattern of multivariate outliers during each search can be exhibited using stalactite plot. They use Mahalanobis distances, which calculate by sequential construction of an outlier free subset of the data, starting from a small random subset. The stalactite plot provides a cogent summary of suspected outliers as the subset size increases. Combined with probability plots and resampling procedures, the stalactite plot, particularly in its normalized form, leads to identification of multivariate outliers, even in the presence of appreciable masking. Starting point of this algorithm is the same as used by Rousseeuw and van Zomeren's (1990) random search algorithm.

Most of the above methods suffer from a lack of generality and a computational cost that escalates rapidly with the sample size. For samples of a sufficient size to support sophisticated methods, the computation cost often makes outlier detection unattractive.

Therefore, Billor et al. (2000) propose a general approach called **B**locked **A**daptive **C**omputationally **E**fficient **O**utlier **N**ominators (BACON) that is based on Hadi (1992a, 1994) and Hadi and Simonoff (1993) methods that can be computed quickly regardless of the sample size. This method first identifies initial clean subset that can safely be assumed free of outliers. Initial clean subset selected in two different ways: based on Mahalanobis distances (Version 1) or based on distances from medians (Version 2). For version 1 one has to compute the Mahalanobis distances, which is given by equation (3.4), identify $m=p.c$ observations, which constitute basic subset, with the smallest values of $d_i(\bar{x}, s)$. Here c is a small integer chosen by the data analyst, generally $c=4$ *or* 5 are suggested (Billor et al., 2000).

On the other hand for version 2 potential basic subset is determined by finding the smallest $m=p.c$ values of $\|x_i - m\|$, where m is a vector containing the coordinatewise median, x_i is the *ith* row of X, and $\|\cdot\|$ is the vector norm.

After selecting initial clean subset, the analysts fit an appropriate model to clean subset and calculate the discrepancies (d_i) for each of the observations by using this

model. Then a larger basic subset consisting of observations known to be homogeneous with the basic subsets is determined. This procedure is repeated until the size of clean subset no longer changes. At the end the observation(s) excluded by the final clean subset is declared as outliers.

V1 is affine equivariant (i.e. it is not affected by the linear transformations of the explanatory variables) at the expense of a somewhat lower breakdown point (which *indicates the maximum proportion of gross outliers which the induced estimators can tolerate*), about %20, whereas Version 2 (V2) requires more computations than Version 1 (V1) because of the computational cost of finding medians in all dimensions. In addition, V2 is nearly affine equivariant and has high breakdown point upwards of 40%.

By an extensive simulation study it has been shown that *BACON* methods match the performance of more computationally expensive methods on standard test problems and demonstrate their superior performance on large simulated data sets (Billor et al., 2000).

3.2.2. Direct Methods in Regression Data

Of particular importance to researchers in many fields is the detection of outlying observations (in x, in y or in xy) in linear regression modeling, because of the widespread use of regression methodology. The most work has been done in this area in the past three decades, particularly on the detection of a single outlier. However, as I mentioned above, in the presence of multiple outliers, these procedures do not reliably detect outliers because of masking and swamping problems; therefore, are not effective. Beckman and Cook (1983), Rousseeuw and Leroy (1987), and Hadi and Simonoff (1993) discuss these problems in details. Therefore, the emphasis has shifted to find methods, which are effective in the presence of multiple outliers.

Marasinghe (1985) adapted such methods into a multistage approach. In this method k candidate outliers are determined by using sequential application of a single-case diagnostic, each candidate outlier is tested for outlyingness until one is deemed non-significant. For determining candidate outliers, the analysts fit the

regression model to the complete data and determine the observation with maximum absolute studentized residual in this model, remove this observation from the data and fit the model again with the largest absolute studentized residual from this fit. We repeat this procedure until a subset of size k observations is determined. Once the k observations are identified they are sequentially tested by the test statistic $F_k=(S-Q(D_m))/S$ where $S=(n-p)s^2$ and $Q(D_m)$ is the reduction in the residual sums of squares resulting from the deletion of the suspect observations. Marasinghe (1985) provides percentage observations for F_k, which allows formal testing to determine which potential outliers are outlying observations.

Paul and Fung (1991) propose a generalized extreme studentized residual (*GESR*) multiple outlier detection procedure in an attempt to improve the procedure of Marasinghe (1985). Unfortunately, both of these methods require knowledge of k, which typically unknown. Also, because both use single-case diagnostics at the first stage, they are susceptible to masking and swamping effects.

The method proposed by Paul and Fung (1991) as an improved algorithm of Marasinghe has two phases, therefore called "two-phase procedure". In phase 1, this method identifies two sets of suspect observations, one by using the *GESR* and the other by using one of the diagnostics, Cook's distances (D_i) or Least Median Squares (*LMS, which is a robust method*), sequentially. The potential list of outliers will be the union of these two sets. In phase *2*, the observations from this combined group are then tested from least to most extreme, using the *GESR* procedure.

The method of Paul and Fung (1991) uses the Cook's distance in phase 1 to be able to identify outliers in x direction as well as in y direction since the Cook's distance is a function of residual and leverage values, and needs less computation than *LMS*. However, it is well known that the Cook's distance is subject to masking and swamping, so the two-phase procedure will be also. Therefore Hadi and Simonoff (1993) show that how this procedure fails in identifying xy-space outliers that form a cluster.

Kianifard and Swallow (1989, 1990) introduce procedures using recursive residuals, calculated on adaptively ordered observations, to identify multiple outliers in a

linear regression model. It involves first ordering the observations by a single-case diagnostic, and then using recursive residuals (see Brown et al., 1975) based on the k observations with smallest diagnostic to identify outlying observations. They propose using Cook's distance, the studentized residual, or some other single case diagnostic for ordering observations. Test statistics, which use for determining the initial ordering observations, suffer from masking. Therefore, this method is not effective in the existence of multiple outliers. Swallow and Kianifard (1996) define a new version of this method, which is called *recursive residual forward search algorithm* to decrease the drawbacks of the previously proposed method. However this method is computationally very expensive when the dimension of data matrix increases and still not resistant to masking and swamping problems in the data matrix (Winskowski et al., 2001).

Hadi and Simonoff (1993) propose two test procedures for detection of multiple outliers. The first procedure is an adaptation of the method proposed by Hadi (1992, 1994) and is related to the proposal of Hawkins et al. (1984). Atkinson and Mulira (1993) propose a graphical representation of potential multivariate outliers based on the similar idea as well. The second procedure is an adaptation of the clustering-based backward-stepping method proposed by Simonoff (1991).

Both procedures suggested by Hadi and Simonoff (1993) attempt to separate the data into two groups: set of clean observations and a set of potential outliers and do analysis in two parts. In the first part potential outliers are determined and in the second part potential outliers are tested to see how extreme they are relative to the clean subset, using an appropriately scaled version of the prediction error.

Steps for these two procedures are:

- **Step 1:** Find the least squares estimate (*LSE*) for the full data, the n observations are ordered by appropriate diagnostic measure, like absolute value of the adjusted residual $a_i = e_i / (1 - h_{ii})^{1/2}$, or Cook's distance ($D_i$).

For the first procedure: the first p observations form the initial basic subset. A model is fitted to the basic subset, and the residuals are standardized and ordered. The basic set is increased one by one, by ordering the standardized residuals and fitting a model to the basic subset until the basic subset reaches a size equal to the integer part of $h=(n+p-1)/2$.

For the second procedure: we determine the initial clean basic subset from the single linkage-clustering tree, and then order clusters from most to least extreme by the order of joining; the later a cluster joins, the more extreme it is. When the number of identified observation reaches $n-h$, the h remaining observations constitute the basic subset.

- **Step 2:**After determining the basic subset, both of the procedures require the calculations of the following statistics:

$$t_i = \begin{cases} \dfrac{y_i - x_i' \hat{\beta}_B}{\hat{\sigma}_B \sqrt{1 - x_i' (X_B' X_B)^{-1} x_i}}, & i \in B \\ \dfrac{y_i - x_i' \hat{\beta}_B}{\hat{\sigma}_B \sqrt{1 + x_i' (X_B' X_B)^{-1} x_i}}, & i \notin B \end{cases},$$

where B denotes observations in the basic subset. Then t_i values are ordered and the lowest $p + 1$ cases form the new subset B. The procedure continues to add observations until the $(r + 1)^{st}$ ordered residual measure exceeds $t_{(\alpha/2(r+1),r-p)}$, where r is the number of observations in the current subset. Observations $r + 1$ to n are the outliers. The key to the success of the method is to obtain a clean initial subset of data.

These methods are not highly computationally intensive compared to the ones suggested before, and relatively resistant to both masking and swamping effects. Hadi

and Simonoff (1997) improve this algorithm by using the robust distance measures in Hadi's (1992, 1994) forward selection algorithm to determine the initial clean subset of size h observations.

Pena and Yohai (1995) present influence matrix algorithm, which identifies influential subsets in a linear regression model. This matrix is known as the uncentered covariance of a set of vectors, which represents the effect on the fit of the deletion of each data point. They suggest looking at the eigenvalues of an influence matrix, which is defined by $M = EDHDE/ps^2$ where E is the diagonal matrix of least squares residuals, D is the diagonal matrix with elements $(1-h_{ii})^{-1}$, h_{ii} is the *ith* diagonal element of the hat matrix, $H = X(X'X)^{-1}X'$, and s^2 is the usual mean square error estimate of the error variance. After examining the eigenvalues of M we determine the eigenvectors corresponding to large eigenvalues, then order the co-ordinates of the eigenvector v_i, obtaining $v_{i(1)} \le v_{i(2)} \le ... \le v_{i(n)}$ and compute the ratios $a_j = v_{i(j)}/v_{i(j-1)}$ for $j=n,...,n-c_1$ and $b_j = v_{i(j)}/v_{i(j+1)}$ for $j=1,...,c_2$. The constants c_1 and c_2 are smaller than $n/2$. Later find the candidate outlier sets, *seta* and *setb*, by determining the observations which satisfy $|a_j| > 2.5$ and $|b_j| > 2.5$, respectively. Then all candidate outliers are removed and the standard F-and t-statistics to test for groups or individual outliers are used. Pena and Yohai (1995) claim that large eigenvalues of this matrix will correspond to observations that are good candidates for joint influence however, they suggest that in instances where outliers form a high leverage cluster, this method may not work.

In the presence of high-leverage points identifying outliers in a regression data becomes more difficult. Therefore Chatterjee and Mächler (1997) propose an iteratively weighted least squares procedure as a robust fit for linear models. This algorithm gives low weight to points with high leverage and /or high residual. The weights are a function of leverage and residuals. Chatterjee- Mächler procedure is not very effective when there is extensive masking because it is based on the standard measure of leverage h_{ii}, which is affected by masking problem.

Sebert, et al. (1998) suggest one clustering-based approach, which is called clustering algorithm, for multiple outlier identification that utilized the predicted and residual values obtained from a least squares fit of the data. This approach clusters the standardized predicted and standardized residual values from a least squares fit and obtain a cluster tree. Based on Mojena's stopping rule, they cut the tree and form groups at a height of $\bar{h} + 1.25 s_h$ where $\bar{h}$ is the average of the tree cluster heights for all *n-1* clusters, and s_h is the unbiased standard deviation of the heights of the *n-1* clusters. And then they identify the group with a majority of the observations in it as the clean base subset. All other observations are candidate influential observations. For graphical display I give weights to the observations. If *ith* observation is a candidate outlier, then I give 1, otherwise 0 as a weight to that observation. Later by using multiple-row diagnostics all the candidate influential observations are assessed.

Billor et al. (2000) define the BACON regression method as an alternative to the method of Hadi and Simonoff (1993, 1997). In BACON regression method initial basic subset of at least m observations that is free of outliers is determined by applying the following steps which is different from Hadi and Simonoff (1993 and 1997):

- **Step 0:** Compute the discrepancies for all observations based on (3.5) relative to the final basic subset of BACON algorithm and determine the minimum m observations with the smallest values of the $d_i(\bar{x}_b, s_b)$. Let X_m be the subset (that contains m clean observations) of the matrix X and y_m is the response vector of m observations that corresponds to the m rows of the matrix X_m. If X_m is not of full rank, then increase the basic subset by adding observations with smallest values of $d_i(\bar{x}_b, s_b)$, until it has full rank. Then compute

$$t_i(X_m, y_m) = \begin{cases} \dfrac{y_i - x_i' \hat{\beta}_m}{\hat{\sigma}_m \sqrt{1 - x_i'(X_m' X_m)^{-1} x_i}}, & i \in m \\ \dfrac{y_i - x_i' \hat{\beta}_{mB}}{\hat{\sigma}_m \sqrt{1 + x_i'(X_m' X_m)^{-1} x_i}}, & i \notin m \end{cases} \quad \text{for } i=1,2,..,n,$$

where $\hat{\beta}_m$ and $\hat{\sigma}_m^2$ are the least-squares estimates of β and σ^2 based on the observations in the subset y_m and X_m. Later identify the $p+1$ observations with the smallest $|t_i(X_m, y_m)|$, and declare them to be the initial basic subset y_b and X_b.

- **Step 1:** If this new X_b is not full rank, increase the subset by as many observations as needed for it to become full rank, adding observation with smallest $|t_i(X_m, y_m)|$ first, and increase m by the number of observations added to make the subset full-rank. Compute

$$t_i(X_b, y_b) = \begin{cases} \dfrac{y_i - x_i' \hat{\beta}_b}{\hat{\sigma}_b \sqrt{1 - x_i'(X_b' X_b)^{-1} x_i}}, & i \in b \\ \dfrac{y_i - x_i' \hat{\beta}_b}{\hat{\sigma}_b \sqrt{1 + x_i'(X_b' X_b)^{-1} x_i}}, & i \notin b \end{cases} \quad \text{for } i=1,\ldots,n, \qquad (3.6)$$

where $\hat{\beta}_b$ and $\hat{\sigma}_b^2$ are the least-squares estimates of β and σ^2 based on the observations in the basic subset y_b and X_b.

- **Step 2:** If r is the size of current basic subset. Identify the $r+1$ observation with smallest $|t_i(X_b, y_b)|$ and declare them to be the new basic subset.
- **Step 3:** Repeat steps 1 and 2 until the basic subset contains m observations.

- **Step 4:** Find discrepancies, $t_i(X_b, y_b)$ as in (3.6)
- **Step 5:** The new basic subset consists of all points with distances less than $t_{(\alpha/2(r+1),r-p)}$, where r is the size of the current basic subset.
- **Step 6:** Iterate steps 4 and 5 until the size of the basic subset no longer changes.
- **Step 7:** Nominate the observations excluded by the final basic subset as outliers.

This algorithm is more computationally efficient than the method of Hadi and Simonoff (1993). Because, in order to determine the basic subset one needs very few regression calculations and evaluations of discrepancies. In this method there is no longer any needs to order the discrepancies, only one needs to check them against a constant, which requires only n operations.

Billor et al. (2000), propose using a diagnostic plot, which is a useful tool in analysis. This plot can be obtained by plotting the predicted residuals $t_i(X_b,y_b)$ obtained at the final iteration of this method versus the distances $d_i(\bar{x}_b, s_b)$ obtained in step 1.

Since Chatterjee-Mächler procedure is based on the standard measure of leverage h_{ii}, which is affected by masking problem, it is not effective when there is extensive masking. So Billor et al. (2001) proposes a procedure, called **Re-Weighted Least Squares** (*RWLS*), which uses a robust distance proposed by Billor et al. (2000) instead of using h_{ii} in the Chatterjee and Mächler algorithm (1997).

The RWLS algorithm consists of the following steps:

Step1: Obtain the initial weights as follows:

- Find m_d, which is the median of $(d_1, d_2, \ldots, d_n)$ where $(d_1, d_2, \ldots, d_n)$ is the normalized Hadi's distances obtained in the final step of BACON algorithm when it is applied to the matrix X.

- Compute the squared normalized version of the $d_i=1/max(d_i,m_d)$ by

$$d_i = \frac{d_i^2}{\sum_{i=1}^{n} d_i^2},$$

- Obtain the initial weighted least squares estimate of the regression coefficients ($\hat{\beta}_0$) by using d_i as a weight for the *ith* observations.

Step 2: Compute the weighted least squares estimate of the regression coefficients until convergence by using the following steps:

- Replace the residuals of the last fit e_i^{j-1} by its normalized version, where $e_i^{j-1} = y - \hat{y}^{j-1} = y - X\hat{\beta}^{j-1}$.
- Compute

$$a_i = \frac{1-d_i}{max\left(\left|e_i^{j-1}, m_e^{j-1}\right|\right)},$$

where m_e^{j-1} is the median of $\left(\left|e_1^{j-1}\right|, \left|e_2^{j-1}\right|, ..., \left|e_n^{j-1}\right|\right)$.

- Compute the new weights,

$$w_i^j = \frac{a_i^2}{\sum_{i=1}^{n} a_i^2},$$

- Compute the *jth* weighted least squares estimate of the regression coefficients $\hat{\beta}_j$ when using w_i^j as a weight for the *ith* observation.

The rationale for using the initial weights in step 1 is finding a robust measure of leverage. Since the leverage points are outliers in the x-space, the distances d_i are less

sensitive to the masking effect than the h_{ii}. These distances are normalized so that $0<d_i<1$. This way d_i will mimic the property of h_{ii}. The same idea applies to the weights.

This measure eliminates the distorting effect of masking by constructing a measure of location and dispersion for the variables in x, which is free from the effects of multivariate outliers, and clustering in the x-space. A diagnostic plot is used to indicate whether an observation is a leverage point, outlier, or a masking point, that is, a low leverage influential point. This method is resistant against outliers, conceptually simple, and not computer intensive.

3.2.3. Graphical Methods

Finding an outlier in regression data is important but sometimes methodology identifies too many outliers. So most of the analysts prefer using graphical methods after their methodology.

Graphical methods can be examined in two different groups. The graphical methods in one group are used as detection tools for finding outliers and the others are used for confirmatory analysis, which help us on deciding whether observations are really outlying or not.

3.2.3.1. Graphical methods for detecting of outliers

McCulloch and Meeter (1983) and Gray (1983, 1985, 1986) propose *contour* plots of single-case influence measures in terms of the diagonal elements of hat matrix and studentized residuals. Barnett and Ling (1992) propose the generalizations of the multiple-case measures. But this method cannot summarize the influence information on one single graph.

There are other plots such as *Leverage-Residual* (L-R) plot (Gray 1983, 1986), and *Potential-Residual* (P-R) plot (Hadi, 1992), which were proposed mainly for

detecting single outliers and influential observations. Despite the L-R plot can deal with single cases, P-R plot can deal with multiple cases.

3.2.3.2. Graphical Methods for confirmatory analysis

Atkinson (1986) suggests a method, which has two parts. In the first part outliers are determined by using LMS and in the second part the diagnostic methods of least-squares regression are used to confirm the findings of the robust method. In this study he proposes employing graphical methods for confirmation of outliers and defines adding-back plot. Later Fung (1993) proposes a confirmatory analysis alternative to the method of Atkinson (1986). He claimed that Atkinson's adding back statistics only shows up the observations for which the residuals is large in magnitude compared to other observations in the data set, does not show up high-leverage observations and his measure is not bounded above. Fung's method is a stepwise method with reference to the adding-back plots. In the analysis, first, observations are deleted from the reduced sample if they are outliers, and put back observations if they are not. This is done repeatedly until all the omitted observations are outliers.

On the other hand, Fung (1995) suggests that it would be informative to study the interrelationship using, if possible, one single graph for summarizing influence information of multiple observations. So he proposes two graphical methods with contours of constant measure values: Adding-back Leverage Outliers (ALO) and Generalized Leverage-Outlier (GLO) plot by using multiple-deletion and adding-back approaches, respectively. In these plots *–log (1-GL$_I$)* and *–log (1-OUT$_I$)* values are plotted together with contours for *–log (1-AP$_I$)*, D_I and/or other influence measures, where $GL_I = 1 - \left| X_{(I)}{}' X_{(I)} \right| / \left| X'X \right|$ (Hoaglin and Welsch, 1978), $OUT_I = 1 - s^2_{(I)} / s^2$ (Barnett and Lewis, 1994), and $AP_I = 1 - \left| Z_{(I)}{}' Z_{(I)} \right| / \left| Z'Z \right|$ (Andrews and Pregibon, 1978), Z is the augmented matrix of X and y.

ALO and GLO plots are useful as a diagnostic tool for summarizing the influence information and for determining whether a given set of observations is outliers or influential relative to the remaining observations.

These graphs are however not used as a detection tool as L-R and P-R plots are for finding outliers or influential observations. Therefore these graphics are not rivaled or in contradiction to other outlier detection methods (Gray and Ling, 1984, Simonoff, 1991) and robust methods (Atkinson, 1986, Rousseeuw and van Zomeren, 1990), but they are in fact complementary.

3.3. Indirect (Robust) Methods

3.3.1. Indirect Methods in Multivariate Data

The detection of outliers in multivariate data is recognized to be an important and difficult problem in physical, chemical, and engineering sciences. Most standard multivariate analysis methods rely on the assumption of normality and require the use of estimates for both the location and scale parameters of the distribution in their analysis.

It has been long recognized that classical methods are not effective in detection of outliers because they are sensitive to outliers. One way out of this problem is to use robust methods, which produce estimates resistant to the presence of outliers and/or to violations of distributional assumptions.

There are many robust methods, which work well (see Hawkins, 1980, Huber, 1981, Chatterjee and Hadi, 1988, Barnett and Lewis, 1994, Atkinson and Riani, 2000). The usual strategy of robust methods is based on the computation of robust distance for each observation $x_i \in R^p$, defined as

$$RD_i^2 = (x_i - C(X))' (V(X))^{-1} (x_i - C(X)), \qquad (3.7)$$

where $C(X)$ and $V(X)$ correspond to the robust estimators for the center and the covariance matrix, respectively. Many robust alternatives for these estimators have been proposed in the literature, such as the estimator based on minimum volume ellipsoid (MVE) (Rousseeuw, 1985) and its derivations such as the minimum covariance determinant (MCD) method.

Rousseeuw (1985) proposes the MVE method, which seeks the ellipsoid of minimum volume containing at least half of the data. The centre and the covariance matrix of the observations included in the MVE are robust location and covariance matrix estimators. Its standard algorithm is an approximate one based on the basic resampling scheme (Rousseeuw, 1984, Marazzi, 1991, Hadi and Simonoff, 1993). This proceeds by taking a subset of size $p+1$ of the cases and finding its sample mean vector and covariance matrix. The concentration ellipsoid is then rescaled until it includes just $n-h$ cases, where h is the integer portions of $(n+p+1)/2$ and the volume of this resulting scaled ellipsoid is computed. Doing this for all possible subsets of size $p+1$ (or for a large random sample if the total number of possible subsets is unmanageably large) one then approximates the MVE by the smallest scaled ellipsoid found in the search. The advantage of MVE estimators is that they have a breakdown point of approximately 50% (Lopuhaä and Rousseeuw, 1991). However it is computationally expensive. To deal with this problem, several algorithms have been suggested; one of them is the basic resampling algorithm (Rouseeuw and Leroy, 1987).

The basic resampling algorithm for approximating the MVE, called MINVOL, considers a trial subset of $p+1$ observation and calculates its mean and covariance matrix. The corresponding ellipsoid is then inflated or deflated to contain exactly h observations. This procedure is repeated many times, and the ellipsoid with the lowest volume is retained. For small data it is possible to consider all subsets of size $p+1$, whereas for larger datasets the trial subsets are drawn at random.

Several other algorithms have been proposed to approximate the MVE for example Woodruff and Rocke (1993). Some researchers developed algorithms to compute the MVE exactly. This work started with the algorithm of Cook et al. (1992),

which carries out an ingenious but still exhaustive search of all possible subsets of size h. In practice, this can be done for n up to about 30. Agullo (1996) developed an exact algorithm for the MVE that is based on a branch and bound procedure that selects the optimal subset without requiring the inspection of all subsets of size h. This is substantially faster and can be applied up to $n \leq 100$ and $p \leq 5$. Because for most data sets the exact algorithms would take too long, the MVE is typically computed by versions of MINVOL.

Alternative to MVE, Rousseeuw (1985) suggests MCD estimator. Its objective is to find h observations (out of n) whose covariance matrix has the lowest determinant. The MCD estimate of location is then the average of these h points and the MCD estimate of scatter is the covariance matrix of these h observations. This method suggests the following algorithm: generate all possible elemental sets, finding the mean vector and covariance matrix of each. Use this to compute cases Mahalanobis distances and hence to rank the cases from least to most outlying; trim the h most outlying cases, and use the sample mean vector and covariance matrix of the untrimmed cases. Take as the estimate the covariance matrix if minimum determinant found in this way.

MCD breakdown value equals that of the MVE, but the MCD has several advantages over the MVE. Its statistical efficiency is better because the MCD is asymptotically normal (Butler et al., 1993). Robust distances based on the MCD are more precise than those based on the MVE and hence better suited to expose multivariate outliers.

In spite of all these advantages, computing these estimators is very expensive, so approximations based on iterative algorithms are used. One of them is called FAST minimum covariance determinant (FAST-MCD), defined by Rousseuw and van Driessen (1999), which is actually much faster than any existing MVE algorithm. The basic idea begins with the classical estimator computed from an initial randomly selected subset of $p+1$ point called a "start," from which the Mahalanobis distances are computed for all n points. At the next iteration, the classical estimator is computed on

the $c \approx n/2$ cases corresponding to the smallest distances. This iteration continues until convergence. The final subset of c cases is named the "attractor" of the start. We use K starts and compute K estimators from the resulting attractors where $K>0$ is a large constant. The algorithm estimator is the one, which minimizes the *MCD* criteria.

For small data FAST-MCD typically finds the exact MCD, whereas for larger data it gives more accurate results than existing algorithms. Another advantage of FAST-MCD is its ability to detect exact fit situations.

3.3.2. Indirect Methods in Regression Data

The classical analysis of variance (ANOVA) technique based on the least squares is safe to use if the underlying experimental errors are normally distributed. However, data often contain outliers due to recording or other errors. It is important to distinguish these extreme points and determine whether they are outliers or important extreme cases. Robust regression is an important tool for analyzing data that are contaminated with outliers. It can be used to detect outliers and to provide resistant (stable) results in the presence of outliers. In general these procedures down-weight outlying observations in order to lessen their influence on the fitted regression.

An indirect approach to outlier identification is through a robust regression estimate. The main purpose of robust regression is to compute resistant (stable) estimator of the regression parameters in the presence of outliers. Most of them are computed by minimizing a symmetric function of the residuals

$$\min_{\beta} \sum_{i=1}^{n} \rho(e_i) \tag{3.8}$$

where β is the trial regression fit and ρ is the residual weighting function. Several residual weighting functions are possible. As special cases I note that $\rho(e_i) = e_i^2$ yields

the ordinary least square estimators, $\rho(e_i)=|e_i|$ yields the sample median, whilst $\rho(e_i)=-log\, f(e_i)$ yields the maximum likelihood estimators, where f is a symmetric function with a unique minimum at zero. These functions are selected depending on the assumptions of the model and problems of data. Historically, three classes of problems have been addressed in robust regression techniques:

- Problems with outliers in the y-*space* (response direction)
- Problems with multivariate outliers in the covariate space (i.e. outliers in the x- space, which are also referred to as leverage points)
- Problems with outliers in the *xy-space*

The multiple outlier detection methods for linear regression selected in this study are either those most recently published or those most frequently cited in the literature. These estimators are M-estimator (Huber, 1973), Least Median Squares estimator (LMS) (Rousseeuw, 1984), S estimator (Rousseeuw and Yohai, 1984), Least Trimmed Squared estimator (LTS) (Rousseeuw, 1985), Feasible Solution Algorithm for Least Trimmed Squares estimator (FAST-LTS) (Rousseeuw and van Driessen, 1999).

Huber (1973) presents M-estimation, which is the simplest approach both computationally and theoretically. This popular method of robust fitting replaces the sum of squares of the residuals function with a function of the residuals that is less sensitive to extreme outliers. These estimators are the solutions to $min_\beta \sum_{i=1}^{n} \rho(e_i/s)$, where s is the scale estimate to ensure that if the y values are multiplied by a constant c, and then the estimated regression coefficients will also be multiplied by c. Several residual weighting functions are possible based on the down weighting philosophy. More detail can be found in Montgomery and Peck (1992) and Rousseeuw and Leroy (1987). Although it is not robust with respect to leverage points, it is still used extensively in analyzing data for which it can be assumed that the contamination is

mainly in the response direction. Krasker and Welsch (1982) presented a generalized-M estimator (bounded influence estimators), which is less sensitive to leverage points.

Rousseeuw (1984) proposes LMS estimator, as a high-breakdown regression estimator, defined by

$$\underset{\beta}{minimize}\ \underset{i=1,\ldots,n}{median}\, e_i^2(\hat{\beta}), \tag{3.9}$$

where $e_i(\hat{\beta}) = y_i - x_i\hat{\beta}$ is the *ith* residual. After computing $\hat{\beta}$ one also calculates the corresponding scale estimate, given by

$$\hat{\sigma} = k\sqrt{\underset{i=1,\ldots,n}{mediane_i^2(\hat{\beta})}}, \tag{3.10}$$

where k is a positive constant. The standardized *LMS* residuals $e_i/\hat{\sigma}$ can then be used to indicate regression outliers, that is, points that deviate from the linear pattern of the majority (Rousseeuw, 1984). The *LMS* estimator has high breakdown point, namely $([(n-p)/2]+1)/n$, and is affine equivariant; however in methods based on *LMS*, the distributional properties of the errors are unknown. Therefore, cutoff values for the magnitude of residuals are unclear. The analyst typically just looks for large values of the residuals to determine outliers. This can lead to a subjective use of this procedure.

LTS estimator is proposed as an efficient alternative to *LMS* (Rousseeuw 1985). Its objective is to minimize

$$\sum_{i=1}^{h} e_i^2(\hat{\beta}), \tag{3.11}$$

where $e_1^2 \le \ldots \le e_h^2 \le \ldots \le e_n^2$ are the ordered squared residuals. This corresponds to minimizing the sum of the h smallest squared residuals; where h are the integer portions of $(n+p+1)/2$. The *LTS* regression estimate is then the least squares fit to these h points.

The breakdown value of *LTS* is *(n-h)/n*. Moreover, *LTS* regression has several advantages over *LMS*. *LTS* is more sensitive to local effects than *LMS and LTS* estimator's statistical efficiency is better than *LMS*, because the *LTS* estimate is asymptotically normal (Hossjer, 1994) whereas the *LMS* estimator has a lower convergence rate (Rousseuw, 1984). In spite of all these advantages, until now the *LTS* estimator has been applied less often because it was harder to compute than the *LMS*.

Therefore, Rousseeuw and van Driessen (2000) construct a new *LTS* algorithm, which is actually faster than all existing *LMS* algorithms, called *FAST-LTS*. The basic ideas of this method are an inequality involving order statistics and sums of squared residuals, and techniques, which we call selective iteration and nested extensions. For small data *FAST-LTS* typically finds the exacts *LTS*, whereas for larger data it gives more accurate results then existing algorithms for *LTS* and is faster by orders of magnitude.

Atkinson (1994) suggests using a fast robust method for detection of multiple outliers. The algorithm starts with a small, randomly selected subset of observations intended to be outlier free. This is incremented in such a way that outliers are unlikely to be included as long as the subset remains outlier free. As the search progresses forward through the data, the minimum value of the LMS or MVE is recorded, together with the pattern of outliers that it indicates. The process is repeated several times from random starting points. A similar search for the MVE was used by Hadi (1992), who used robustly estimated starting points for his single search. The pattern of multivariate outliers found during each search can be exhibited using the stalactite plot of Atkinson and Mulira (1993). This procedure will not be very effective when the number of variables p is large.

As mention in Section 3.2.2., Swallow and Kianifard (1996) propose a method called *recursive residual forward search algorithm* alternative to the Kianifard and Swallow (1986). In their study, they consider two familiar robust scale estimators: the interquartile range (*IR*) and the median absolute deviation from the median (*MAD*).

The algorithm first orders the magnitudes of the studentized residual values from a least squares fit to form the basis of p clean observations. Recursive residuals, w_j, are scaled by the median absolute deviation from the median (*MAD*) estimate of scale $\hat{\sigma}$. The w_j are defined as

$$w_j = \frac{y_j - \mathbf{x}'\hat{\boldsymbol{\beta}}_{j-1}}{(1+\mathbf{x}_j'(\mathbf{X}_{j-1}'\mathbf{X}_{j-1})^{-1}\mathbf{x}_j)^{1/2}}, \qquad j = p+1,...,n \tag{3.12}$$

where $\hat{\beta}_{j-1}$ is the parameter estimates computed over *j-1* observations, whose residuals are minimum. The test statistic $|w_j / \hat{\sigma}|$ for each observation is compared to a cutoff value to identify outliers. A correction factor for the *MAD* estimate of scale and the cutoff value come from simulation under the null hypothesis of no outliers. In this study they found that, *IR* and *MAD* worked equally well and robustly scaled statistics using recursive residuals performs as well or better than those using ordinary residuals.

All of the above algorithms require computation time that increases exponentially with the number of predictor variables, and thus can be applied only when this number is not too large. Therefore, Pena and Yohai (1999) propose a different type of approximate solution to the high-breakdown point estimates just mentioned that could be applied to a data set with a large number of predictor variables. This procedure succeeds in detecting groups of outliers in many situations where, due to a masking effect, the usual diagnostic procedures fail and robust estimates require prohibitive computer time. The main idea of this method is to use a robust scale for evaluating a set of possible solutions. These solutions are determined by applying *LSE* to sub samples in which sets of potentially outlier observations have been deleted. This method has two stages. The first stage is iterative, in each step, a robust estimate is found using the criterion of minimizing a robust scale of the residuals and observations with large residuals according to the current best estimate are deleted then by using the remaining sample, principal sensitivity components are computed. For each of these components,

extreme points are deleted, and an element of A is obtained as the *LSE* on the remaining sample. Elements of A are parameter estimates obtained by choosing at random N sub samples of different data points, where N is an integer number. The iterations continue until convergence.

In the second stage the efficiency of the robust estimate obtained in the first stage is improved. The residuals based on the estimate found in the first stage are computed, all observations with large residuals are eliminated, and the points deleted are tested one by one by using the studentized residual for outlyingness. The final estimate is computed by *LSE* using the cleaned sample. The estimate computed by the procedure is affine equivariant.

Very Recently, Hawkins and Olive (2002) introduce an algorithm based on clustering the data, called *X*-cluster, for large high-dimension multiple regression data and using L_1 fits within clusters. The aim of this method is to minimize the *LTS* objective function by taking successive concentration steps. Unlike most high breakdown estimators, it is increasingly successful with increasing sample size. Unfortunately it has some drawbacks. One of them is that it needs more computations especially for large n and p. (Rupert, 2002, Hubert et al., 2002, Hawkins et al., 2002). Because: this method consists of forming the within-cluster mean vector and covariance matrices of some random starting allocation and obtaining the inverse of the covariance matrices. Thereafter, one has to compute the impact of moving each case from its current cluster to each other cluster, and, if an improvement is possible, then the analyst has to update the two inverse covariance matrices. Moreover the initial setup involves more computations to get the starting mean vectors, covariance matrices, and their inverses. Investigating any one case for a possible move to another cluster needs more computation to get Mahalanobis distances, too. In addition to these, it is not foolproof and the breakdown value is not clear (Ruppert, 2002).

3.3.3. Graphical Procedures for Identification of Outliers

High-breakdown estimators are useful for detecting leverage points and masked observations. But they declare too many observations as extreme because of their high robustness. Therefore one needs confirmatory analysis, which seems necessary. So Atkinson (1986) suggests a new method, which is done in two-stages. The first stage of this method uses *LMS* and in the second stage the diagnostic methods of least-squares regression are used to confirm the findings of the robust method (i.e. Cook's distance). He proposes employing graphical methods for confirmation of outliers and defined adding-back plot.

LMS is useful for detecting outliers in *y* direction. But outliers in x-space are also of interest. Therefore Rousseeuw and van Zomeren (1990) define the robust version of the Mahalanobis distances based on minimum volume ellipsoid (*MVE*), where *C(x)* and V*(x),* defined in equation (3.1), are the robust mean and covariance, obtained from *MVE,* respectively. If *ith* observations robust distance value (RD_i) is greater than $\sqrt{\chi^2_{p-1,0.975}}$ then *ith* observation is regarded as x-space outliers or leverage points and the points outside the horizontal tolerance band *[-2.5, 2.5]* are regarded as regression outliers. In this study *LMS* residuals versus robust distances graph, which is called Robust Distances plot (*RD*), is proposed for identifying outliers easily. This type of plot presents a visual classification of the data into four categories: the **regular observations** with small RD_i and small robust residuals, the **vertical outliers** with small RD_i and large robust residuals, the **good leverage points** with large RD_i and small robust residuals, and the **bad leverage points** with large RD_i and large robust residuals. But unfortunately it needs more computational time and finds many outliers even in innocent data.

Alternative to *RD* plot, Rousseeuw and van Driessen (1999) suggest using *DD* plot, which is a plot of classical Mahalanobis distances (MD_i) versus robust distances (RD_i). Different forms of RD_i are defined. It depends on the choice of the estimators for example analysts can select *MVE* or *MCD* as a robust location estimator in RD_i.

This plot can be used as a diagnostic for determining multivariate normality and elliptical symmetry and to assess the success of numerical transformation towards elliptical symmetry. If the data distribution is multivariate normal, then the points in the *DD* plot will cluster tightly about the identity line; if the distribution is non-Gaussian but elliptically countered distribution, the points will still cluster tightly about a line but with no unit slope. In addition, the *DD* plot can often detect departures from elliptical symmetry such as outliers, the presence of two groups, or the presence of a mixture distribution (Olive 2002).

4. A NEW PROPOSED METHOD: BACON ROBUST PRINCIPLE COMPONENT ANALYSIS

4.1. Introduction

Observations, which are grossly outliers in a single component, can often be detected by applying univariate techniques to each variable. However it is difficult to cope with outliers in a high dimensional data. First of all, the computational cost to analyze such data (e.g. by cluster analysis) can be high. Furthermore, many statistical analyses are sensitive to outliers in the presence of correlated variables, which are known as multicollinearity problem. To overcome the latter problem, the classic principle component analysis (PCA) is often used to reduce the dimension of a data matrix by creating a new set of uncorrelated variables, which corresponds to eigenvectors of the sample covariance matrix. Because the covariance matrix is based on the sample mean it is excessively sensitive to outliers. Therefore, using **robust** or **combining methods of PCA** is necessary in the case of possibility of the existence of outliers.

Robust principal component analysis can easily performed by computing the eigenvalues and eigenvectors of a robust covariance or correlation matrix. The first study on robust principal component analysis is introduced by Campbell (1980), which is called robust principle component analysis (RPCA) method. This method does provide the usual valuable features of PCA as an aid to the understanding and interpretation of multivariate data, and also computes weights for the contribution of each observation to the robust estimation of each component. Low weights indicate outliers. This method gives a useful way of examining the structure of the data and testing for outliers at the same time. But since it is based on M-estimation method, it provides protection against small fraction of outliers (Rousseeuw and van Zomeren, 1990). Campbell (1980) also proposed a useful graphical display to detect outliers in multivariate data, which is the probability plot of Mahalanobis squared distances using

robust M-estimates of the mean and the covariance matrix. Later Lie and Chen (1985) propose a method based on projection pursuit (PP) for finding low dimensional projections that provide the most revealing views of full dimensional data. The idea of Lie-Chen method is to search for the directions in which the projected observations that have the largest robust scale. In subsequent steps, each new direction is then constrained to be orthogonal to all previous directions. However this method has some drawbacks. It has been shown (Croux and Ruiz-Gazen, 2000, Croux and Filzmoser, 2001, Pires and Branco, 2001) that the corresponding influence functions are unbounded, thereby showing a lack of local robustness. Another drawback of the method is that it is not obvious how to compute the PP-based estimators. Croux and Ruiz-Gazen (1996, 2000) introduced computationally much more attractive C-R algorithm than the PP algorithm. Although the C-R algorithm performs well in low dimension, it lacks numerical stability in high dimensions. Croux and Ruiz-Gazen (2000) compared the C-R algorithm with the PP algorithm for the efficiency and robustness of several robust scale estimators. Hubert, et al. (2001) gave an improvement of C-R algorithm through dimension reduction, called a faster two-step algorithm denoted by RAPCA, which stands for reflection-based algorithm for principal components analysis. This method is more stable numerically than the C-R algorithm and can deal with situations where there are more variables than observations. Furthermore it combines numerical accuracy with computational speed. On the other hand, Caroni (2000) made an extensive study on Campbell's (1980) method. In this study a simulation study is carried out to determine critical values for the weights of observations in RPCA under the null hypotheses $x_i \sim N_p(\mu, \Sigma)$ $i = 1,...,n$. By calculating these critical values, this method can be determined correctly what constitutes a low weight. Therefore he showed that in this way the RPCA could be employed as a formal test for outliers.

Unfortunately, neither of these approaches are without problems. There has been great difficulty developing robust methods that combine all of the desirable properties of a statistical procedure: *efficiency at the assumed model, strong resistance to different*

types of outliers and *relative ease of computation and applications*. Thus, in recent years, some of the algorithms are proposed as an alternative to robust methods as given in Chapter 3, *combined* methods. Identification of outliers using appropriate *combined* methods can be considerably more effective than the direct use of a robust analysis.

In Section 4.2., I propose a new *combined* method on principle component analysis by using *BACON* algorithm (Billor et al., 2000). The performance of the methodology on some typical data and a simulation study are given in Sections 4.3. and 4.4., respectively. A final discussion and conclusions are given in Section 4.5.

4.2. BACON Robust Principle Components Analysis (*BRPCA*)

Proposed methodology is based on using the classical estimators of the mean and the covariance matrix obtained from *BACON* algorithm instead of using robust *M*-estimators in *RPCA* (Campbell, 1980) or the estimators obtained by *RAPCA* (Hubert et al. 2002) since the estimators obtained from these methods are still sensitive to outlying observations.

The main idea in this approach is to determine clean data by using a technique called *BACON* (Billor et al., 2000), which is proven to be very effective for large data sets and then is to use the classical principal component analysis. Therefore, the analysts will be able to find the significant components that are free of outliers and, that contain most of the information in a data matrix X.

Proposed Algorithm:

Step 1: Determine the initial basic subset of $m>p$ observations by using *Version 1 or Version 2* approach of the *BACON* algorithm, where p is the dimension of the data and $m(=p.c)$ is an integer chosen by the data analyst. Generally c=*4 or 5* are suggested.

Step 2: Compute the discrepancies

$$d_i(\bar{x}_b, s_b) = \sqrt{(x_i - \bar{x}_b)' s_b^{-1} (x_i - \bar{x}_b)} \quad i=1,2,...,n.$$

$\bar{x}_b$ and s_b are the mean and the covariance matrices of the basic subset respectively.

Step 3: Determine the new basic subset consisting of observations satisfied $d_i(\bar{x}_b, s_b) < C_{npr} \cdot \chi_{p,\alpha/n}$. Generally, these are the observations with smallest discrepancies. $\chi^2_{p,\alpha}$; is the $1-\alpha$ percentile of the chi-square distribution with p degrees of freedom. $C_{npr} = C_{np} + C_{hr}$ is a correction factor, r is the size of the current basic subset, $C_{hr} = max\{0,(h-r)/(h+r)\}$, $C_{np} = 1 + \frac{p+1}{n-p} + \frac{1}{n-h-p}$ and $h = \left[\left[\frac{n+p+1}{2}\right]\right]$.

Step 4: Iterate step *2* and *3* until no longer changes on the size of the current basic subset.

Step 5: Nominate the observations excluded by the final basic subset as outliers.

Step 6: Determine the reduced data $X_{(I)}$ by eliminating outliers from the data.

Step 7: Calculate the eigenvalue and eigenvector pairs (λ_i, u_i) of variance covariance of $X_{(I)}$ matrix as in the form $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p > 0$.

Step 8: Determine the principle component matrix of $X_{(I)}$ as

$$y = X_{(I)} U = (y_1^*, y_2^*, ..., y_p^*)$$

where $U=(u_1, u_2, ..., u_p)$ is the eigenvectors matrix which corresponds to the variance-covariance matrix of $X_{(I)}$ and y_i^* is the *ith* column of y.

Proposed Graphical Method:

I propose some graphical displays, which will be helpful for identifying outlying observations visually.

Algorithm:

Step1: Calculate BACON Robust Principle Component distances (BRPCD) by using the following step

- Calculate the mean ($\bar{y}$) and the covariance matrix (s_y) of principle components of y obtained from *BRPCA*
- Calculate the principle components of X (original data matrix) denoted by x_i^*
- Calculate the robust distances by using the following equation

$$BRPCD_i = d_i = (x_i^* - \bar{y}).(s_y)^{-1}.(x_i^* - \bar{y})' \quad i = 1,2,...,n. \qquad (4.1)$$

Step 2: Draw the following graphs

1. Quantile-Quantile *(Q-Q)* plot of cube root of *BRPCD* or
2. Classical Mahalanobis distances *(MD)* versus *BRPCD* or
3. Index plot of *BRPCD*

The first graph produces a graphical display to test the distribution of data and determines the outlying observations. On the other hand the second and third graphs help only us to find outliers.

A normal distribution is often a reasonable model for the data. The most useful tool for assessing normality is a Q-Q plot. Q-Q plot is a scatter plot with the quantiles of the scores on the horizontal axis and the expected normal scores on the vertical axis. Generally Mahalanobis distances are used as quantiles. Curvature of the points indicates departures of normality. This plot is also useful for detecting outliers. The outliers appear as points that are far away from the overall pattern of points.

The plotting can be done either using the table of appropriate quantiles of the distribution (Wilk et al., 1962) or by the way of a Normalizing transformation; the later may be expected to work better as the number of degrees of freedom increases. For a large number of degrees of freedom, both the square root and cube root are used as normalizing transformations.

So I prefer to use cube root of distances and propose using BRPCD instead of classical Mahalanobis distances in the Q-Q plot.

I propose using the second plot, which displays BRPCA-based on robust distances versus Mahalanobis distances. This plot compares the BRPCA-based on mean and covariance matrices with the classical mean and covariance matrix. For comparing these parameter analyst has to draw the lines of $x = C_{npr} \cdot \chi_{p,\alpha/n}$ and $y = C_{npr} \cdot \chi_{p,\alpha/n}$ in this plot. These lines determine a rectangular, observations lie in the rectangle are regular, whereas the outliers lie outside of the rectangle. This plot becomes more useful in higher dimensions, where it is not so easy to visualize the data and the ellipsoids. If data were not contaminated (say, if all the data would come from a single bivariate normal distribution), then all points in this plot would lie together in the rectangular.

The third plot is also used for finding outliers. For determining outliers one has to draw horizontal line of $y = C_{npr} \cdot \chi_{p,\alpha/n}$, observations lie in the upper region of this line are considered as outliers.

Furthermore, if one wishes to know if there is any outlying observation after eliminating outliers from data, one can also examines Q-Q plots of the components of y.

4.3. Applications of the Proposed Method

I illustrate the computational efficiency of the proposed method using two data sets: The Hawkins, Bradu and Kass (*HBK*) data and the Philips data.

HBK data (Hawkins, et al., 1984) consists of *75* observations on three variables, which satisfy the linear regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i, \qquad \text{for all } i=1,2,...,75$$

where ε_i corresponds to the error term of *ith* observation. It is a contaminated data and known that the first 14 observations are outliers.

The Philips data consist of *677* observations on nine variables (diaphragm parts for TV sets which are thin metal plates, molded by press), which satisfy the linear regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_9 x_{i9} + \varepsilon_i, \qquad \text{for } all \ \ i=1,2,...,699$$

where ε_i corresponds to the error term of *ith* observation. It is a real data and found that the observations 491-565 are outliers (Rousseeuw and van Driessen, 1999).

4.3.1. Hawkins, Bradu and Kass (HBK) data

Hawkins et al. (1984) constructed this data for examining the extensive masking effect. It is known that the first ten observations are *xy*-space and the next four observations are *x*-space outliers.

Figure 4.1. shows the scores with respect to the first two components of PCA applied to all four variables x_1, x_2, x_3 and y. If I draw % 97.5 confidence ellipses of principle components in this plot, I can easily say that PCA methods determine only 4

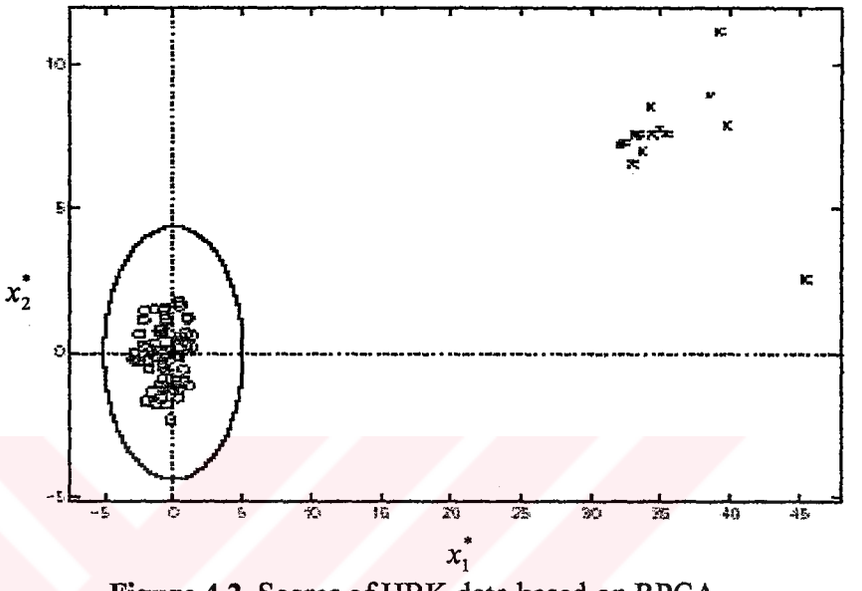

Figure 4.2. Scores of HBK data based on RPCA

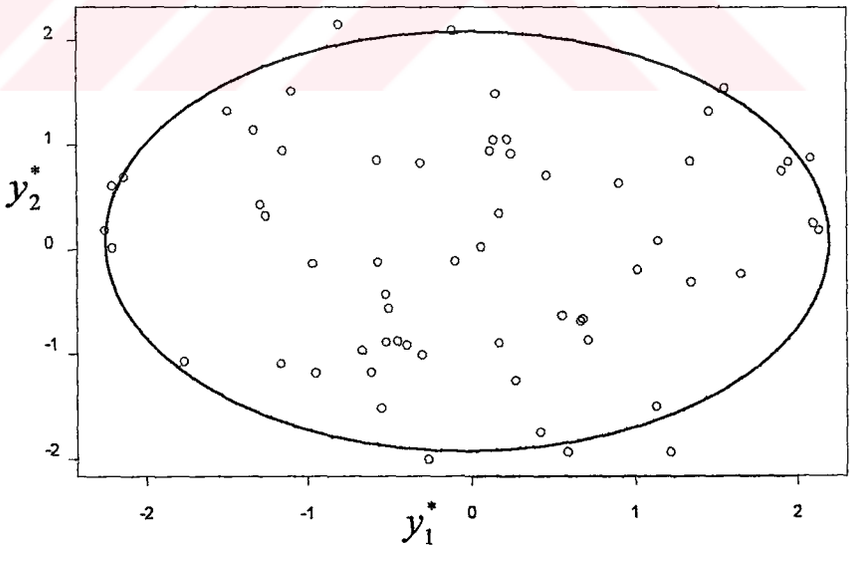

Figure 4.3. Scatter plot of the first two components of HBK obtained from the method of BRPCA

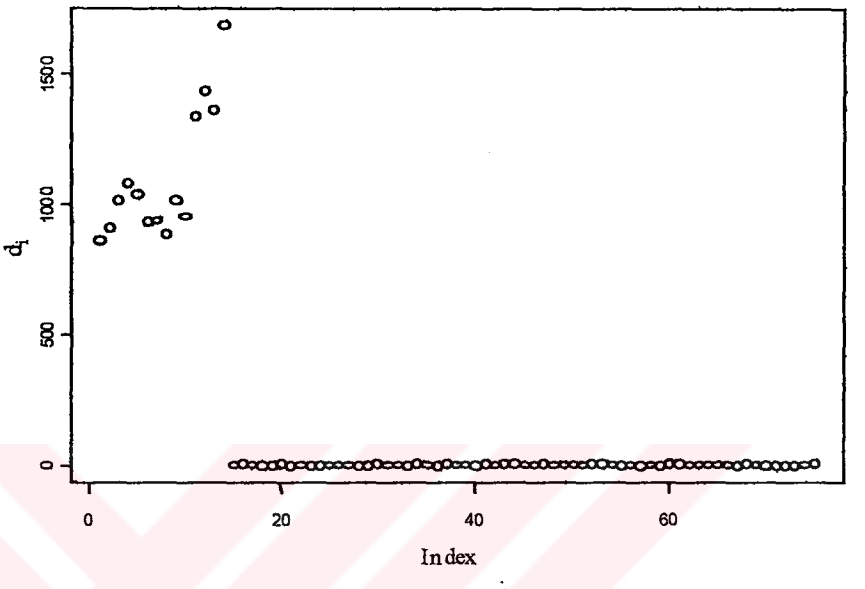

Figure 4.4. Index Plot of BRPCD for HBK data

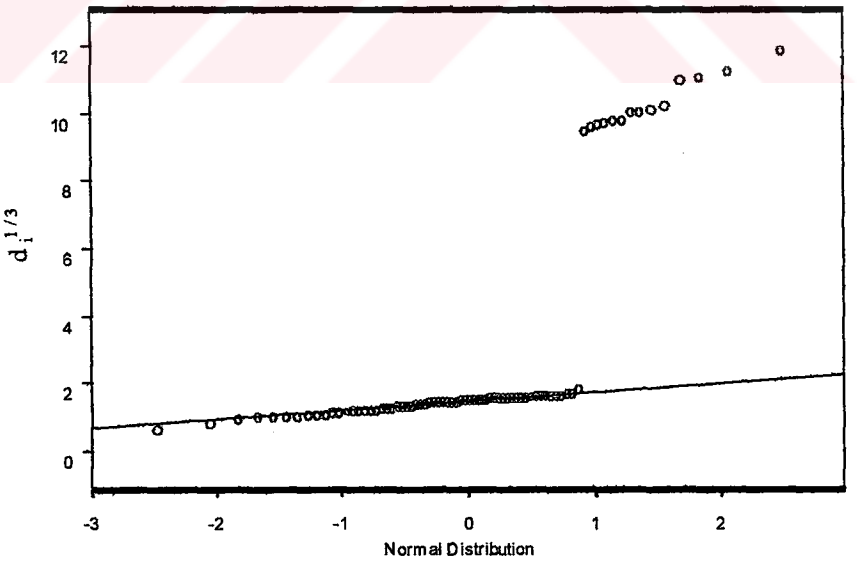

Figure 4.5. Q-Q plot of cube root of BRPCD for HBK data

Because BRPCA method calculates the principle components after eliminating the outliers from the data there is a little observation outside the 97.5% confidence ellipsoid in Figure 4.3. Index and Q-Q plots of BRPCD can separate outliers from the good data correctly, seen in Figures 4.4. and 4.5. Whereas as seen in Figures 4.6. and 4.7., the index and Q-Q plots of classic Mahalanobis distances of PCA cannot separate outliers, respectively.

Because PCA is based on classical estimators, it is sensitive to outliers so it will not provide satisfactory results in the existence of outliers. Index and Q-Q plots of the cube root of the classical Mahalanobis distances show this problem.

For comparing the performance of classical principle component analysis (PCA), robust principle component analysis (RAPCA) (Hubert et al., 2002) and BACON robust principle component analysis (BRPCA), I can also calculate the eigenvalues of variance covariance matrices of this data after applying these three methods, shown in Table 4.1. These eigenvalues (λ_i) correspond to the variance of the *ith* principal component. If all these values are equal to unity, the original X matrix columns are orthogonal, while if one λ_i is exactly equal to zero, this implies a perfect linear relationship between the original X matrix columns. One or more of the λ_i near zero implies that multicollinearity is present.

Table 4.1. Eigenvalues of the HBK data computed from PCA, RAPCA and BRPCA

	PCA	***RAPCA***	***BRPCA***
λ_1^2	223.12	3.47	1.326
λ_2^2	5.538	2.63	1.093
λ_3^2	1.688	2.47	0.956
λ_4^2	0.914	0.67	0.2957

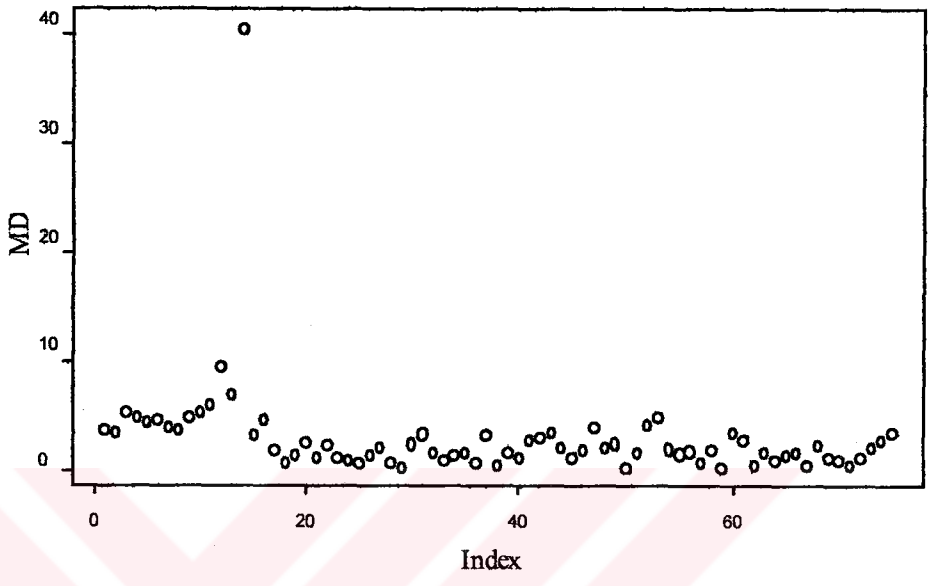

Figure 4.6. Index Plot of MD for HBK data

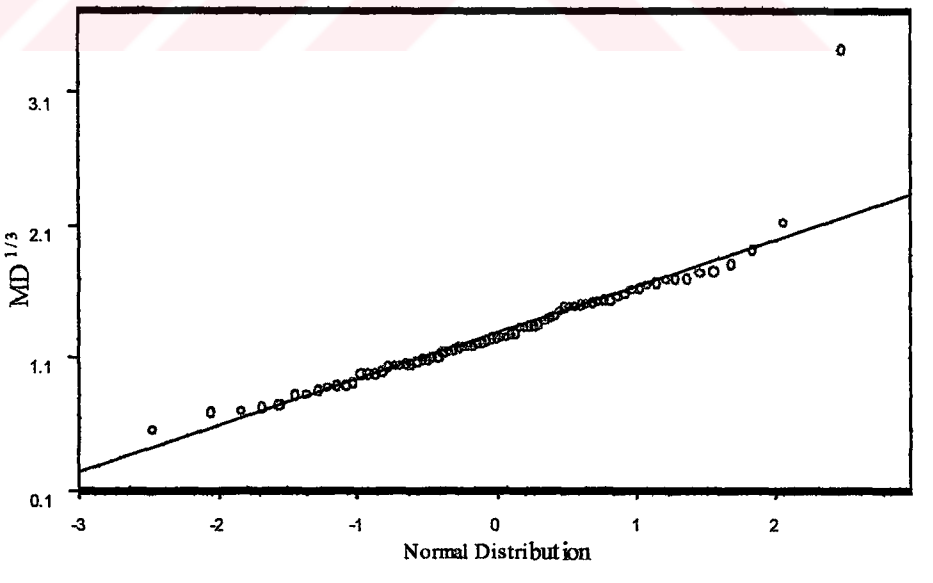

Figure 4.7. Q-Q plot of cube root of MD for HBK data

Because outliers are more dispersed from the majority of data, the eigenvalues computed from PCA are considerably large, which are seen in Table 4.1. On the other hand robust algorithms (RAPCA and BRPCA) result in small eigenvalues. But the BRPCA values are smaller than the values of RAPCA's. These results show that BRPCA method gives better results than the others since I work with the principal components that are free of outliers; therefore it is obvious that the reduction of the dimension of the data matrix can be accomplished more reliably.

4.3.2. Philips Data

Philips data set helps showing the performance of the proposed method on large data sets. This data set consists of *677* observations on nine variables.

Index plot and *Q-Q* plot of cube root of Mahalanobis distances are given in Figures 4.8. and 4.9, respectively. We can straightforwardly see that these plots do not display all outliers in the data. There are a lot of methods for identifying outliers, but unfortunately as I mentioned earlier most of they are ineffective in large data because of the computational problem.

The proposed method (*BRPCA*) is suitable for large data sets and has less computational cost. Moreover because of the BACON algorithm (V2) has a high breakdown point (upwards of *%40*), I expect this method to have a high breakdown point as well. Therefore it has more advantages than the other proposed methods.

Index plot and *Q-Q* plot of cube root of *BRPCD* are given in Figures 4.10. and 4.11., respectively. These plots give us different information from Figures 4.8. and 4.9. and display strongly deviating group of outliers, ranging from index 491 to index 565.

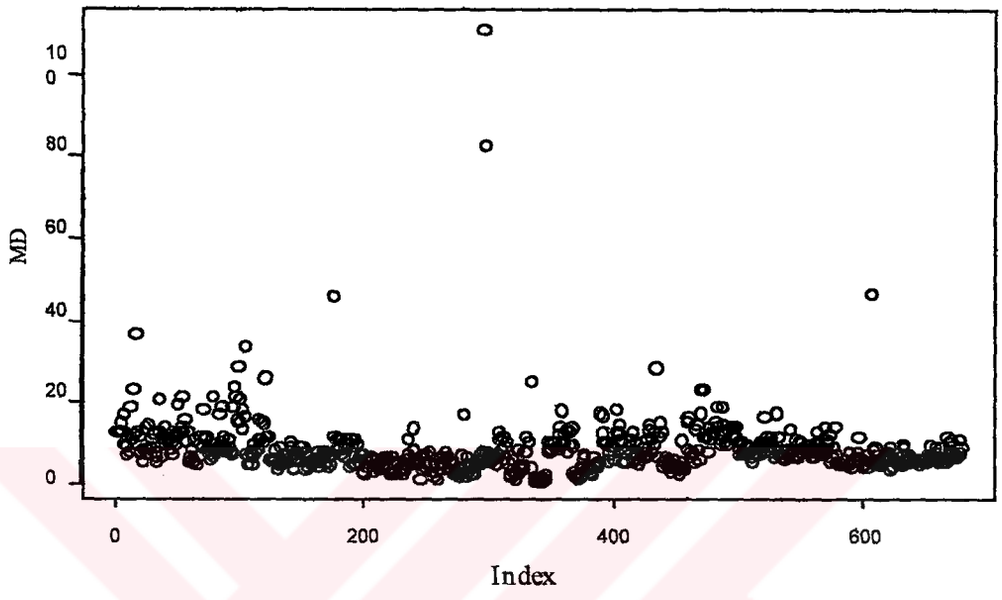

Figure 4.8. Index Plot of MD for Philips Data

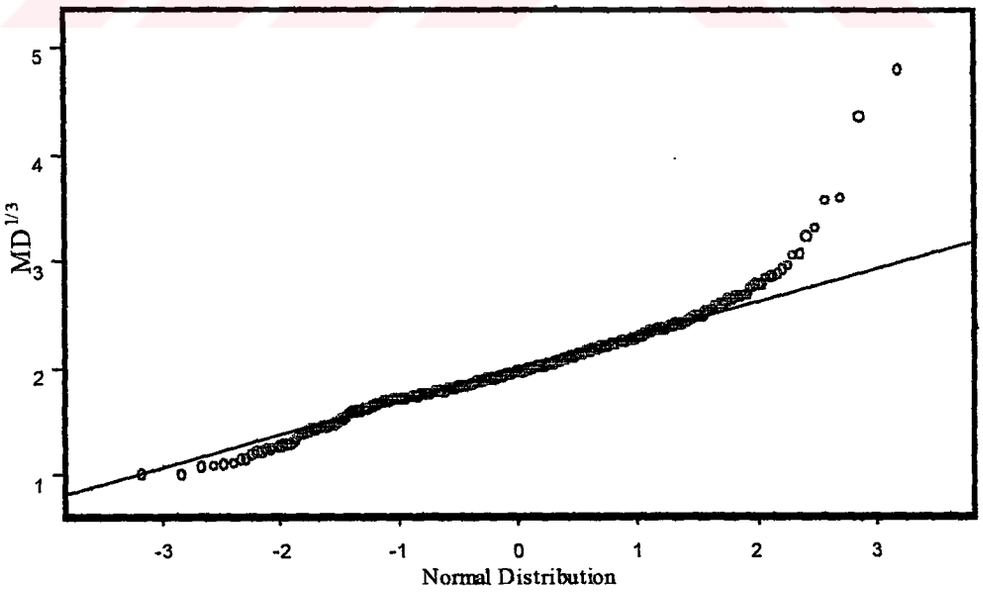

Figure 4.9. Q-Q plot of cube root of MD for Philips Data

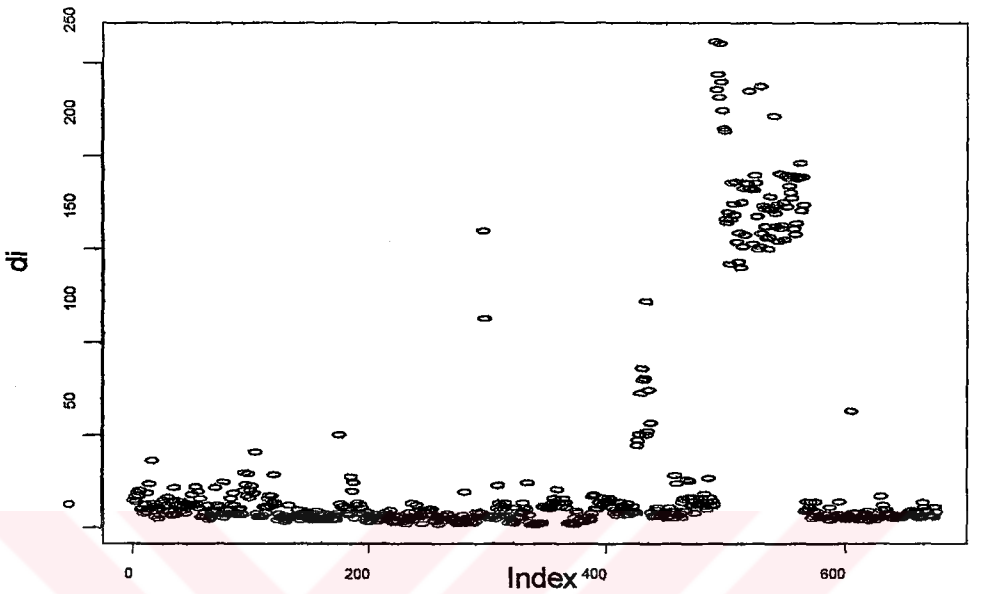

Figure 4.10. Index Plot of BRPCD for Philips Data

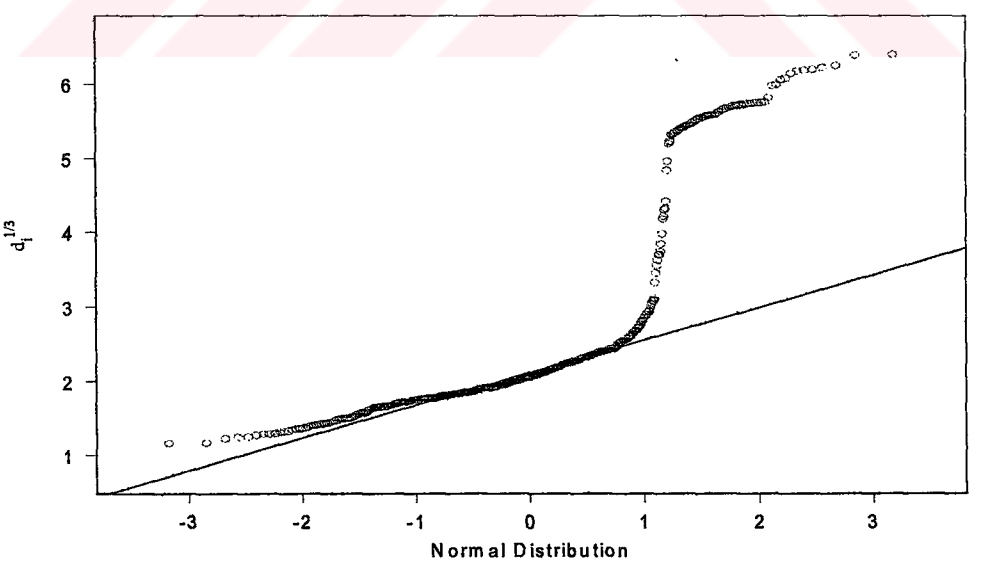

Figure 4.11. Q-Q plot of cube root of BRPCD for Philips Data

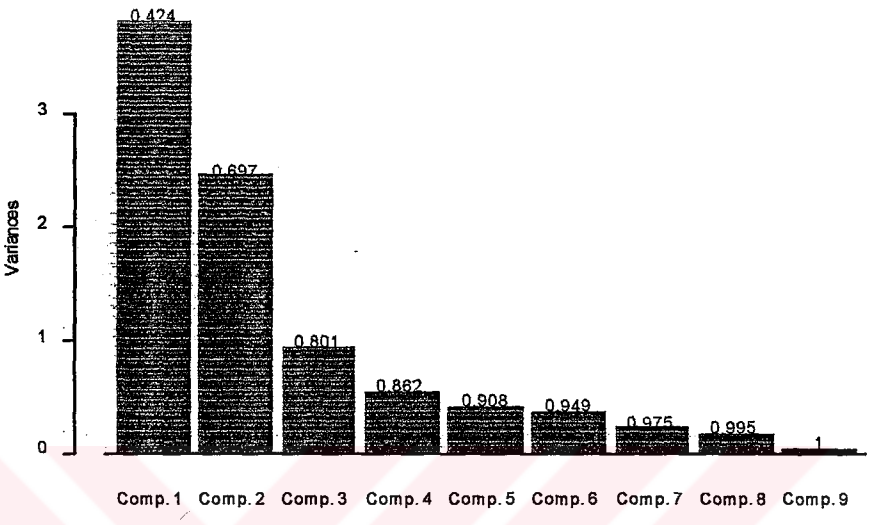

Figure 4.12. Scree plot of Philips data

It is known that the principle component analysis approach combats multicollinearity by using less than the full set of principal components in the model. To obtain the principal components of X matrix, columns are arranged in order of decreasing eigenvalues, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p > 0$. If we suppose the last s of these eigenvalues are approximately equal to zero then the principal components corresponding to near-zero eigenvalues are removed. That is we use only the first p-s components in the analysis.

Figure 4.12. shows the scree plot of the components of Philips data. Scree plot helps us to decide how many principal components should be preserved.

If we examine this plot, we see that the variances decrease nicely. After the third components there is a small change on the variances. So we conclude that the first three components are important in our analysis. We can easily find the same information if we look at the scree plot of the eigenvalues versus index.

In order to see the effect of outliers in determining significant components one can also draw the scree plot after eliminating the outliers from the data. For this eliminated data we obtain the same graph like Figure 4.12. Therefore I conclude that outliers are not effective on detecting of the important components in this data.

4.4. Simulation Study

In this study performance of the proposed method, *BRPCA,* is compared with *PCA* and *RAPCA* (Hubert, et al., 2001) by examining six different configurations.

The design of the experiments is as follows. The clean and contaminated data are generated from *p*-variate Gaussian distributions, $N(0,\Sigma)$ and $N(\mu^*,\Sigma^*)$, respectively. In order to assess the performance of the proposed method, the number of observations n was set to 100 and the number of variables p was set to 4 (low dimension) and 20 (high dimension), and two different contamination levels are taken, %0 (for scenario 1 and 4) and %10 (for scenario 2,3,5, and 6). The variance-covariance matrix of the clean data are defined $\Sigma=diag(4,3,2,1)$and $\Sigma=diag(5,4,3,2,1,0.15,0.14,...,0.02,0.01)$, used in *Tables 4.2.* and *4.3.*, respectively and contaminated data in scenarios (simulation) 2, 3, 5 and 6 are generated from $N(0, 9\Sigma)$, $N\left((0,0,0,10)',\Sigma\right)$, $N(10e_6, \Sigma)$, and $N(10e_5, (1/20)\Sigma)$, respectively, where e_i is the identity vector, whose *ith* element is *1* and the others are *0*.

The performance measures used in this study are given as follows:

- $MSE\left(\hat{\lambda}_i\right)=\frac{1}{100}\sum_{i=1}^{100}\left(\lambda_i-\tilde{\lambda}_i^{(j)}\right)^2$ values where λ_i is the eigenvalue and $\tilde{\lambda}_i^{(j)}$ is the estimated value of λ_i at the *jth* replication.
- Mean of eigenvalues

Table 4.2. Simulation Results for p=4

	Scenario1			Scenario2			Scenario3		
	PCA	BRPCA	RPCA	PCA	BRPCA	RPCA	PCA	BRPCA	RPCA
Mean $\hat{\lambda}_1$	4.190	4.187	4.23	8.197	4.565	5.43	10.278	4.162	5.01
Mean $\hat{\lambda}_2$	2.933	2.917	3.18	5.104	3.116	3.94	4.115	2.916	4.06
Mean $\hat{\lambda}_3$	1.915	1.909	2.11	3.225	2.029	2.67	2.879	1.867	3.25
Mean $\hat{\lambda}_4$	0.974	0.967	1.04	1.594	1.037	1.35	1.874	0.944	2.37
MSE $\hat{\lambda}_1$	0.352	0.356	0.39	20.38	0.811	2.61	39.905	0.327	1.38
MSE $\hat{\lambda}_2$	0.120	0.128	0.18	5.190	0.19836	1.20	1.507	0.154	1.36
MSE $\hat{\lambda}_3$	0.072	0.071	0.13	1.838	0.101	0.62	0.895	0.111	1.73
MSE $\hat{\lambda}_4$	0.017	0.017	0.03	0.477	0.040	0.19	0.821	0.038	2.00

Table 4.3. Simulation Results for p=20

	Scenario4			Scenario5			Scenario6		
	PCA	BRPCA	RPCA	PCA	BRPCA	RPCA	PCA	BRPCA	RPCA
Mean $\hat{\lambda}_1$	5.361	5.361	5.12	9.519	5.532	5.30	10.05	10.05	5.55
Mean $\hat{\lambda}_2$	3.963	3.962	3.83	5.312	4.045	4.07	4.882	4.881	4.65
Mean $\hat{\lambda}_3$	2.887	2.887	2.87	3.940	2.867	3.25	3.627	3.623	3.81
Mean $\hat{\lambda}_4$	1.933	1.933	1.92	2.826	1.910	2.51	2.545	2.542	3.12
Mean $\hat{\lambda}_5$	0.953	0.952	0.96	1.875	0.952	1.78	1.695	1.695	2.24
MSE $\hat{\lambda}_1$	0.602	0.603	0.53	20.54	0.982	0.55	25.59	25.60	0.70
MSE $\hat{\lambda}_2$	0.179	0.179	0.25	2.074	0.293	0.21	1.076	1.070	0.64
MSE $\hat{\lambda}_3$	0.134	0.134	0.15	1.100	0.200	0.24	0.599	0.598	0.81
MSE $\hat{\lambda}_4$	0.063	0.063	0.12	0.793	0.104	0.41	0.401	0.397	1.43
MSE $\hat{\lambda}_5$	0.023	0.023	0.03	0.846	0.041	0.74	0.548	0.548	1.71

Tables 4.2. and 4.3. contain simulation results for p=4 and 20, respectively. For the case of $p=4$ I give the simulation results for all eigenvalues and eigenvectors in Table 4.2., but for the case of $p=20$ I only give the simulation results for the first five largest eigenvalues and corresponding eigenvectors in Table 4.3. for convenience.

Conclusions:

In Table 4.2. and Table 4.3., Mean $\hat{\lambda}_i$ and MSE $\hat{\lambda}_i$ values are displayed for scenarios 1-6. Ideally I wish to have small Mean $\hat{\lambda}_i$ and MSE $\hat{\lambda}_i$ values.

For the scenario 1 the performance measures of PCA and BRPCA methods are small and give almost the similar conclusions. On the other hand the numerical measures for RPCA method are larger than those of PCA and BRPCA for this specific case where there are no outliers. So I conclude that RAPCA does not perform well when there are no outliers. However somewhat PCA performs well for this case, but yet yields unreliable estimates in the presence of contamination (for the scenarios 2 and 3) as reported in Table 4.2. It is clear from the numerical performance results reported in Table 4.2. that BRPCA perform well for all levels of contamination for the case of $p=4$.

It is also known that the method of PCA is ineffective in the existence of outliers, which is immediately seen from Table 4.3., which contains the results for the scenarios 4, 5, and 6. From this table I conclude that the method PCA does not perform well for most of the cases in Table 4.3. Since I know the method PCA does not perform well in the presence of contamination I can examine the performance of the methods of RAPCA and BRPCA in the presence of contamination, corresponds to the scenarios 2, 3, 5, and 6.

In scenario 2 and 3 the performance measures for BRPCA and RAPCA are reasonably small, but the values for BRPCA method values are slightly smaller than those of RAPCA.

Therefore I conclude that for all scenarios 1, 2, and 3 BRPCA is superior to the other methods.

On the other hand in scenario 4, PCA and BRPCA methods provide almost the same results. This means that if there is no contamination and the number of parameters is high, these two methods produce the same performance. RAPCA values are almost the same with the others for Mean $\hat{\lambda}_i$ values. But MSE $\hat{\lambda}_i$ values of RAPCA are slightly bigger than the others. So in this scenario RPCA method does not perform as good as the other methods.

For the scenario 5, BRPCA performs better than RAPCA for most of the components. So again in this scenario I conclude that BRPCA mostly performs better than RAPCA.

In scenario 6, the performance measures of BRPCA show that this method is more sensitive to outliers in the first significant component. But for the other components the BRPCA performs better than RAPCA.

So I can say confidently that BRPCA performs better than PCA and RAPCA as p and contamination level increase.

4.5. Discussion and Conclusion

In this study I suggest the *BRPCA* method to obtain robust principal components. I show that this method performs well on different configurations designed in a simulation study. This method requires less computational time than *RAPCA* and *PCA* and has superior performance on large simulated challenges. Furthermore it is not affected by masking and swamping problems therefore it works better than *RAPCA* and *PCA*.

The codes of *BRPCA* method were written by S-Plus for Windows, Version 6.0. The code of *BRPCA* method is provided in *Appendix B*.

5. EXAMPLES FOR REGRESSION DATA

I will use four typical data sets to illustrate the applicability and the performance of the capability of detecting outliers of some of the methods given in the previous section

- Hawkins Bradu and Kass (HBK) Data (Hawkins et al., 1984),
- Wood Gravity Data (Rousseeuw and Leroy, 1987),
- Stackloss Data (Brownlee, 1965),
- Buxton Data (Buxton, 1920).

Table 5.1. Description of Classic Multiple Outlier Data

No	*Data Name*	*p*	*n*	*% of Outlying Observations*	*% of x-space Outliers*	*% of y-space Outliers*	*% of xy-space Outliers*
1	*HBK Data*	4	75	18.6	5.3	0.0	13.3
2	*Wood Gravity Data*	6	20	20.0	0.0	0.0	20.0
3	*Stackloss Data*	4	21	24.0	0.0	0.0	24.0
4	*Buxton Data*	5	87	5.74	5.74	0.0	0

5.1. Hawkins, Bradu and Kass Data

Hawkins, Bradu and Kass (1984) generate this data set for illustrating some of the merits of a robust technique. The data set consists of 75 observations in four dimensions (one response and three explanatory variables). The first 10 observations are bad leverage points, and the next four points are good leverage points.

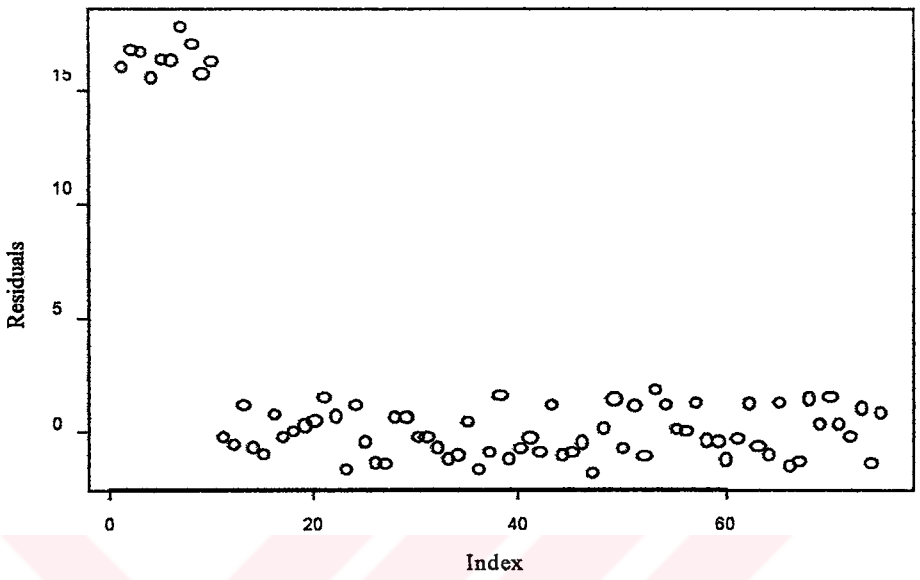

Figure 5.1. HBK data: Index plot of residuals obtained from the method of Hadi and Simonoff (1993)

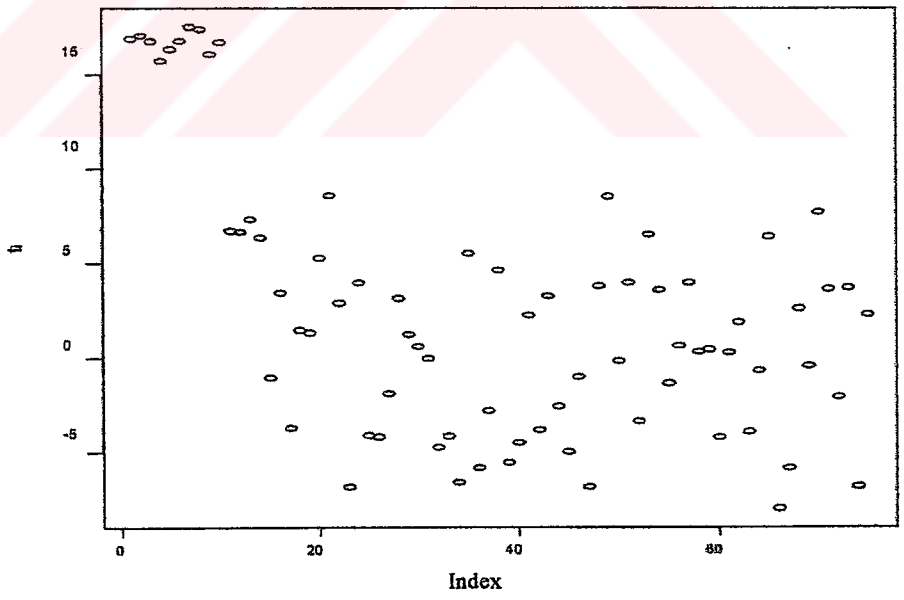

Figure 5.2. HBK Data: Index plot of residuals obtained from the method of BACON Regression

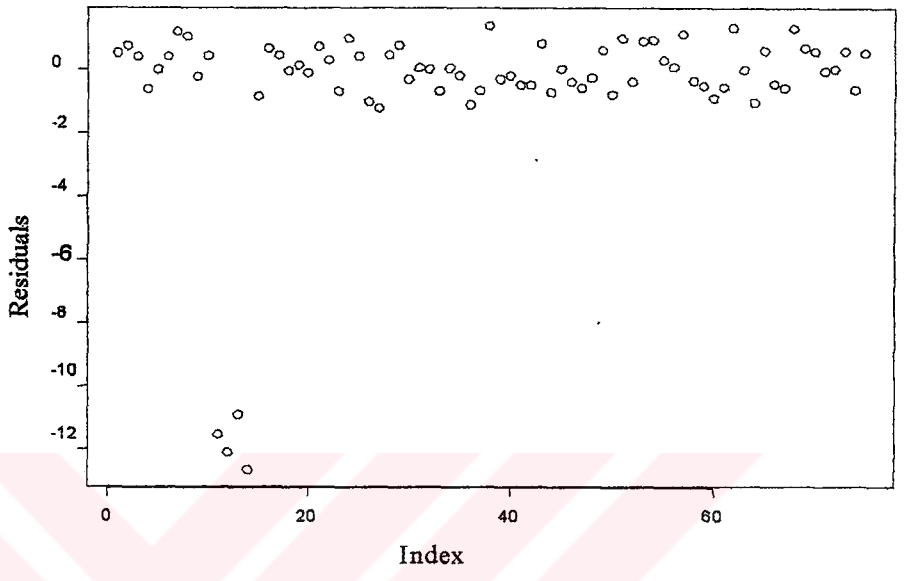

Figure 5.3. HBK Data: Index plot of residuals obtained from the method of Chatterjee and Mächler

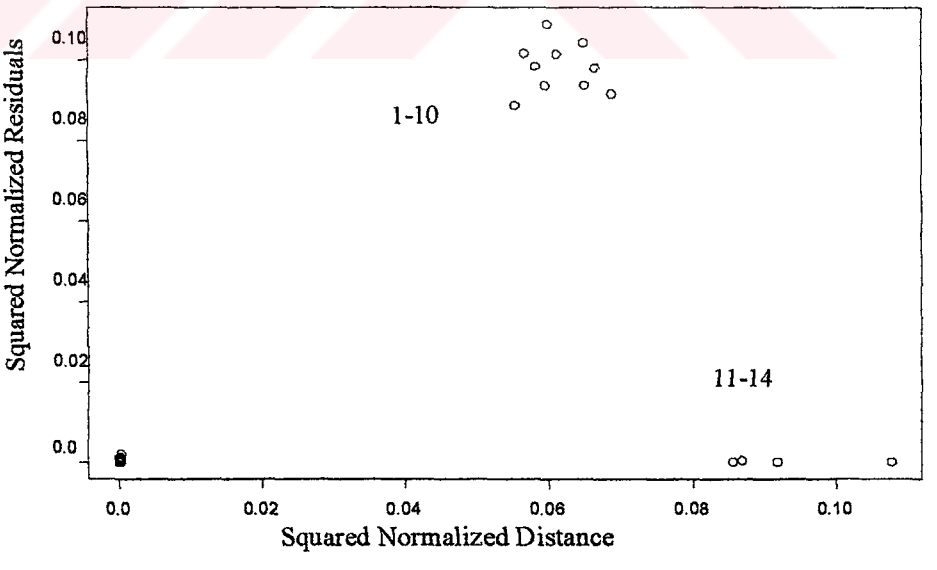

Figure 5.4. HBK data: Scatter plot of squared normalized residuals versus squared normalized robust distances obtained from the method of RWLS

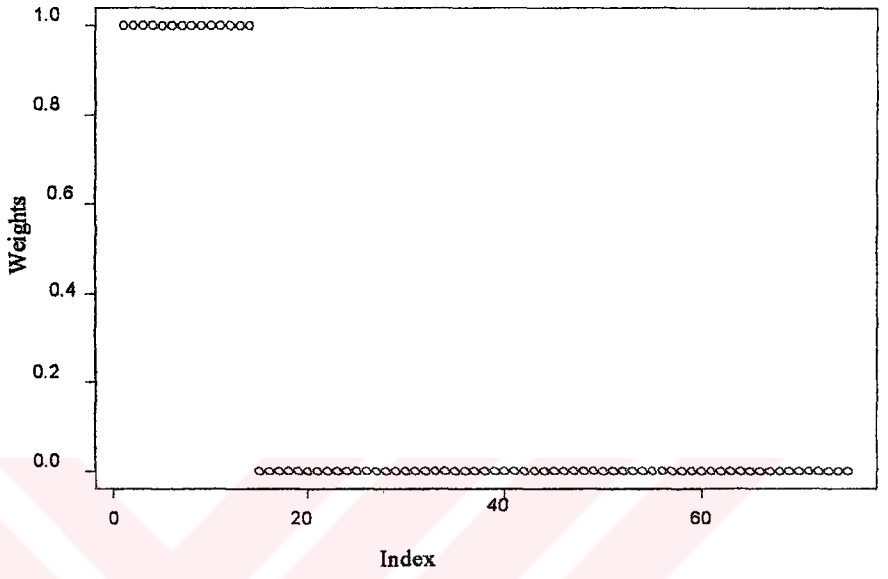

Figure 5.5. HBK data: Index plot of weights obtained from the method of Sebert

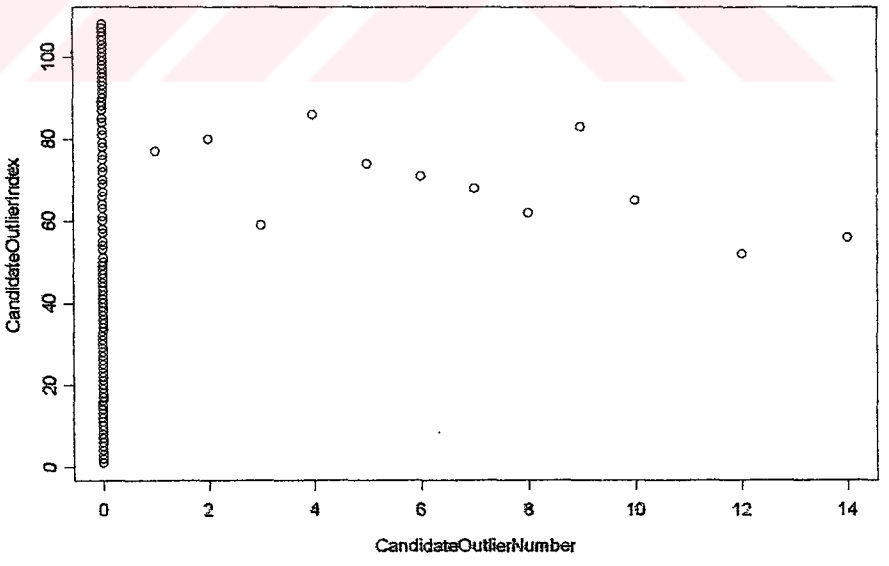

Figure 5.6. HBK data: Index plot of candidate outliers obtained from the method of Pena and Yohai

From the index plots (Figure 5.1. and Figure 5.2.) for the methods of Hadi and Simonoff (1993) and BACON regression, I can easily say that, observations 1-10 are outliers. On the other hand Chatterjee and Mächler method detects only observations 11-14 as outliers (Figure 5.3). RWLS and Sebert's methods detect all the outliers so they have good performance for this data set, which are seen in Figure 5.4. and Figure 5.5., respectively. On the other hand, the method of Pena and Yohai identifies all the outliers except observations 11 and 13 seen in Figure 5.6.

5.2. Stackloss data (*Brownlee*, 1965)

Stackloss data describe the operation of a plant for oxidation of ammonia to nitric acid and consist of 21 four-dimensional observations listed in Table A.2. The stackloss (y) has to be explained by the rate of operation (x_1); the nitric oxides produced that are absorbed in a countercurrent absorption tower; x_2; the inlet temperature of cooling water circulating through coils in this tower and x_3; proportional to the concentration of acid in the tower. Small values of the response correspond to efficient absorption of the nitric oxides.

Summarizing the findings cited in the literature, it could be said that most of the researchers concluded that observations 1, 3, 4, and 21 were outliers. According to some researchers observation 2 is reported as an outlier, too.

It is a real data and has been examined by a great number of statisticians by means of several methods.

The index plots associated with the methods of Hadi and Simonoff (1993), Sebert, BACON regression and residual plot associated with the RWLS method (Figure 5.7., Figure 5.8., Figure 5.11. and Figure 5.9.), respectively show that the observations 1-4, and 21 are the most outlying. However, Chatterjee and Mächler method is only able to find observations 1, 3, 4, and 21 as outliers, which is seen in Figure 5.10.

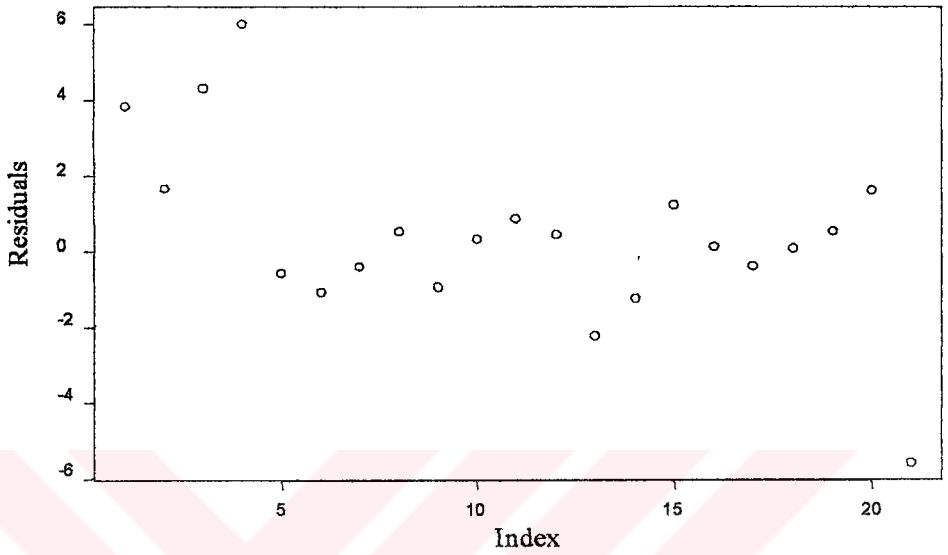

Figure 5.7. Stackloss data: Index plot of residuals obtained from the method of Hadi and Simonoff

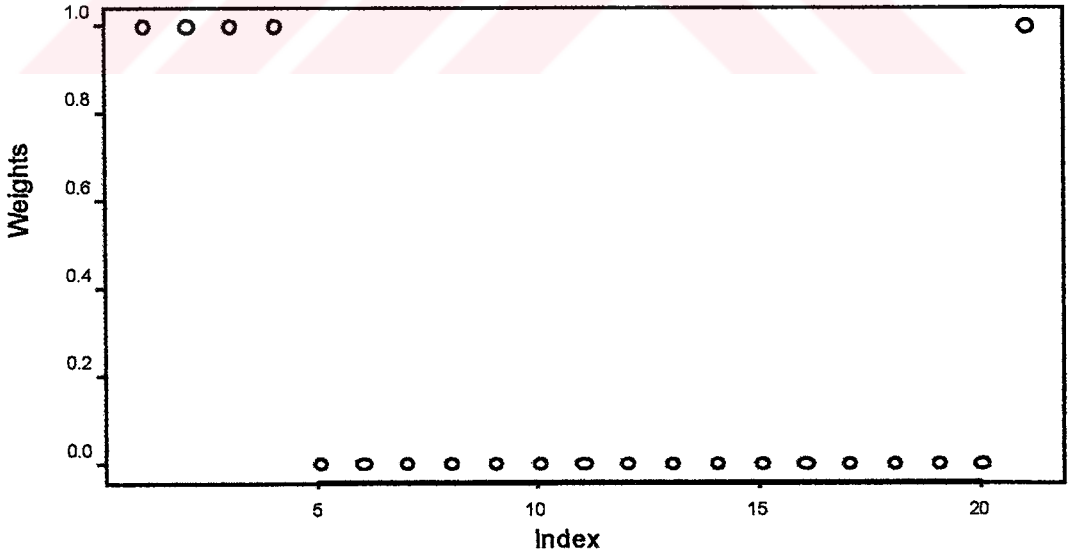

Figure 5.8. Stackloss data: Index plot of weights obtained from the method of Sebert

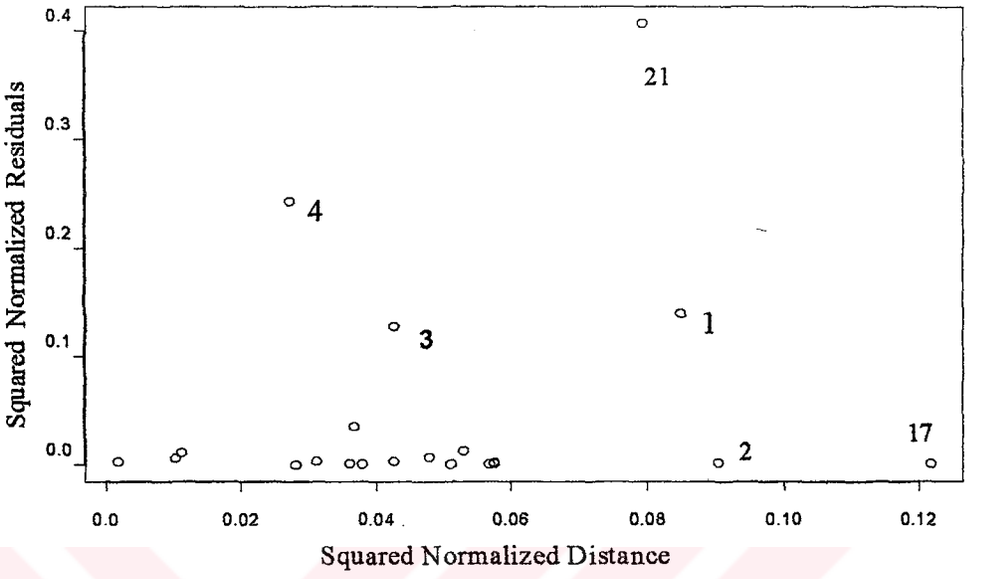

Figure 5.9. Stackloss data: Scatter plot of squared normalized residuals versus squared normalized robust distance obtained from the method of RWLS

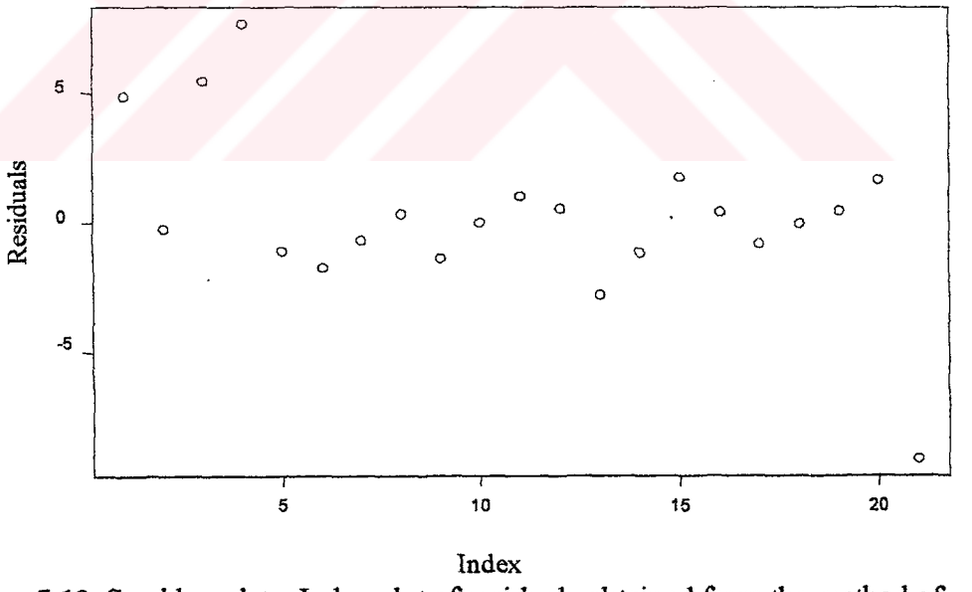

Figure 5.10. Stackloss data: Index plot of residuals obtained from the method of Chatterjee and Mächler

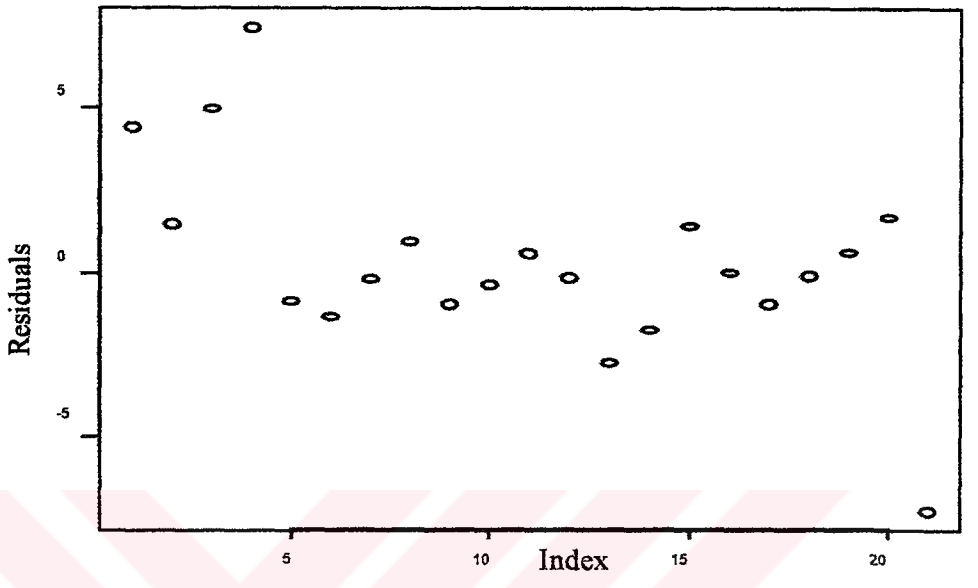

Figure 5.11. Stackloss data: Index Plot of residuals obtained from the method of BACON Regression

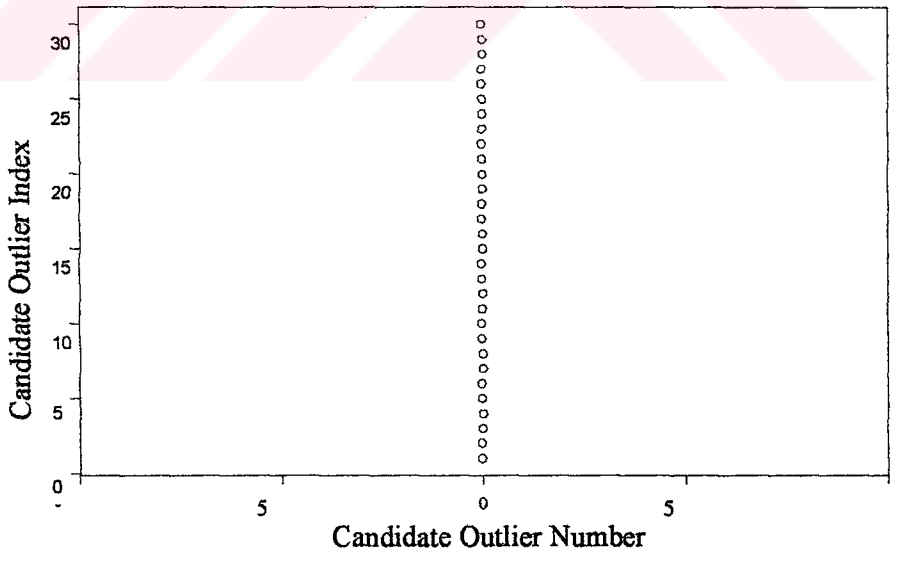

Figure 5.12. Stackloss data: Scatter plot of observation versus outlier observation numbers obtained from the method of Pena and Yohai

On the other hand, the method of Pena and Yohai does not detect any outliers in this data set, as seen in Figure 5.12. Therefore it does not have good performance for this data set.

5.3. Modified Woodgravity data

Modified Woodgravity data is a real data, which is rather well behaved and contaminated by replacing a few observations. This data set consists of 20 points with six parameters to be estimated. The raw data set was originally from Draper and Smith (1966, p. 227) and used to determine the influence of anatomical factors on Wood Specific Gravity, with five explanatory variables and an intercept. (Draper and Smith conclude that x_2 could be deleted from the model, but this matter is not considered for the present purpose).

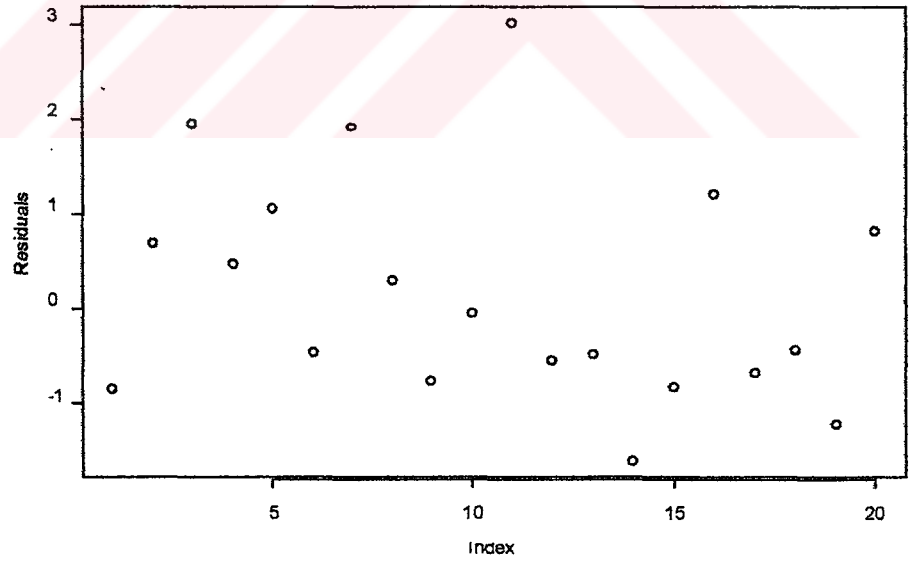

Figure 5.13. Woodgravity data: Index plot of residuals obtained from the method of Hadi and Simonoff

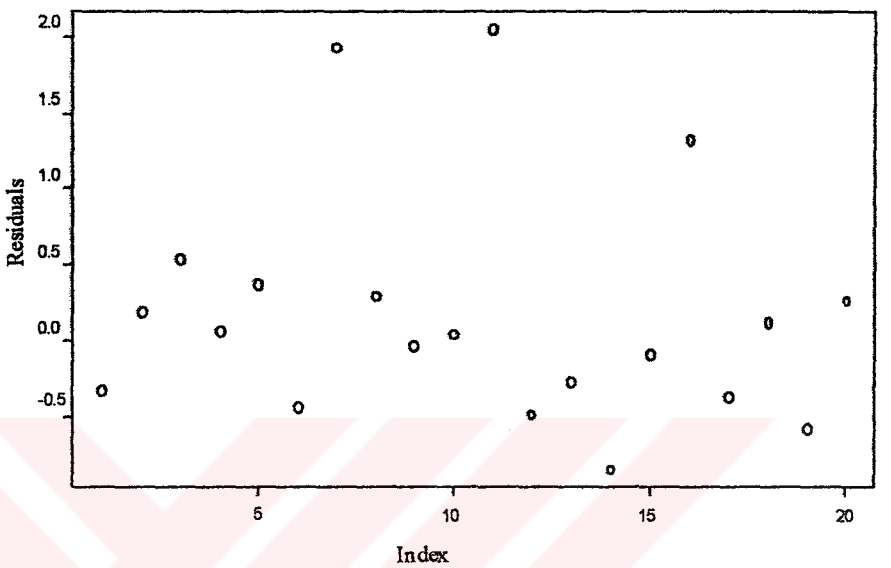

Figure 5.14. Woodgravity data: Index plot of residuals obtained from the method of Chatterjee and Mächler

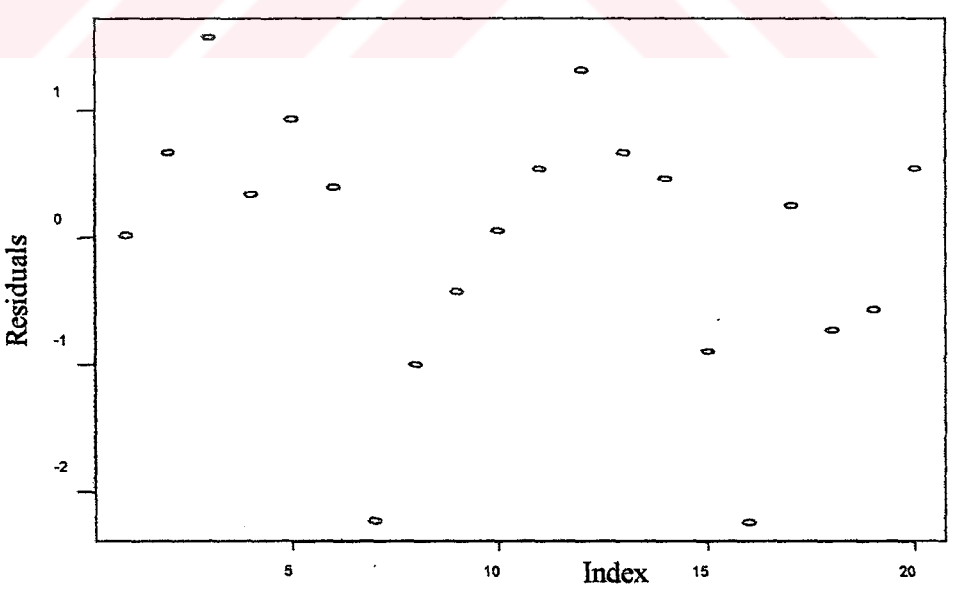

Figure 5.15. Woodgravity data: Index plot of residuals obtained from the method of BACON Regression

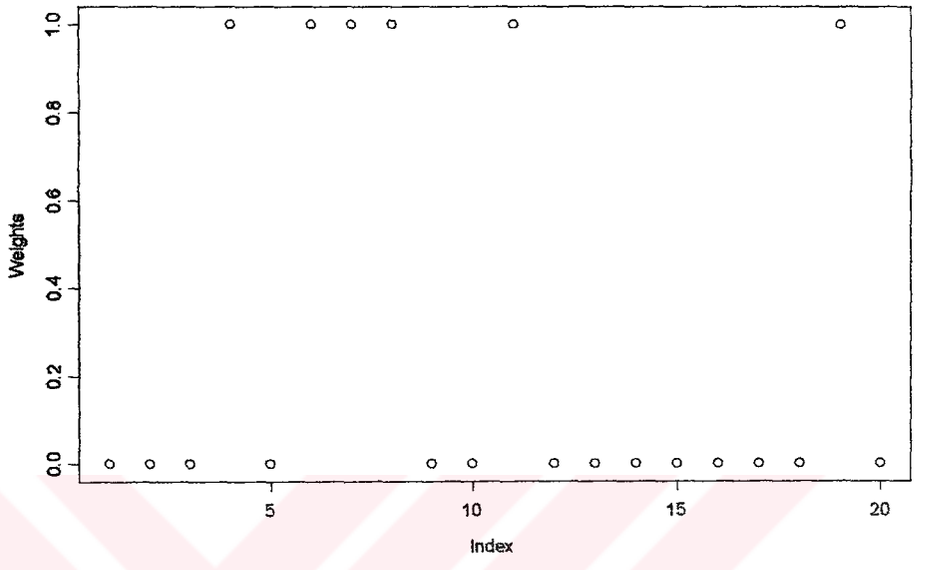

Figure 5.16. Woodgravity data: Index plot of weights obtained from the method of Sebert

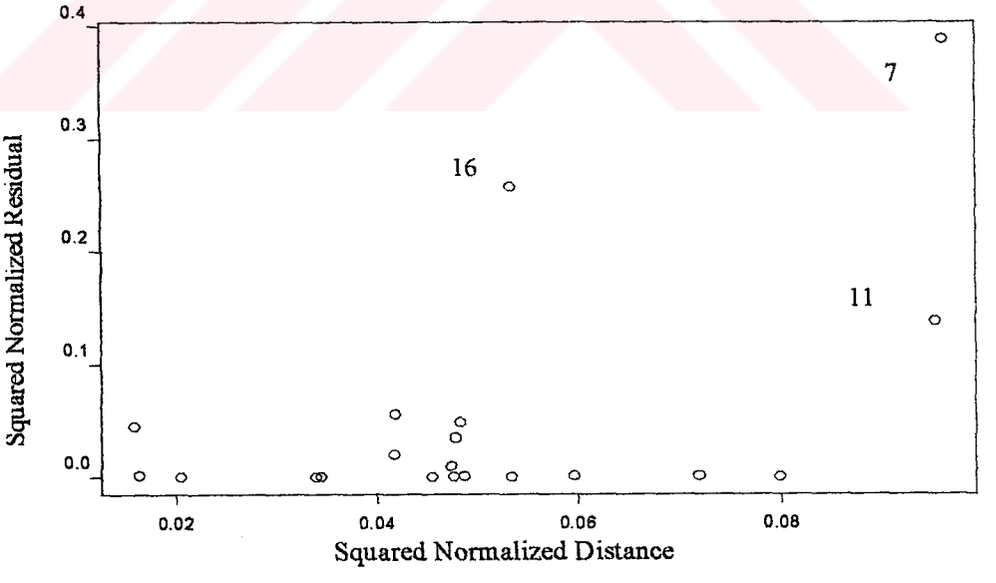

Figure 5.17. Woodgravity data: Scatter plot of squared normalized residuals versus squared normalized robust distance obtained from the method of RWLS

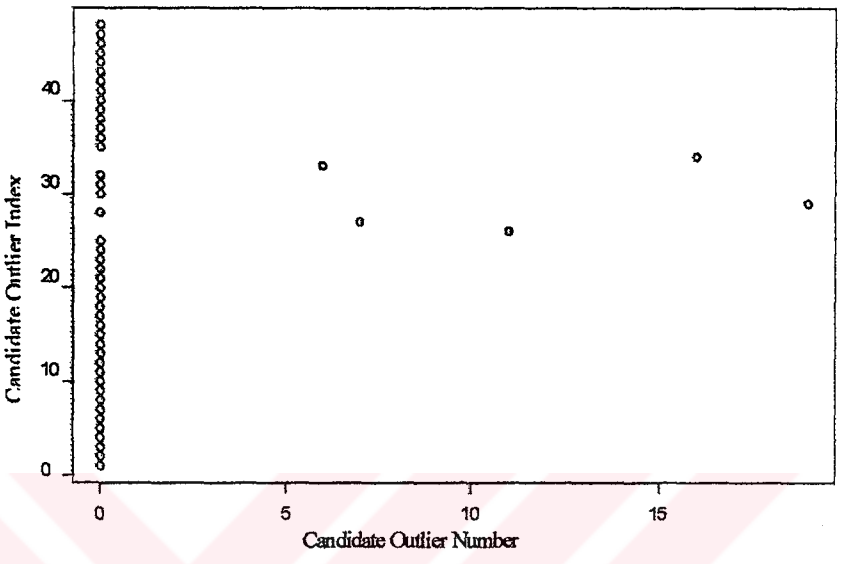

Figure 5.18. Woodgravity data: Scatter plot of observation versus outlier observation number obtained from the method of Pena and Yohai

Table A.4. lists a contaminated version of these data, in which a few observations have been replaced by outliers. According to some statisticians, observations 4, 6, 8, and 19 are reported as outliers.

Hadi and Simonoff, Chatterjee and Mächler, BACON regression methods are not able to detect correct outliers for this data set, which is seen in Figures 5.13., 5.14., and 5.15., respectively. On the other hand, Sebert's method finds all the CORRECT outliers. But as seen from the results of Woodgravity data set, this method is sensitive to swamping. Because even though the observations 7 and 11 are clean observations, this method finds these observations as outliers, which is seen from Figure 5.16. Beside of this RWLS and Pena and Yohai methods are sensitive to masking and swamping problem for this type of data, which is seen in Figure 5.17. and Figure 5.18., respectively. So it is clear that these methods are ineffective on detecting of outliers for this type of data.

5.4. Buxton data

Buxton data (Buxton, 1920), which is recently used by Hawkins and Olive (2002), consist of 88 observations on 20 variables, which satisfy the linear regression model given by

$$y_i = \beta_0 + \beta_1 x_{i1} + ..\beta_{20} x_{i20} + \varepsilon_i, \qquad \text{for all } i=1,2,...,88$$

where ε_i corresponds to the error term of *ith* observation.

Because I want to concentrate on the stature of men, I choose only 4 parameters, which are head length, nasal height, bigonal breadth, and cephalic index as explanatory variables and height of men as a response variable ($n=88$).

In this data observation 9 was deleted because it had missing values. Five individuals, number 61-65, were reported to be about 0.75 inches tall with head lengths well over 5 feets! So it is known that these observations are outliers.

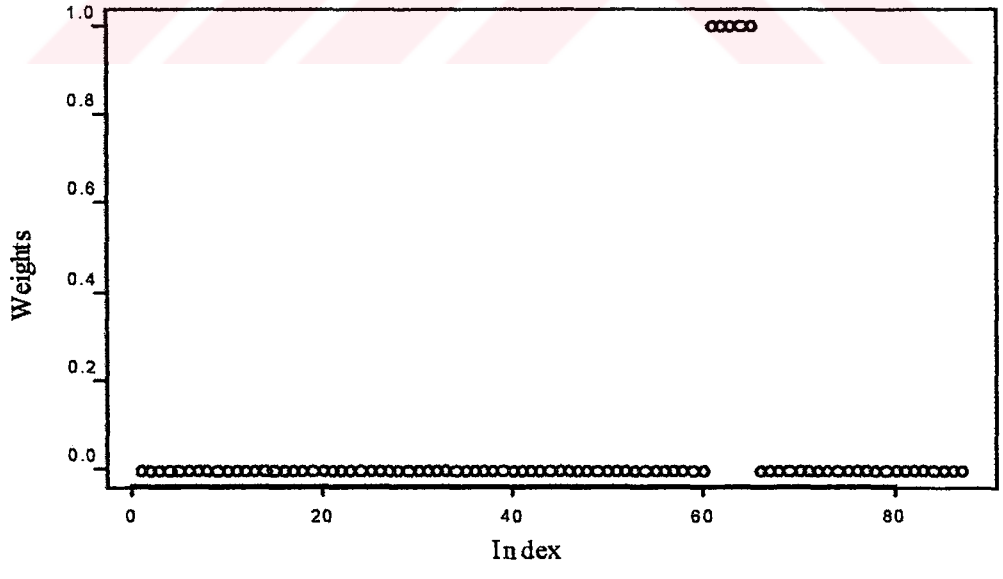

Figure 5.19. Buxton data: Index plot of weights obtained from the method of Sebert

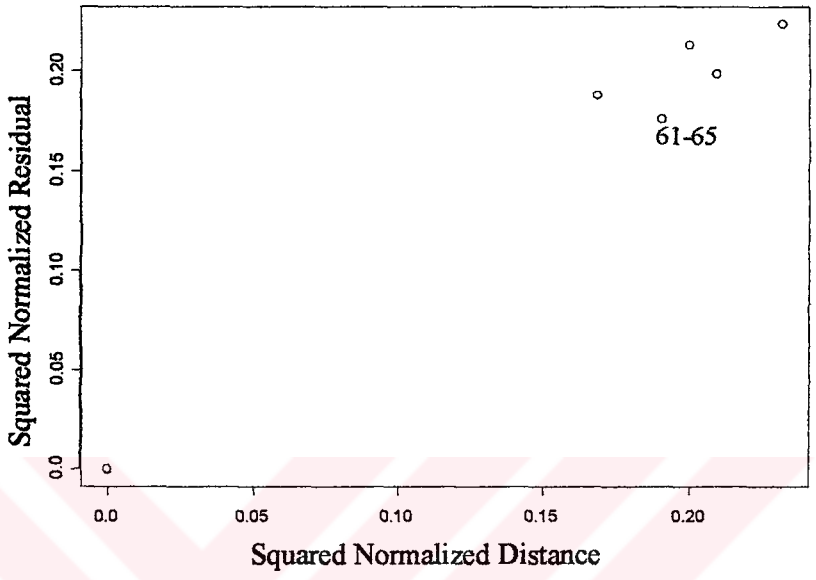

Figure 5.20. Buxton Data: Scatter plot of squared normalized residuals versus squared normalized robust distance obtained from the method of RWLS

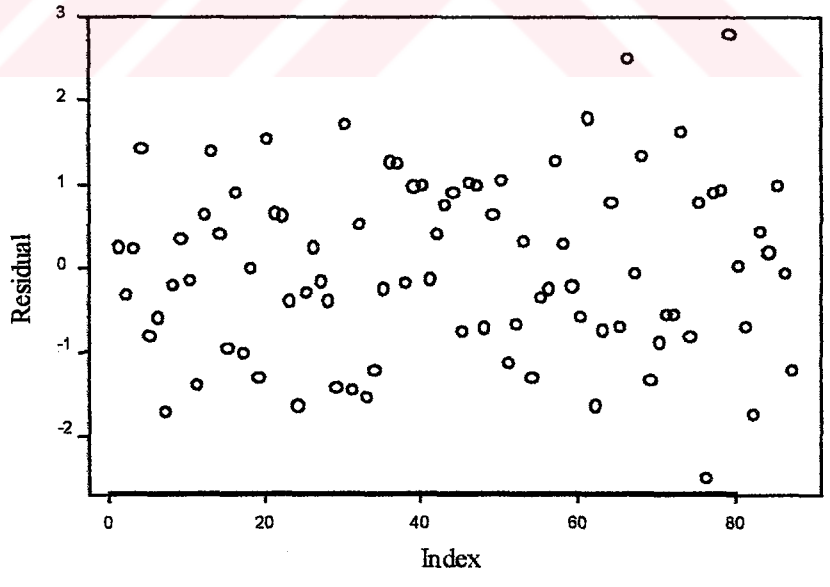

Figure 5.21. Buxton Data: Index plot of residuals obtained from the method of Hadi and Simonoff

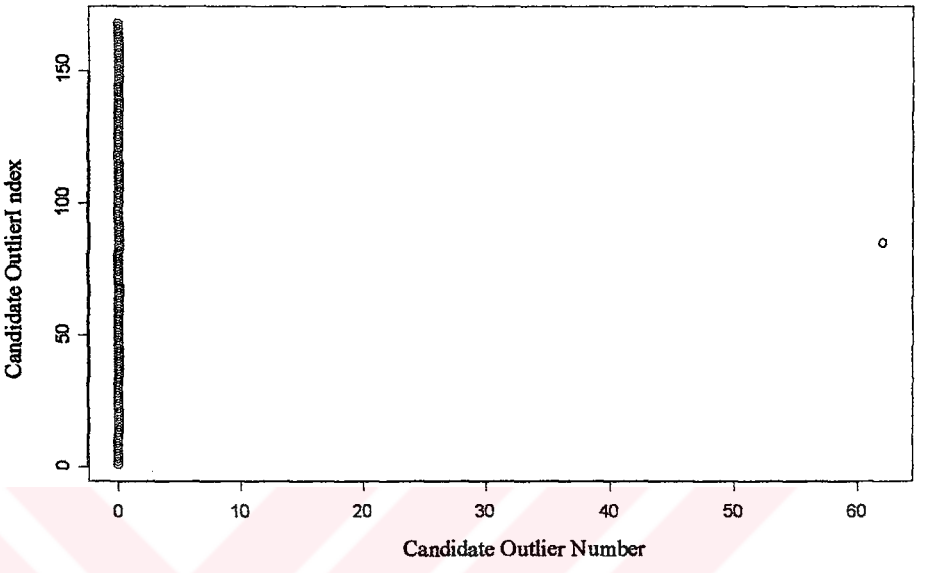

Figure 5.22. Buxton Data: Scatter plot of observation versus outlier observation number obtained from the method of Pena and Yohai

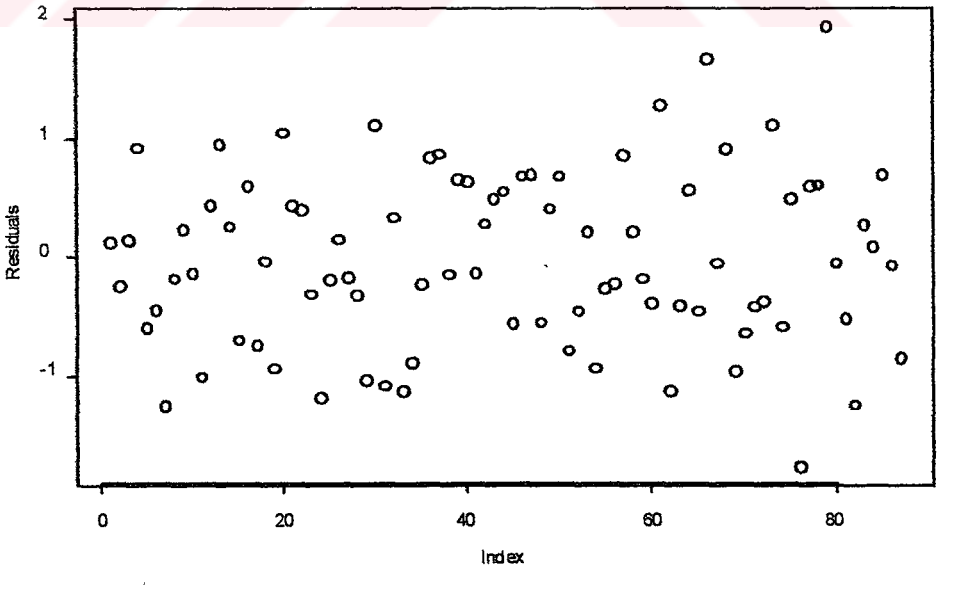

Figure 5.23. Buxton Data: Index plot of residuals obtained from the method of Chatterjee and Mächler

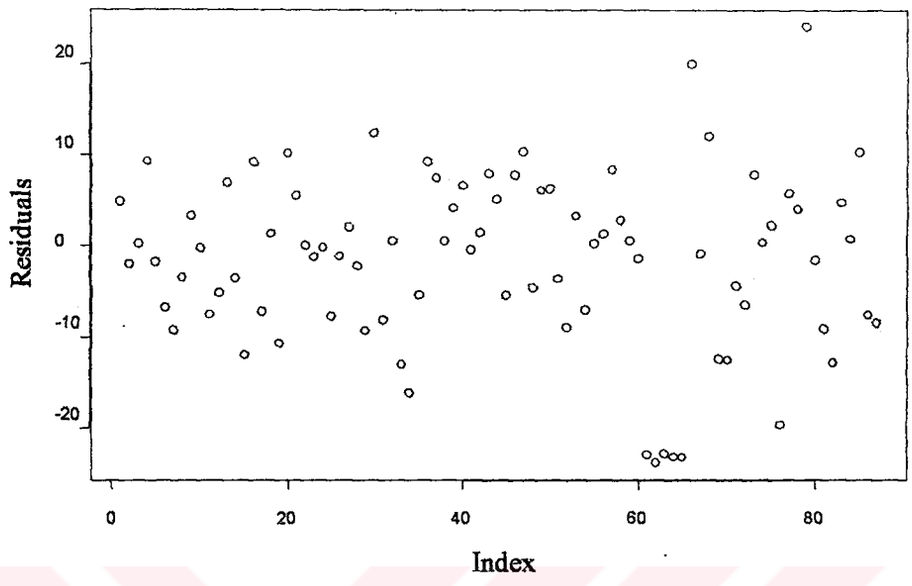

Figure 5.24. Buxton Data: Index plot of residuals obtained from the method of BACON Regression

The residual plot (Figure 5.19.) obtained from the method of Sebert and residual plot (Figure 5.20.) obtained from the RWLS method display that observations 61-65 are outliers which influence the regression model the most. These are the observations which have high leverage and large residuals.

On the other hand, the methods of Hadi and Simonoff, Pena and Yohai, Chatterjee and Mächler, and BACON regression do not detect these outliers. They detect many other observations as outliers as seen in Figure 5.21., Figure 5.22., Figure 5.23., Figure 5.24., respectively although they are not outliers (swamping problem). Hawkins and Olive (2002) applied some high breakdown methods such as LMS and LTS to this data set but these methods also resulted in detecting similar observations as outliers as the methods of Hadi and Simonoff, Pena and Yohai, Chatterjee and Mächler, and BACON regression. This is a warning that even using the objective of high breakdown will not necessarily protect one from extremely aberrant data.

Conclusions:

The results obtained by examining Figures 5.1–5.24. are summarized in Table 5.2. as in the following:

- The methods of Hadi and Simonoff (1993) and BACON regression (2000) give almost the same performance for all data sets. Both procedures seem to be less sensitive to multiple outliers, but both of them fail to identify outliers in the data sets of Woodgravity and Buxton.
- The method of Pena and Yohai (1995) does not perform well on the Buxton and Stackloss data sets and it is sensitive to masking and swamping problems as seen in the HBK and Woodgravity data sets.
- The Chatterjee and Mächler (1997) method is ineffective on detecting outliers for all of the data sets and it is sensitive to masking.
- The method of Sebert is successful on detecting of outliers for all the data sets but sensitive to swamping as the case in Woodgravity data.
- The RWLS method identifies all of the correct outliers for these data sets except Woodgravity data.

In this section I attempt to illustrate the performances of some of the methods, which I think that, are practical and used very often in the literature.

From the results I obtained in this section it is clear that the performances of the methods vary depending on the characteristics of data sets. Unfortunately there is not one method that works the best for every type of data sets. However what I conclude from this exploration is that there are some selective methods, which may work well for the most of the data sets.

In the following section I will give an extensive simulation study by considering many different types of configurations of the data sets to assess the performances of the outlier detection methods used in this section.

Table 5.2. Performances of some methods on Classic Multiple Outlier Data Sets

	True Outliers	Hadi Simonoff (1993) Method 1	Sebert (1998)	Pena and Yohai (1995)	RWLS (2001)	BACON Regression (2000)	Chatterjee and Mächler (1997)
HBK data	1-14	1-10	1-14	1-10,12,14	1-14	1-10	11-14
Buxton Data	61-65	-	61-65	62	61-65	-	-
Stackloss Data	1-4,21	1-4,21	1-4,21	-	1-4,17,21	1-4,21	1,3,4,21
Woodgravity Data	4,6,8,19	11	4,6,7,8,11,19	6,7,11,16,19	7,11,16	-	-

6. A COMPARISON OF MULTIPLE OUTLIER DETECTION METHODS FOR REGRESSION DATA

6.1. Introduction

In this chapter, a comparative analysis of some widely used multiple outlier detection procedures given in Chapter 3 for regression data is carried out by an extensive simulation study under a wide variety of situations.

Section 6.2. contains a summary information about the organization of all situations considered to assess the performances of the recently proposed outlier detection methods for regression type of data. Sections 6.3. and 6.4. consist of the detailed information on the performances of the multiple outlier detection procedures in the presence of interior and exterior x-space regression outliers.

6.2. Simulation Study

There are two broad classes of multiple outlier detection methods: direct methods and indirect methods. In this study I compare multiple outlier detection methods for linear regression from these classes that are either most recently published or most frequently cited in the statistical literature by using an extensive simulation. The indirect methods considered for the comparison in this simulation study are

- Least Median of Squares (LMS) by Rousseeuw (1984),
- Least Trimmed Squared (LTS) by Rousseeuw (1984, 1985),
- M-estimator by Huber (1973).

These three estimators are available as the internal functions of S-plus 6.0.

The direct methods considered for the comparison are

- Sequential point addition algorithm (HS93) by Hadi and Simonoff (1993),

- Influence matrix algorithm (P&Y) by Pena and Yohai (1995),
- Clustering algorithm (SM&R) by Sebert et al. (1998),
- Iteratively weighted least squares approach (C&M) by Chatterjee and Mächler (1997),
- BACON regression method (BR) by Billor et al. (2000),
- Robust iteratively reweighted least squares method (RWLS) by Billor et al. (2001).

The program codes for the aforementioned methods are written in S-plus 6.0 (Given in Appendix B). LMS, LTS, M, HS93, P&Y, SM&R, C&M, BR, and RWLS correspond to the abbreviations of the methods considered for comparison in this study and are used for the tables of the results in this simulation study.

In this simulation study I consider the following factors:

- The number of observations (n),
- The number of explanatory variables (p),
- The percentage of outlying observations (contamination level) and,
- The magnitude of outlying distance off the regression plane (δ_R).
- The magnitude of outlying distance in x-space (δ_L).
- The number of multiple point clouds,
- The proportion of unusual regressor variables out of the total k in the outlying observations

Configurations used in this study constitute close resemblance with those of Kianifard and Swallow (1990), Hadi and Simonoff (1993), Pena and Yohai (1995), Sebert (1998), Winsnowski et al. (2001) simulation studies.

Different data sets are generated depending on the configuration types. All data sets have fixed percentage of clean observations and contaminated outliers. The n_c clean observations for regressor variable levels are generated according to the model

$$y_i = x_i' \beta + \varepsilon_i, \quad i=1,2,...,n_c$$

where $x_i \sim N(7.5, 4^2)$, β is the vector of known regression coefficients arbitrarily selected for the simulations to be 0 for the intercept and 5.0 for each of the k regressor variables, ε_i is the random error term distributed $N(0,\sigma_e^2)$ with σ_e^2 set to 1. For the n_o outlying observations (planted outliers) for regressor variable levels, the *ith* regressor variable value for the *jth* observation is $x_{ij} = \bar{x}_{i,clean} + 4\delta_L + Uniform(0,0.25)$ where $\bar{x}_{i,clean}$ is the average of clean values for the *ith* regressor, and δ_L is the magnitude of the outlying shift distance in x-space in standard deviation units (i.e. 4). Depending on the *ith* observation to be regression outlier the response value is calculated by adding the magnitude of the outlying distance, δ_R, off the regression plane in standard deviation units (i.e. σ_e=1) to $x_i' \beta$.

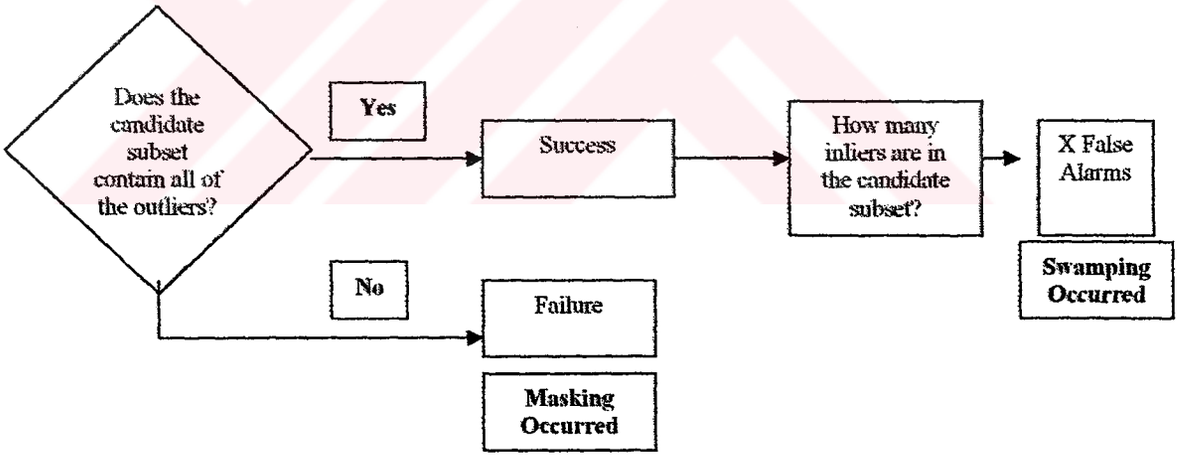

Figure 6.1. Flowchart summarizing the performance assesment of methodology

In this simulation study; the performance of each method is evaluated on its ability to detect outliers and for this reason, two performance measures are computed. One of them is the *probability which a clean observation is swamped* and the other is the *probability which an outlier is masked.* These probabilities are called *false alarm rate* ***(fr)*** and *detection capability* ***(dc)***, respectively. These values are reported for 500 replications. A good method should have high values of ***dc*** ***(i.e.***

close to 1) and a low value of ***fr (i.e. 0)***. The performance assesment of this methodology is summarized in Figure 6.1.

Table 6.1. Factors and Levels for the Simulated Data Sets

Factors	**Levels**
Number of Regressor Variables *(p)*	2, 6
Number of Observations *(n)*	40, 60
Contamination Level *(Cont. Level)*	10%, 20%, 15% and 30% in some studies
Outlier Distances (δ_R, δ_L)	Between 3 and 5 standard deviation units
Number of Clouds	1, 2
Number of Outlying Variables	1, 2, 6

Since the performance of the above methods is to be tested in a wide variety of situations, all the factors considered in this study are summarized in Table 6.1.

Simulation studies are examined in two main categories of regression outliers in order to evaluate the methods across a wide range of possible regression configurations in the most appropriate manner. These are

1) Interior x-space outliers
2) Exterior x-space outliers

Figure 6.2. displays the organization of the simulation study the appropriate table numbers within this chapter for the simulation study results

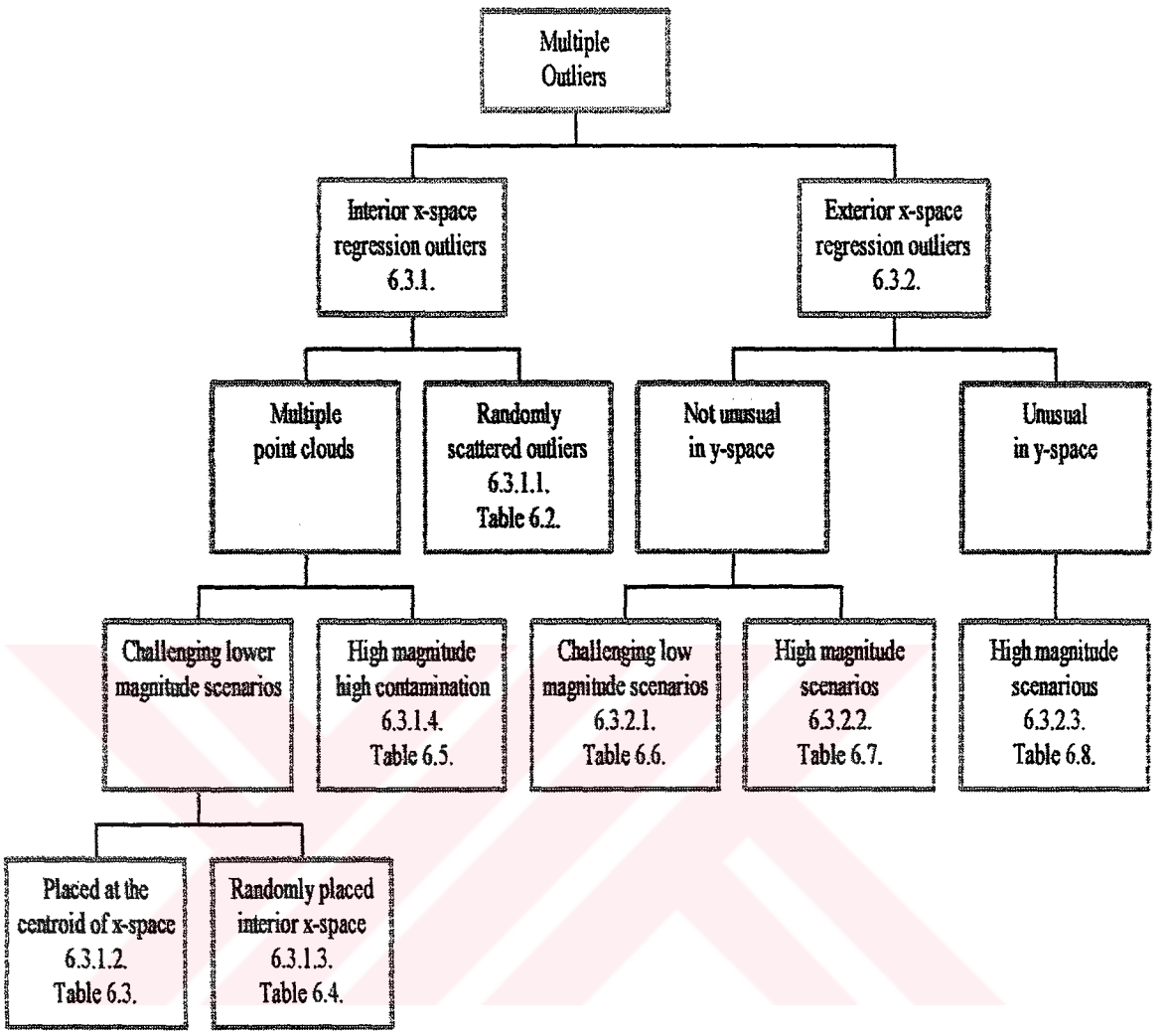

Figure 6.2. Organization Chart for the Simulation Studies

6.3. Analysis and Simulation Study Results

In this section the detailed experiment designs for the interior and exterior x-space regression outliers for the factors and the levels given in Table 6.1. and the detection capability (*dc*) and false alarm rates (*fr*) for the assessment of the performance of each method will be given.

6.3.1. Interior x-space Regression Outliers

In this section the performance of the methods are tested when all explanatory variable values are not unusual in x-space i.e. when there are no high leverage

observations in data sets. The response variables for the interior x-space outlying observations are offset a distance δ_R from the regression plane obtained from the clean cases.

In this section the performances of the methods are assessed under three different outlier configurations. These are:

- Multiple outliers randomly scattered in the interior of x-space
- Multiple point clouds located near the centroid of x-space
- Multiple point clouds randomly placed in the interior of x-space

Simulation study results for these configurations are given in Table A.5., Table A.6., and Table A.7., respectively. The last rows of the tables display the average values of the detection and false alarm rates for all the methods considered in this study.

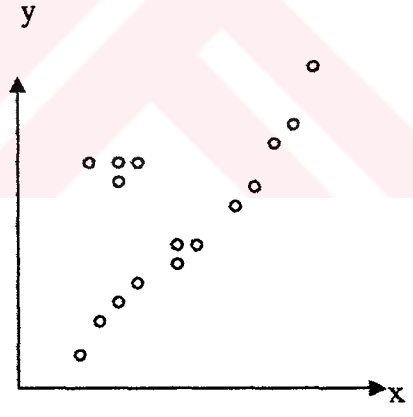

Figure 6.3. Interior x-space regression outliers with a single cloud

6.3.1.1. Multiple Randomly Scattered Regression Outliers in the Interior x-space

The purpose of this section is to determine the performance of the methods when data consist randomly scattered interior x-space regression outliers. The approach to creating data sets is to randomly generate n observations. Of these n observations, n_c "clean" observations are generated and represent the non-outlying

observations. n_o observations are also generated which are outliers ($n_c+n_o=n$). In this setting the n_c clean observations are generated according to the model

$$y_i = x_i' \beta + \varepsilon_i, \quad i=1,2,...,n_c,$$

where $x_i \sim N(7.5, 4^2)$, $\beta = [0,5,...,5]$ is the vector of known regression coefficients, ε_i is the random error term distributed as $N(0,\sigma_e^2)$ with σ_e^2 set to 1. The n_o outlying observations are generated according to the model

$$y_i = x_i' \beta + \delta_R, \quad i=1,2,...,n_o$$

where $x_i \sim N(7.5, 4^2)$, ε_i is $N(0,1)$ and δ_R is the magnitude of the outlying distance off the regression plane in standard deviation units, which is equal to *1*.

The factors used in this study are the number of observations (n), the number of regressors (p), regression outlying shift distance (δ_R), and contamination level (*Cont. Lev.*). The results are given in Table A.5.

The M regression estimators have high detection capability. Especially when we have high regression outlying magnitude ($\delta_R = 4, 5$) and high contamination level (20%) M regression estimators' detection capability stands out. However the false alarm rates of M estimator are high especially for high contamination level (20%), which is expected.

The OLS regression estimator has also excellent detection capability, and its detection capability gets even better as we have high regression outlying magnitude. However it swamps clean observations as we have large δ_R and high contamination level because I get high false alarm rates. This means that the clean data no longer fits well. Results in this configuration are therefore unsatisfactory for some scenarios.

The LMS and LTS methods which are known to have high breakdown points do provide good performance for high regression outlying distance especially at $4\sigma_e$

and above because of high *dc* and low *fr*. For high regression outlying distance scenarios and high contamination level the LMS estimator provides slightly better results than the LTS in high contamination scenarios especially at $4\sigma_e$ and the opposite is true for the low contamination scenarios.

The method of Chatterjee and Mächler (C&M) has the poorest detection capability at these outlying distances among all the methods considered in this study although it has the ideal false alarm rates. In the presence of randomly scattered regression outliers in the interior of x-space the C&M method is not a good method to use.

The method of Pena and Yohai (P&Y) has slightly better detection capability than C&M especially at $\delta_R = 5$. When I increase δ_R to 7 or 8 the P&Y method has much better detection capability.

The Sebert (SM&R) and BACON regression (BR) methods have high detection capability but they suffer from large false alarm rates in many scenarios. Especially the BR estimator has larger false alarm rates than the SM&R method. Therefore the SM&R method is slightly preferred over the BR estimator.

The Hadi and Simonoff method (HS93) has only high detection capability for high regression outlying distance especially for $5\sigma_e$ and above. For other factors and levels, the HS93 regression estimator has abnormally low false alarm rates and the detection capability is low.

The RWLS method has clearly better performance than the methods of P&Y, HS93, BR and C&M in low contamination level and high regression outlying distance, particularly for $4\sigma_e$ and above in terms of both measures since I have high detection capability and low false alarm rates for this method.

6.3.1.2. Regression Outliers in Multiple Point Clouds at the Centroid of x-space

In this section the performance of the methods are examined in the presence of regression outliers forming one or two clouds at the centroid of x-space.

In this configuration *ith* clean response value is generated by

$$y_i = x_i' \beta + \varepsilon_i,$$

where $\beta = [0,5,...,5]$, $x_i \sim N(7.5,4.0^2)$ and $\varepsilon_i \sim N(0,1)$. On the other hand the *ith* contaminated response value is generated from

$$y_i = x_i' \beta + \delta_R,$$

where x_i is the vector of k regressor variables distributed *Uniform(7.325, 7.625)* and δ_R is the outlying distance off the regression plane in standard deviation units. If there are two clouds then the outliers in the first cloud are generated by

$$y_i = x_i' \beta + \delta_R$$

and the response values for the second cloud are generated by

$$y_i = x_i' \beta - \delta_R.$$

The factors in this configuration are n, contamination levels, δ_R, and the number of clouds. The results are given in Table A.6.

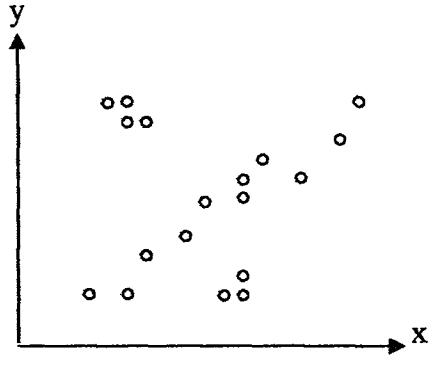

Figure 6.4. Interior x-space regression outliers with multiple point clouds

Results of this section show that the methods are more responsive to detecting outlying observations in clouds at the centroid of x-space than randomly scattered regression outliers. As it was the case in Section 6.3.1.1. again, the OLS and M

methods provide excellent results in terms of the performance measure of the detection capability for this configuration. But as indicated in the previous section the OLS method has serious swamping problems when there is a single cloud. When there are two clouds this problem disappears. Because there will be an equal and opposite pull on the regression plane from each cloud therefore the parameter estimates will remain unchanged from those obtained from the clean set of observations.

The Sebert (SM&R) method has high detection capabilities. But it suffers from high false alarm rates. The false alarm rates even higher than those obtained in Section 6.3.1.1. Therefore in the presence of swamping problem when I have two clouds the SM&R method should be used cautiously.

The LMS method provides perfect results for the most of regression outlying distances. Therefore it is preferred over the LTS regression estimator for this type of configuration.

The C&M and P&Y methods do not provide satisfactory results at all although their false alarm rates are nearly ideal.

The HS93 method has high detection capability in high regression outlying distances and abnormally low false alarm rates. However in the other scenarios it does not perform well.

The BR estimator has good performance in high regression outlying distance. When there is a single cloud I obtain low detection capability and high false alarm rates in high regression outlying distances and low contamination level, but when there are two clouds its performance appears to be satisfactory in almost all scenarios. But the false alarm rates remain high which is undesired situation.

The RWLS method has high detection capability and low false alarm rates in low contamination level and high regression outlying distances. Therefore this method is preferred over HS93, P&Y, SM&R, C&M and BR when I have two clouds, low contamination level (10%) and high regression outlying distances.

Most of the results in Table A.6. are consistent with the results in Table A.5.

6.3.1.3. Regression Outliers in Multiple Point Clouds: Regressor Variables Randomly Scattered on the Interior of x-space

This section consists of the results for the performance of the direct and indirect methods considered in this simulation study in the presence of multiple outlier clouds placed at different locations in x-space.

In this configuration the *ith* clean response value is generated by

$$y_i = x_i' \beta + \varepsilon_i,$$

where $\beta = [0,5,...,5]$, $x_i \sim N(7.5,4.0^2)$ and $\varepsilon_i \sim N(0,1)$. I determine the location of the regressors for outlying observations in a single cloud by finding the median of the *first* three clean observations for each regressor variable. Then the *ith* contaminated response value is generated from

$$y_i = x_i' \beta + \delta_R,$$

where the regressor variables for the outlying observations x_i vary as *Uniform(0,0.25)* around the calculated median value and δ_R is the outlying distance off the regression plane in standard deviation units. Outliers in a second cloud vary also around the median value of the *last* three clean observations for each variable. This way of generating outliers assures us having randomly scattered planting outliers in two clouds

The factors used in this configuration are n, contamination levels, number of clouds, and δ_R. The simulation results for this configuration are given in Table A.7.

The M and OLS methods have high detection capabilities especially in high regression outlier scenarios. The detection capability of the M estimator is moderately higher than the detection capability of the OLS estimator and the false alarm rates of the M estimator are lower than those of the OLS estimator, although the false alarm rates are well above nominal levels (0.05) in the high dimension,

high contamination cases. False alarm rates of these methods are moderately increased in high regression outlying scenarios. The OLS estimates do not fit the outlying clouds well as evidenced by the high detection capability, but these estimates chase these observations enough to swamp some clean observations.

The high breakdown methods (LMS and LTS) have only high detection capability for high regression outlying distances especially $4\sigma_e$ and beyond and are not affected with high false alarm rates. In most of the scenarios the LMS estimator performs better than the LTS estimator.

The SM&R method fails in this configuration since it has low detection capability and high false alarm rates except high regression outlying distance for two clouds scenarios.

The P&Y performs slightly better with the increased leverage for these outlying clouds, but still not competitive with the other methods.

The C&M estimator performs much better in this configuration than the previous configurations, but still has considerably low probabilities of detection. Therefore it is not competitive with any other methods.

The HS93 method has slightly better performance in high outlying distances than the previous configuration. It has low false alarm rates in all scenarios.

The RWLS method has reasonable good performance for low contamination level and high outlying distances, especially for $4\sigma_e$ and above and has also considerably low alarm rates. Although the probabilities of the detection capability for this configuration are lower than those of the previous configuration the RWLS method is competitive with the methods of HS93, P&Y, SM&R and C&M and BR.

The BR method has higher detection capability than the previous configuration, but it still has very high false alarm rates which cause serious problems. Therefore this method is not competitive with any other methods since it has the highest false alarm rates. The detection capability of this method increases in high regression outlying distances. As the dimension decreases, its false alarm rates increases.

6.3.1.4. Additional Runs: Higher Regression Outlying Distances and Higher Contamination Levels

In this section the effectiveness of the methods are tested in higher regression outlying distances and in higher contamination levels. I will consider two contamination levels (%15 and %30) and two regression outlying distances (5 and 10) which I did not consider these cases in the previous configurations. The results are given in Table A.8.

The first two scenarios in this table are randomly scattered outliers in x-space, the next four scenarios correspond to the multiple point clouds placed at or near the centroid and the last eight scenarios are to assess the detection capability of the methods when the outliers are placed in clouds randomly in the interior of x-space similar to those of Section 6.3.1.3.

In the presence of high contaminated data and high regression outlying distances the methods produce some considerable different results. When I set the contamination higher to 15% or 30% it is now even clearer that the false alarm rates of the OLS and M estimators are very high whereas the detection capabilities of these methods are excellent for this configuration except 12^{th} scenario, where the contamination level is set to 30% and the outlying distance is set to 5. So the detection capability of the OLS and M estimators break down at high contamination level and for high outlying distance.

The Hadi and Simonoff (HS93) method has good performance in almost all scenarios except high regression outlying distance (i.e. $5\sigma_e$) that corresponds the 7^{th}, 10^{th} and 12^{th} scenarios. The first two scenarios have moderate values but for 12^{th} scenario where the contamination level 30%, the HS93 method has relatively low detection capabilities. As a result of this as the contamination level and outlying distances increase the HS93 method breaks down in terms of detecting capability.

The Pena and Yohai (P&Y) method provides reasonably high detection capability and also low false alarm rates for low contamination level and high regression outlying distances. Therefore I can say that the Pena and Yohai procedure is successful only for low contamination and high regression outlying distance

scenarios. It provides more satisfactory results in this configuration than the previous one.

The LMS and LTS estimators perform well except the 12^{th} scenario. Both of them give perfect results for detection capabilities and false alarm rates at other times.

Although the Sebert method has high detection capability, especially for high regression outlying distance, except 12^{th} scenarios this method is vulnarable to swamping since it produces high false alarm rates.

The Chatterjee and Mächler method (C&M) does not perform well, therefore is not competitive with any other methods for this configuration.

The RWLS method has much better performance than the previous configuration since it has much higher detection capability for low contamination (15%) and high regression outlying distances scenarios (δ_R=10).

The BACON regression (BR) method has high detection capability for low contamination level and high regression outlying distances, but has very high false alarm rates. Therefore this method is again not competitive with any other methods since it produces the third highest false alarm rates after the OLS and M estimators.

As a result of this configuration, all procedures fail when contamination level is set to 30% and the outlying distance is set to $5\sigma_e$ and in a single cloud scenarious.

6.3.2. Exterior x-space Regression Outliers

This section investigates the performance of the methods when data consist of exterior x-space regression outliers that have high leverage and large residual.

Two different configurations are examined in this section. The first one has multiple point clouds at various leverage locations when the regressor variable values are remote in all p regressors for the planted outliers. The second one is nearly the same as the first one except this one ensures that the response values for the outliers are not unusual with respect to the clean responses.

The factors for these studies are contamination levels, regression outlying distances (δ_R), leverage outlying distances (δ_L), the number of multiple point clouds and outlying parameter numbers.

Simulation results are given in Table A.9., Table A.10., and Table A.11.

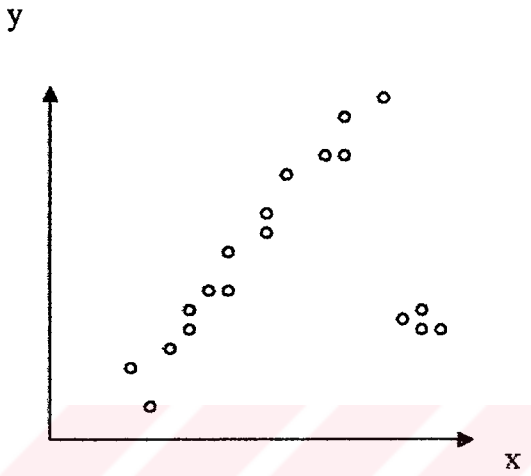

Figure 6.5. Exterior x-space regression outliers with a single point cloud

6.3.2.1. Regression Outliers in Clouds that are Unusual in x-space for All Regressors

In this study, there are multiple point clouds at various leverage locations. And the explanatory variables are remote in all p regressors for the planted outliers.

The ith clean response value is generated by

$$y_i = x_i' \beta + \varepsilon_i ,$$

where $x_i \sim N(7.5, 4.0^2)$, β is the same vector of known regression coefficients, ε_i is the random error term distributed $N(0, \sigma_e^2)$ with σ_e^2 set to 1. And ith contaminated response value is generated by

$$y_i = x_i' \beta + \delta_R ,$$

where δ_R is the outlying distance off the regression plane in standard deviation units. The *value* of the *ith* explanatory variable for the *j*th planted outlier is computed by

$$x_{ij} = \bar{x}_{i,clean} + \delta_L + \varepsilon_{ij}^*,$$

where $\bar{x}_{i,clean}$ is the mean of the clean observations for the *ith* explanatory variable, δ_L is the leverage magnitude in standard deviation units, σ_x, and ε_{ij}^* is distributed *Uniform(0,0.25).*

When the leverage magnitude is low, that is $\delta_L=2\sigma_x$, a multiple point cloud will be at the boundary of interior x-space. When the leverage magnitude is high, that is $\delta_L=5\sigma_x$, the cloud will be considerably far away from the interior x-space. If there are two clouds, the explanatory variables in the first and second clouds are determined as in the form given above but the response values are located $2\sigma_e$ above that of the first cloud i.e. if regression outlying distance is chosen equal to $5\sigma_e$ for the first cloud than the second cloud regression outlying distance must be chosen equal to $7\sigma_e$.

I can give an example to explain this configuration. Consider the case for $p=2$, $\delta_L=2\sigma_x$, $\delta_R=3\sigma_e$, the regressor variable values for the *i*th planted outlying observation in either cloud are

$$x_1 = 7.5 + 2\,(4) + Uniform(0,0.25) \text{ and } x_2 = 7.5 + 2\,(4) + Uniform(0,0.25).$$

The response value for the *i*th outlying observation in the first and second cloud is obtained from

$$y_i = 5\,x_{1i} + 5\,x_{2i} + 3 \text{ and } y_i = 5\,x_{1i} + 5\,x_{2i} + 5,$$

respectively.

In this section I examine 32 different scenarios: 16 for one cloud and 16 for two clouds. The results are given in Table A.9.

The Hadi and Simonoff method (HS93) has poor detection capability in almost all scenarios. We know that the logic of this method is based on adding sequentially one observation that has the smallest OLS residual to the clean set. Since the high leverage observations have small OLS residual values and therefore, often masked. On the other hand it has very low false alarm rates.

The Pena and Yohai method (P&Y) has slightly better detection capabilities than Hadi and Simonoff method and its performance is reasonable when there are two clouds.

The Sebert method (SM&R) successfully detects the outlying clouds in all scenarios. Its detection capability is virtually perfect and the false alarm rates are reasonably low.

The OLS method performs very poorly for this configuration. The false alarm rates for low leverage and high regression outlier distance are high and detection capabilities are small almost in all of the scenarios.

The LTS, LMS and M methods have poor detection capabilities as expected. These methods perform well only for low leverage, high regression outlying distance, and low contamination scenarios, particularly in small dimension scenarios.

The Chatterjee and Mächler procedure is the worst method for this configuration.

The BACON regression method is troubled in high contamination level but it has reasonable performance in low contamination, high regression and leverage outlying distance scenarios. This method has moderately reasonable detection capability, but very high false alarm rates. Its performance in terms of detection capability is better for two clouds scenarios.

The RWLS method has moderate performance in high leverage, high regression outlying distance, and low contamination level. However it has poor detection capability in high contamination. Since it has reasonably low false alarm rates the RWLS estimator performs better than the methods of HS93, C&M, BR, and OLS, therefore this method is competitive with the methods of HS93, C&M, BR, and OLS.

6.3.2.2. Regression Outliers which are Unusual in x-space for All Explanatory Variables but the Outlier Responses are not unusual in y-space

This study is similar which is given in Section 6.3.2.1. But here contaminated response values are not unusual with respect to the clean response values.

In this study *ith* clean response value is generated by

$$y_i = x_i' \beta + \varepsilon_i ,$$

where $x_i \sim N(7.5, 4.0^2)$, β is the vector of known regression coefficients as before, ε_i is the random error term distributed $N(0, \sigma_e^2)$ with σ_e^2 set to 1 and *ith* contaminated response variable is generated by

$$y_i = x_i' \beta + \delta_R ,$$

where δ_R is the outlying distance off the regression plane in standard deviation units. The *j*th value of *ith* explanatory variables for the contaminated data is determined by

$$x_{ij} = \begin{cases} \bar{x}_{i,clean} + 4\delta_L + \varepsilon_{ij}^* & \text{for i} = 1,3,5 \\ \bar{x}_{i,clean} - 4\delta_L + \varepsilon_{ij}^* & \text{for i} = 2,4,6 \end{cases}$$

where $\bar{x}_{i,clean}$ is the mean of the clean observations for the *ith* explanatory variable, δ_L is the leverage outlying distance in standard deviation units, σ_x, and ε_{ij}^* is distributed *Uniform(0,0.25)*.

If there are two explanatory variables then the first contaminated explanatory variable is calculated by adding $4\delta_L$ to the original data and the second contaminated one is calculated by subtracting $4\delta_L$ from the original data. This way

of defining the contaminated data will prevent the outliers from being y-space outliers. The results are given in Table A.10. The four scenarios are randomly selected from those in Section 6.3.2.1. for the outliers generated as described in this subsection.

Most of the methods have problems because of low detection capability, particularly for the single cloud scenarios in Table A.10.

In contrast the other configurations, the Sebert method fails in this configuration since it has low detection capability and high false alarm rates in almost all scenarios.

The OLS, LTS, LMS, and M estimators do not provide satisfactory results.

The HS93 and P&Y methods have poor detection capability except the first scenario. The BACON regression algorithm gives the best results in terms of detection capability; however it is vulnerable to swamping because of very high false alarm rates. The Chatterjee and Mächler method fails in this configuration like in the other configurations.

The RWLS method has the best detection capability and low false alarm rates for this configuration among all the methods in the class of the direct methods.

6.3.2.3. Outliers are Unusual in the First Column of x-space (Dash and Dot Configuration)

In this section I investigate the performance of these methods if the outliers are unusual on the first column of x-space with high leverage outlying distances. This configuration is a milder version of the Buxton data set which was used in Chapter 5 to assess the performances of the methods on real data sets. I have shown that many methods fail to detect outliers in the Buxton data set. Therefore it will be useful to consider this type configuration and to assess the performances of the available methods. Hawkins et al. (2002) defined this configuration and called Dash and Dot configuration.

The *ith* response value for the clean case is generated from

$$y_i = x_i\beta + \varepsilon_i,$$

where the first value of x_i vector is generated from *uniform U(-3,3)* and the others are generated from *N(0,1)*, $\beta = [0,5,...,5]$ and ε_i is the random error term distributed as $N(0,\sigma_e^2)$ with σ_e^2 set to 1.

On the other hand the *i*th response value for the contaminated case is generated from

$$y_i = x_i\beta + \delta_R,$$

where the first value of x_i vector is generated from *Uniform(19,20)* and the others are generated from *N(0,I)*. If there are two clouds, the response value in the first cloud is generated by

$$y_i = x_i\beta + \delta_R$$

and the other is generated by

$$y_i = x_i\beta - \delta_R,$$

where regression outlying distance δ_R is chosen equal to 6 for one cloud and +6 or –6 for the second cloud. The sign is determined at random. The factors for this configuration are n, the number of regressors, contamination levels and the number of clouds. The results are given in Table A.11.

The P&Y and OLS methods have moderately high performance for two clouds scenarios. But for the single cloud scenarios both of them have poor detection capabilities. The Sebert method shows perfect performance in almost all scenarios.

The LMS method has moderate detection capabilities and false alarm rates in both low contamination level and two clouds scenarios. But in single cloud scenarios it is not preferred. LTS has poor performance than LMS in this configuration. It is

expected results because it is known that these methods become ineffective on the exterior x-space scenarios.

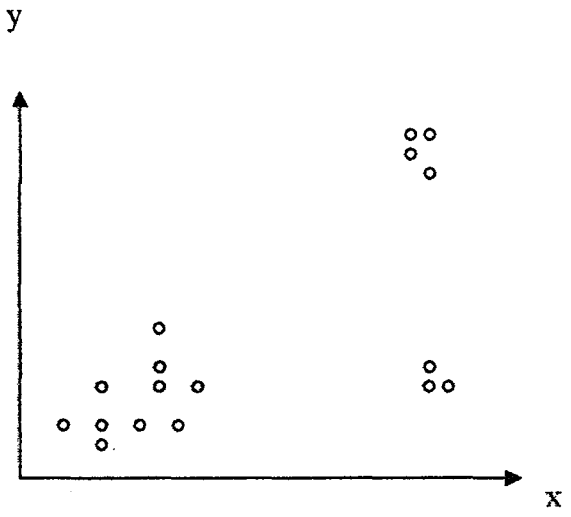

Figure 6.6. Exterior x-space regression outlier with multiple point clouds

The M method fails in this configuration because of known vulnerabilities in high leverage situations. Therefore it has poor detection capabilities.

The HS93 method has moderate detection capability for two clouds scenarios but for single cloud it does not perform well. The Chatterjee and Mächler method has low detection capability especially in single cloud scenarios.

The RWLS method has reasonable performance in low contamination level. The detection probabilities are considerably high particularly in two clouds scenarios.

The BR method has high detection capabilities but it is troubled by high false alarm rates particularly when dimension of the data set is small.

6.4. Summary of the Comparative Analysis and Discussion of the Results

In this section all of the above results are summarized and the performances of the methods are discussed.

6.4.1. Summary of the Performance of the Methods

The *Hadi and Simonoff* method has good performance in high regression outlying distance scenarios in Section 6.3.1.1. When there are regression outliers in the interior of x-space. This method has low false alarm rates in almost all scenarios. Winskowski et al. (2001) presented that the detection capability is increased by increasing the value of α (significance level) from 0.05 to 0.20 without severe impact to false alarm probabilities. Detection capability moderately declines in the high dimension and high contamination level scenarios. If I compare the results of HS93 results in Section 6.3.1.1. and 6.3.1.2., it is easily seen that in multiple point cloud configuration its detection capabilities are low for high regression outlying distances, opposite is true for high regression outlying distance scenarios, especially for $5\sigma_e$ and above. Furthermore it has good performance in low contamination level and detection capability is increased to reliable levels if the regression outlying distance is sufficiently large relative to the leverage outlying distance in the exterior of x-space. In high magnitude scenarios it has poor performance except the first scenario. Its performance gets better in low contamination level. In dash and dot configuration when there is a single cloud and large regression outlying distance the HS93 method does not perform well whereas when there are two clouds it has reasonable performance. So I can confidently say that the Hadi and Simonoff method is effective for interior x-space regression outlier configurations when regression outlying distance is large and contamination level is low. It has even better performance in two clouds scenarios. However overall performance noticeably decreases in the presence of exterior x-space outlier, i.e. as leverage is added as a factor.

The *Pena and Yohai* method detects regression outliers in interior x-space regression outlier configurations for both high regression outlying distance especially $5\sigma_e$ and $7\sigma_e$ and low contamination level up to %15. This procedure rarely swamps clean observations. When the magnitudes of δ_L and δ_R are large enough then this procedure can be used effectively. In dash and dot configuration it has reasonable performance in two clouds scenarios.

The *Sebert* method has good performance in almost all configurations except high regression and low leverage outlying distance scenarios when randomly placed multiple point clouds are located in the interior x-space. As the regression outlying distance increases its false alarm rates decreases. If the response values are not unusual in y-space and outlying distance is low than it works well but in high outlying distance it breaks down.

The *Chatterjee and Mächler* method is ineffective in both interior and exterior x-space regression outlier configurations. Despite the comparable false alarm rates, it suffers from severe from detection capability in all configurations.

The *BACON regression* method has good performance in high regression outlying distance scenarios in Section 6.3.1.1., when the regression outliers are in the interior of x-space. But it has high false alarm rates. Its detection capability declines in high contamination level. It has better performance at the existence of randomly scattered interior x-space regression outlier than multiple point clouds in the interior x-space regression outlier scenarios. In Tables A.9. and A.10. I can see that the detection capability of this method increases to reliable levels when the number of regressors is small and the regression outlying distance is sufficiently large relative to the leverage outlying distance. Overall performance noticeably degrades as the contamination level decreases and the magnitude of outlying distance increases. This method performs well when there are two clouds with low contamination level and high outlying distance. But it suffers from severe false alarm rates in almost all of the configurations.

The *RWLS method* is generally effective in the presence of interior and exterior x-space regression outliers for low contamination level in terms of performance measures; detection capability probabilities and false alarm rates. It has low breakdown point (10%-15%). As the regression outlier distance increases its detection capability increases. In exterior x-space regression outliers it has high detection capability and low false alarm rates on low contamination and high regression outlying distance scenarios. In two clouds scenarios, detection capability and false alarm rates are slightly better than a single cloud scenarios. In dash and dot configuration it has moderate detection capability and low false alarm rates.

Particularly in dash and dot configuration for low contamination level and two clouds scenarios it performs well.

Both the *LMS* and *LTS* procedures detect regression outliers in the interior of x-space if the regression outlying distance is high. These estimators also have low false alarm rates across all scenarios for high regression outlying distance. But they are not effective on exterior x-space regression outlier configurations. In high regression outlying distance and high contamination level scenarios the LMS method has slightly better performance than the LTS method. Detection capability decreases for both estimators as the outliers become more remote in x-space. The LMS and LTS methods have significant masking and swamping problems in high leverage outlying distance scenarios.

The *M estimator* fails in the high leverage outlying distances scenarios of Section 6.3.2. In interior x-space configuration it has excellent detection capability but have moderate false alarm rates. Its performance degrades in exterior x-space regression outlier configurations. The scenarios where this method has good performance are: outliers in interior x-space, low leverage magnitude, high regression outlying distance, and low contamination level. This method is not competitive with the others in exterior x-space configurations except the scenarios, which are given above.

6.4.2. Summary of the Results

In general it is expected that the performance of the outlier detection methods:

- increases in high regression outlying distance (δ_R)
- increases in low leverage outlying distance (δ_L)
- decreases as the number of regressor variables increases
- decreases in high contamination level
- increases in larger number of multiple point clouds

However, I showed that there are some scenarios that the expected results given above in the itemized list are not the cases for all methods and all factors. Some factors behave opposite to the general rule.

From these results, I should point out that small number of studies when the number of observations (n) is small in a proposed procedure is not sufficient to speculate on its performance in higher dimension especially if the contamination level is high.

I conclude that

From the *interior x-space* studies

- The Hadi and Simonoff algorithm can be recommended if the regression outlying distance is large since the probabilities of detection capability are large and false alarm rates are low.
- The RWLS method can be recommended for low contamination level and high outlying distances (both δ_R and δ_L) scenarios because this method has moderately high detection capability and low false alarm rates.

From the *exterior x-space* studies in Section 6.3.2.

- Even though the Sebert method does not perform well for the case of outliers in interior x-space, it has nearly excellent performance in most of the scenarios where the outliers are planted in the exterior x-space since the probabilities of detection capability are very large and false alarm rates are reasonably low. This method does not perform well only in high regression outlying distance scenarios when observations are *not unusual* in y-space. In these scenarios not only its detection capabilities are low, but also its false alarm rates are high.

- The RWLS method has good performance for low contamination level scenarios, high regression outlying distances (both δ_R and δ_L) because of high detection capability and low false alarm rates. Furthermore this method performs the best for high regression outlying distance scenarios when observations are *not unusual* in y-space.
- Although the BACON regression method has the ***highest*** detection capability among all the direct methods it has the ***highest*** false alarm rates which make this method inefficient in practice.
- While high-breakdown methods perform well for the interior of x-space scenarios as expected they become ineffective for the exterior x-space scenarios.

I also conducted a separate simulation study for another challenging configuration called "Dash and Dot". As a result of this study I conclude that

- The Sebert method performs the best for all scenarios in this configuration. The RWLS method performs reasonably well for all the scenarios in this configuration.
- Among high breakdown methods the LMS method performs relatively better than the other robust methods.

REFERENCES

AGULLÓ, J., 1996. Exact iterative computation of the multivariate minimum volume ellipsoid estimator with a branch and bound algorithm. In Proceeding in Computational Statistics, ed. A. Prat, Heidelberg: Physica-Verlag, pp. 175-180.

ANDREWS, D. F., 1972. Plots of High-Dimensional Data. Biometrics, 28, 125-136. (309).

ANDREWS, D. F., PREGIBON, D., 1978.Finding the outliers that matter. J. Royal Stat. Soc. B, 40, 85-93 (339, 345, 350, 373, 374).

ATKINSON, A. C., MULIRA H. M., 1993, The stalactite plot for the detection of multivariate outliers, Statistics and Computing, 3, 27-35.

ATKINSON, A.C., 1986. Masking unmasked. Biometrica, 73,3,533-541.

ATKINSON, A.C., 1994. Fast very robust methods for the detection of multiple outliers, Journal of the American Statistical Association, 89,1329-1339.

ATKINSON, A.C., RIANI, M., 2000. Robust Diagnostic Regression Analysis. New York; Springer.

ATKINSON, A.C., 1982a. Regression diagnostics, transformation and constructed variables (with discussion). J. Roy. Statist. Soc. B, 44,1-36.(322)

BACON-SHONE, J., FUNG, W.K., 1987. A new graphical method for detecting single and multiple outliers in univariate and multivariate data. Journal of the Royal Statistical Society (C), 36, No.2, 153-162.

BARNETT, B.E., LING, R.F., 1992. Genaral classes of influence measures for multivariate regression. J. Amer. Statistics Assos., 87, no.417, 184-191.

BARNETT, V., LEWIS, T., 1994. Outliers in Statistical Data. 3rd edition, New York: John Wiley and Sons.

BECKMAN R. J., COOK, R.D., 1983. Outlier.....s Technometrics. 25:119-163.

BELSLEY, D. A., KUH, E., WELSCH, R. E., 1980. Regression Diagnostic: Identifying Influential Data and Sources of Collinearity. Wiley, New York. (317, 349, 370, 373).

BILLOR, N. , HADI, A. S., VELLEMAN, P. F., 2000. BACON: Blocked Adaptive Computationally-Efficient Outlier Nominators, Computational Statistics and Data Analysis, 34, 279-298.

BILLOR, N., CHATTERJEE, S., HADI, A. S., 2001. Iteratively Re-weighted Least Squares method for outlier detection in linear regression. Bulletin of the International Statistical Institute, 1, 470-472.

BROWN, R.L., DURBIN, J., EVANS, J.M., 1975. Techiniques for testing the constancy of regression relationships over time. Journal of the Royal Statistical Society, Series B, 37: 149-192.

BROWNLEE, K.A., 1965. Statistical Theory and Methodology in Science and Engineering. 2nd edn, Wiley, N.Y.

BUTLER, R. W., DAVIES, P.L., JHUN, M., 1993. Asymptotics for the minimum covariance determinant estimator. The Annals of Statistics, 21, 1385-1400.

BUXTON, L. H.D., 1920. The anthropology of cyprus. The Journal of the Royal

Anthropological Institute of Great Britain and Ireland, 50, 183-235.

CAMPBELL, N. A., 1980. Robust procedures in multivariate analysis I: Robust covariance estimation. Applied Statistics, 29 , 231-237.

CARONI, C., PRESCOTT, P., 1992. Sequential application of Wilk's multivariate outlier test. Applied Statistics, 41, 355-364.

CARONI, C., 2000. Outlier detection by robust principal components analysis. Commun. Statist.-Simula., 29(1), 139-151.

CHATTERJEE, S., HADI, A.S., 1988. Sensitivity Analysis in Linear Regression. New York: John Willey.

COOK, R. D., 1977. Detection of influential observation in linear regression. Technometrics, 19, 15-18. (370,373)

COOK, R.D., HAWKINS, D.M., WEISBERG, S., 1992. A comparision of model misspecification diagnostics using residuals from least mean of squares and least median of squres fits. Journal of American Statistical Association, vol. 87, no. 418, 419-424.

CROUX, C. RUIZ-GAZEN A., 2000. High breakdown estimators for principle components: the projection-pursuit approach revisited, under revision, http://homepages.ulb.ac.be/~ccroux.

CROUX, C. FILZMOSER P., 2001. A Projection-Pursuit based Measure of Association between two Multivariate Variables. In Preparation.

DRAPER, N.R. , SMITH, H., 1966. Applied Regression Analysis. New York: John Willey.

FUNG, W.K., 1993. Unmasking outliers and leverage points:A confirmation. J. Amer. Statist. Asso., 88, 515-519.

GNANADESIKAN, R., KETTENRING, J. R., 1972. Robust estimates, residuals and outlier detection with multiresponse data. Biometrics, 28, 81-124. (269, 274, 278, 279, 298, 302, 304, 308, 309, 369).

GRAY, J. B., LING, R. F., 1984. K-clustering as a detection tool for influential subsets in regression . Technometrics, 26, 305-318. (454).

GRAY, J.B., 1986. A simple graphic for assessing influence in regression. Journal of Statistical Computation and Simulation, 24, 121-134.

HADI , A. S., 1992. Identifying multiple outliers in multivariate data. Journal of the Royal Statistical Society, series(B), 54, 761-771.

HADI, A. S., 1994. A modification of a method for the detection of outliers in multivariate samples. Journal of the Royal Statistical Society, series(B), 56, No. 2.

HADI A., S., JOO, H. , MUN, S. SON, 1996. A comparision of methods for detection of outliers in multivariate data. The Korean Statistical Society, pp, 53-67.

HADI, A. S. , SIMONOFF, J. S., 1993. Procedures for the identification of multiple outliers in linear models. Journal of the American Statistical Association, Vol. 88 ,414, 1264-1272.

HADI, A. S. , SIMONOFF, J.S., 1997. A more robust outlier identifier for regression data. Bulletin of International Statistical Institude, 281-282.

HAWKINS, D. M., 1980. Identification of Outliers. London: Chapman and Hall.

HAWKINS, D. M. , BRADU, D. , KASS, G. V., 1984. Location of several outliers in multiple regression data using elemental sets. Technometrics, 26,197-208.

HAWKINS, D.M., OLIVE, D.J., 2002. Inconsistency of resampling algorithms for high-breakdown regression estimators and a new algorithm. Journal of the American Statistical Association, Vol. 97, No. 457, Theory and Methods, 136-159.

HEALY, M. J. R., 1968. Multivariate normal plotting. Applied Statistics, Vol. 17, No. 4, 1662-1663.

HOAGLIN, D.C., WELSCH, R. E., 1978. The hat matrix in regression and ANOVA. American Statistician, 32, 17-22. (374).

HÖSSJER, O., 1994. Rank-Based estimates in linear model with high breakdown point. Journal of the American Statistical Association, 89, 149-158.

HUBER, P., 1981. Robust Statistics. New York: John Wiley and Sons.

HUBERT, M., ROUSSEEUW, P.J., VERBOVEN, S., 2002. A fast robust method for principal components with applications to chemometrics. Chemometrics and Intelligent Laboratory Systems, 60, 101-111.

HUBERT, P.J., 1973. Robust regression-asymtotics, conjectures and Monte Carlo. Ann. Statist., pp.799-821.

KIANIFORD, F., SWALLOW, W. H., 1989. Using recursive residuals, calculated on adaptively-ordered observations to identifiy outliers in linear regression.

Biometrics, 45, 571-585.

KIANIFORD, F., SWALLOW, W. H., 1990. A Monte Carlo comparision of five procedures for identifiying outliers in linear regresssion. Com. Statist., Part A- Theory and Methods 19, 1913-1938.

KRASKER, W. S., WELSCH, R. E., 1982. Efficient bounded-influence regression estimation. J. Amer. Statist. Assn., 77, 595-604. (324).

LOPUHAÄ, H.P. , ROUSSEEUW, P.J., 1991. Breakdown points of afine equavariant estimatos of multivariate location and covariance matrices. The Annals of Statistics, 19, 229-248.

MARASINGHE, M.G., 1985. A multistage prcedure for detecting several outliers in linear regression. Technometrics 27:395-399.

MARAZZI, A., 1991. Algorithms and Programs for Robust Linear Regression, In:W. Stahel and S. Weisberg, Eds. Directions in Robust Statistics and Diagnostics: Part 1. New York, Springer Verlag, 183-199.

MARDIA, K. V., 1970. Measures of multivariate skewness and kurtosis with applition. Biometrica , Vol. 57, 3,519-530.

MONTGOMERY, D.C. , PECK, E.A., 1992. Introduction to Linear Regression Analysis, Wiley, N.Y.

OLIVE, D.J., 2002. Applications of robust distances for regression. Technometrics, 44,no1, 64-71.

PAUL, S.R. , FUNG, K. Y., 1991. A generalization extreme studentized residual multiple outlier detection procedure in linear regression, Technometrics,

33,229-348.

PENA, D., YOHAI, V., 1995. The detection of influential subsets in linear regression by using an influence matrix. Journal of the Royal Statistical Society, Series B, Vol. 57, No.1 , pp.145-156.

_____, 1999. A fast procedure for outlier diagnostics in large regression problems. Journal of the American Statistical Association, Vol 94, 424-445.

PIRES, A.M. , BRANCO, J.A., 2001. Projection-Pursuit Approach for robust Linear Discriminant Analysis. Preprint, Instituto Superior Tecnico, Dept. of Math.

PRESCOTT, P., 1975. An approximate test for outliers in linear models. Technometrics 17: 129-132.

ROHLF, F. J., 1975. Generalization of the GAP Test for detection of multivariate outliers. Biometrics, Vol. 31, 93-101.

ROUSSEEUW, P.J., 1984. Least Median of Squares regression, Journal of the American Statistical Association, 79, 871-886.

_____, 1985. Multivariate estimation with high breakdown point in mathematical statistics and applications, Vol B ,eds. W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, Dordrecht: Reidel, 283-297.

ROUSSEEUW, P.J., VAN DRIESSEN, K., 1999. A fast algorithm for the minimum covariance determinant estimator. Journal of the American Statistical Association, Vol 41, 212-223.

_____, 2000. Computing LTS regression for large data sets, Technical report, Department of Mathematics and Computer Science, Belgium (submitted).

ROUSSEEUW, P.J., VAN ZOMEREN, B.C., 1990. Unmasking multivariate outliers and leverage points (with discussion). Journal of the American Statistical Association, Vol. 85, 633.

ROUSSEEUW, P.J. , LEROY, A. M., 1987. Robust Regression and Outlier Detection, New York: John Wiley and Sons.

ROUSSEEUW, P. J. , YOHAI, V. J., 1984. Robust regression by means of S-estimators, in robust and nonlinear time series analysis. Lecture Notes in Statistics Springer Verlag: New York , 26, 256-272.

SCHWAGER, S. J. , MARGOLIN, B. H., 1982. Detection of multivariate normal outliers. Ann. Statist., 10, 943-954.

SEBERT D.M.,DOUGLAS, MONTGOMERY C., ROLLIER DWAYNE A., 1998. A clustering algorithm for identifying multiple outliers in linear regression. Computational statistics and data analysis, 27, 461-484.

SEBERT, D.M., 1996. Identifying multiple outliers and influential subsets in linear regression: a clustering approach. Unpublished dissertation. Dept. of Industrial Engineering, Arizona State University.

SIMONOFF, J. S., 1991. General approaches to stepwise identification of unusual values in data analysis, directions in robust statistics and diagnostics: Part II, W. Stahel and S. Weisberg, eds.,Springer–Verlag: New York, 223-242.

SWALLOW, W. H. , KIANIFARD, F., 1996. Using robust scale estimates in detecting multiple outliers in linear regression. Biometrics, 52, 545-556.

TIETJEN, G. L., MOORE, R.H., BECKMAN, R.J., 1973. Testing for a single

outlier in simple linear regression. Technometrics 15: 717-721.

WILK, M.B., GNANADESIKAN, R., HUYETT, M.J., 1962. Probability plots for Gamma distribution. Technometrics, 4, 1-20.

WILKS, S., 1963. Multivariate statistical outliers, Sankhya, A25, 407-426.

WINSNOWSKI, W.J., MONTGOMERY, D.C., JAMES, R.S., 2001. A comparative analysis of multiple outlier detection procedures in the linear regression model. Computational Statistics and Data Analysis. 36, 351-382.

WOODRUFF, D.L., ROCKE, D.M., 1993. Heuristic search algorithms for the minimum volume ellipsoid. Journal of Computational and Graphical Statistics, 2, 69-95.

YAHOI, V. J., 1987. High breakdown-point and high efficiency robust estimates for regression. The Annals of Statistics, 15, 642-656.

RESUME

I was born in 1973 and completed my primary and secondary education in Adana. I graduated from the Department of Mathematics at Cukurova University in 1993. During the following year I was admitted to the MSc program in Applied Mathematics at the Department of Mathematics at Cukurova University and at the same time, started working as a research assistant at the Department of Econometrics. I have received MSc degree in 1996. In the following year I started PhD study in Department of Mathematics at Cukurova University. From March 2002 to September 2002, I was a visiting researcher at University of Iowa, USA to study in view of improving my research skills and to collect materials for my then ongoing research.

Gözlem No	x1	x2	x3	y	Gözlem No	x1	x2	x3	y
1	10.10	19.60	28.30	9.70	39	2.10	0.00	1.20	-0.70
2	9.50	20.50	28.90	10.10	40	0.50	2.00	1.20	-0.50
3	10.70	20.20	31.00	10.30	41	3.40	1.60	2.90	-0.10
4	9.90	21.50	31.70	9.50	42	0.30	1.00	2.70	-0.70
5	10.30	21.10	31.10	10.00	43	0.10	3.30	0.90	0.60
6	10.80	20.40	29.20	10.00	44	1.80	0.50	3.20	-0.70
7	10.50	20.90	29.10	10.80	45	1.90	0.10	0.60	-0.50
8	9.90	19.60	28.80	10.30	46	1.80	0.50	3.00	-0.40
9	9.70	20.70	31.00	9.60	47	3.00	0.10	0.80	-0.90
10	9.30	19.70	30.30	9.90	48	3.10	1.60	3.00	0.10
11	11.00	24.00	35.00	-0.20	49	3.10	2.50	1.90	0.90
12	12.00	23.00	37.00	-0.40	50	2.10	2.80	2.90	-0.40
13	12.00	26.00	34.00	0.70	51	2.30	1.50	0.40	0.70
14	11.00	34.00	34.00	0.10	52	3.30	0.60	1.20	-0.50
15	3.40	2.90	2.10	-0.40	53	0.30	0.40	3.30	0.70
16	3.10	2.20	0.30	0.60	54	1.10	3.00	0.30	0.70
17	0.00	1.60	0.20	-0.20	55	0.50	2.40	0.90	0.00
18	2.30	1.60	2.00	0.00	56	1.80	3.20	0.90	0.10
19	0.80	2.90	1.60	0.10	57	1.80	0.70	0.70	0.70
20	3.10	3.40	2.20	0.40	58	2.40	3.40	1.50	-0.10
21	2.60	2.20	1.90	0.90	59	1.60	2.10	3.00	-0.30
22	0.40	3.20	1.90	0.30	60	0.30	1.50	3.30	-0.90
23	2.00	2.30	0.80	-0.80	61	0.40	3.40	3.00	-0.30
24	1.30	2.30	0.50	0.70	62	0.90	0.10	0.30	0.60
25	1.00	0.00	0.40	-0.30	63	1.10	2.70	0.20	-0.30
26	0.90	3.30	2.50	-0.80	64	2.80	3.00	2.90	-0.50
27	3.30	2.50	2.90	-0.70	65	2.00	0.70	2.70	0.60
28	1.80	0.80	2.00	0.30	66	0.20	1.80	0.80	-0.90
29	1.20	0.90	0.80	0.30	67	1.60	2.00	1.20	-0.70
30	1.20	0.70	3.40	-0.30	68	0.10	0.00	1.10	0.60
31	3.10	1.40	1.00	0.00	69	2.00	0.60	0.30	0.20
32	0.50	2.40	0.30	-0.40	70	1.00	2.20	2.90	0.70
33	1.50	3.10	1.50	-0.60	71	2.20	2.50	2.30	0.20
34	0.40	0.00	0.70	-0.70	72	0.60	2.00	1.50	-0.20
35	3.10	2.40	3.00	0.30	73	0.30	1.70	2.20	0.40
36	1.10	2.20	2.70	-1.00	74	0.00	2.20	1.60	-0.90
37	0.10	3.00	2.60	-0.60	75	0.30	0.40	2.60	0.20
38	1.50	1.20	0.20	0.90					

Table A.1. Hawkins Bradu and Kass data

Gözlem No	y	x_1	x_2	x_3
1	42	80	27	89
2	37	80	27	88
3	37	75	25	90
4	28	62	24	87
5	18	62	22	87
6	18	62	23	87
7	19	62	24	93
8	20	62	24	93
9	15	58	23	87
10	14	58	18	80
11	14	58	18	89
12	13	58	17	88
13	11	58	18	82
14	12	58	19	93
15	8	50	18	89
16	7	50	18	86
17	8	50	19	72
18	8	50	19	79
19	9	50	20	80
20	15	56	20	82
21	15	70	20	91

Table A.2. Stackloss Data

Gözlem No	y	x_1	x_2	x_3	x_4	x_5
1	0.53	0.57	0.11	0.47	0.54	0.84
2	0.54	0.65	0.14	0.53	0.55	0.89
3	0.57	0.61	0.13	0.49	0.52	0.92
4	0.45	0.44	0.16	0.45	0.42	0.99
5	0.55	0.55	0.11	0.53	0.52	0.92
6	0.43	0.44	0.16	0.43	0.41	0.98
7	0.48	0.49	0.12	0.56	0.46	0.82
8	0.42	0.41	0.17	0.42	0.43	0.98
9	0.47	0.54	0.12	0.59	0.46	0.85
10	0.49	0.69	0.16	0.63	0.56	0.91
11	0.55	0.66	0.16	0.51	0.48	0.87
12	0.52	0.70	0.13	0.52	0.48	0.81
13	0.49	0.65	0.14	0.63	0.52	0.89
14	0.52	0.59	0.11	0.51	0.56	0.89
15	0.50	0.53	0.11	0.52	0.57	0.89
16	0.51	0.52	0.13	0.51	0.61	0.92
17	0.52	0.58	0.12	0.55	0.61	0.95
18	0.51	0.45	0.10	0.52	0.53	0.92
19	0.40	0.42	0.17	0.41	0.41	0.98
20	0.57	0.53	0.11	0.42	0.57	0.91

Table A.3. Woodgravity data

Gözlem No	x1	x2	x3	x4	y	Gözlem No	x1	x2	x3	x4	y
1	168	51	104	89.29	1720	45	178	49	99	85.39	1637
2	176	55	106	83.52	1698	46	186	49	110	78.49	1716
3	179	55	104	83.80	1727	47	173	57	92	80.35	1790
4	176	46	114	88.64	1739	48	168	56	117	90.48	1694
5	177	48	101	80.79	1622	49	172	52	107	87.79	1742
6	175	54	102	88.57	1684	50	185	47	106	82.70	1712
7	170	48	116	90.00	1583	51	175	53	103	79.43	1637
8	177	54	110	85.88	1697	52	196	55	108	73.68	1645
9	179	51	97	83.24	1711	53	181	54	111	79.01	1715
10	178	50	107	83.59	1672	54	177	51	108	80.79	1610
11	171	51	108	85.96	1617	55	173	54	93	84.39	1700
12	189	61	114	83.07	1772	56	171	47	100	88.89	1664
13	183	57	115	80.87	1792	57	184	46	103	83.70	1723
14	190	51	108	82.63	1696	58	178	59	100	79.78	1755
15	180	54	103	87.22	1656	59	174	54	110	83.33	1697
16	173	55	108	83.82	1770	60	181	55	106	75.69	1670
17	180	50	109	81.67	1617	61	1755	53	102	86.63	18
18	178	50	110	83.15	1679	62	1537	44	124	79.01	18
19	179	53	97	84.92	1631	63	1650	55	90	76.09	19
20	178	57	108	83.71	1814	64	1675	47	107	88.20	19
21	188	45	105	77.66	1668	65	1610	48	104	83.63	19
22	173	60	102	90.75	1796	66	175	44	111	86.29	1786
23	182	45	105	81.32	1620	67	183	53	112	79.23	1685
24	162	45	92	86.42	1587	68	180	49	97	81.11	1750
25	187	63	108	79.68	1735	69	187	50	115	79.68	1587
26	189	49	110	79.37	1670	70	181	59	112	85.64	1686
27	170	49	118	86.47	1672	71	176	51	102	86.36	1663
28	176	50	112	85.23	1660	72	185	58	101	79.46	1693
29	173	50	103	86.71	1610	73	191	49	105	80.10	1750
30	169	47	112	91.72	1773	74	179	45	103	78.21	1598
31	172	43	105	91.28	1571	75	184	49	113	84.78	1710
32	178	58	109	85.39	1763	76	172	56	105	87.79	1590
33	173	55	114	89.60	1635	77	175	58	105	85.14	1790
34	185	51	106	86.49	1613	78	179	53	103	87.15	1758
35	175	52	108	90.86	1690	79	177	48	105	85.31	1835
36	176	52	110	86.36	1770	80	167	52	107	95.21	1720
37	184	54	104	80.98	1771	81	179	57	114	85.47	1684
38	184	47	106	78.80	1640	82	170	55	103	88.82	1631
39	182	56	108	82.97	1770	83	170	56	105	87.06	1758
40	178	50	108	86.52	1740	84	182	46	108	84.32	1661
41	175	51	112	85.71	1683	85	173	57	92	80.35	1790
42	189	52	110	76.72	1696	86	185	55	107	85.41	1704
43	174	50	109	85.63	1730	87	189	50	110	74.60	1590
44	173	46	103	93.06	1722						

Table A.4. Buxton data

n	k	Cont. level	δ_R	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
					dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	3	1	0.10	0.03	0.92	0.13	0.49	0.01	0.70	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.70	0.01	0.17	0.00	0.56	0.36
60	6	10	3	1	0.02	0.03	0.96	0.15	0.47	0.02	0.00	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.58	0.02	0.08	0.00	0.90	0.14
40	2	20	3	1	0.20	0.04	0.85	0.19	0.46	0.03	0.08	0.02	1.00	0.09	1.00	0.08	0.00	0.00	0.00	0.00	0.19	0.00	0.07	0.64
60	6	20	3	1	0.07	0.04	0.90	0.18	0.40	0.07	0.00	0.02	1.00	0.08	1.00	0.08	0.00	0.00	0.00	0.01	0.11	0.00	0.10	0.32
40	2	10	4	1	0.35	0.02	1.00	0.10	1.00	0.01	1.00	0.00	1.00	0.05	1.00	0.06	0.00	0.00	0.98	0.00	0.84	0.00	0.59	0.34
60	6	10	4	1	0.14	0.03	1.00	0.11	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.06	0.00	0.00	0.98	0.01	0.86	0.00	0.92	0.14
40	2	20	4	1	0.61	0.03	0.99	0.15	0.99	0.02	0.99	0.01	1.00	0.09	1.00	0.12	0.00	0.00	0.05	0.00	0.81	0.00	0.08	0.63
60	6	20	4	1	0.31	0.04	1.00	0.14	0.99	0.02	0.34	0.01	1.00	0.09	1.00	0.11	0.00	0.00	0.00	0.00	0.84	0.00	0.11	0.32
40	2	10	3	2	0.04	0.02	0.88	0.11	0.49	0.01	0.73	0.01	1.00	0.05	1.00	0.04	0.00	0.00	0.70	0.01	0.10	0.00	0.98	0.33
60	6	10	3	2	0.01	0.02	0.94	0.12	0.49	0.02	0.01	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.65	0.02	0.04	0.00	1.00	0.14
40	2	20	3	2	0.04	0.04	0.81	0.15	0.49	0.01	0.50	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.12	0.00	0.11	0.00	0.85	0.26
60	6	20	3	2	0.02	0.04	0.91	0.15	0.49	0.02	0.12	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.05	0.01	0.06	0.00	0.94	0.08
40	2	10	4	2	0.13	0.02	0.99	0.06	1.00	0.01	1.00	0.00	1.00	0.05	1.00	0.04	0.03	0.00	0.97	0.00	0.79	0.00	0.99	0.33
60	6	10	4	2	0.03	0.02	1.00	0.07	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.04	0.01	0.00	0.95	0.01	0.81	0.00	1.00	0.14
40	2	20	4	2	0.14	0.04	0.98	0.11	1.00	0.01	1.00	0.01	1.00	0.05	1.00	0.04	0.00	0.00	0.29	0.00	0.74	0.00	0.90	0.26
60	6	20	4	2	0.05	0.04	0.99	0.10	1.00	0.02	0.54	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.17	0.00	0.78	0.00	0.96	0.08
Average Probability					0.14	0.03	0.94	0.12	0.73	0.02	0.56	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.44	0.00	0.45	0.00	0.68	0.28

Table A.6. Design matrix with detection and false alarm probabilities for regression outliers in multiple clouds at the centroid of x-space. A single cloud is placed δ_R off the regression surface. In the case of two clouds, one is placed at $+\delta_R$ and the other $-\delta_R$ off the regression plane.

n	k	Cont. level	δ_R	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
					dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	3	1	0.10	0.03	0.92	0.13	0.49	0.01	0.70	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.70	0.01	0.17	0.00	0.56	0.36
60	6	10	3	1	0.02	0.03	0.96	0.15	0.47	0.02	0.00	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.58	0.02	0.08	0.00	0.90	0.14
40	2	20	3	1	0.20	0.04	0.85	0.19	0.46	0.03	0.08	0.02	1.00	0.09	1.00	0.08	0.00	0.00	0.00	0.00	0.19	0.00	0.07	0.64
60	6	20	3	1	0.07	0.04	0.90	0.18	0.40	0.07	0.00	0.02	1.00	0.08	1.00	0.08	0.00	0.00	0.00	0.01	0.11	0.00	0.10	0.32
40	2	10	4	1	0.35	0.02	1.00	0.10	1.00	0.01	1.00	0.00	1.00	0.05	1.00	0.06	0.00	0.00	0.98	0.00	0.84	0.00	0.59	0.34
60	6	10	4	1	0.14	0.03	1.00	0.11	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.06	0.00	0.00	0.98	0.01	0.86	0.00	0.92	0.14
40	2	20	4	1	0.61	0.03	0.99	0.15	0.99	0.02	0.99	0.01	1.00	0.09	1.00	0.12	0.00	0.00	0.05	0.00	0.81	0.00	0.08	0.63
60	6	20	4	1	0.31	0.04	1.00	0.14	0.99	0.02	0.34	0.01	1.00	0.09	1.00	0.11	0.00	0.00	0.00	0.00	0.84	0.00	0.11	0.32
40	2	10	3	2	0.04	0.02	0.88	0.11	0.49	0.01	0.73	0.01	1.00	0.05	1.00	0.04	0.00	0.00	0.70	0.01	0.10	0.00	0.98	0.33
60	6	10	3	2	0.01	0.02	0.94	0.12	0.49	0.02	0.01	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.65	0.02	0.04	0.00	1.00	0.14
40	2	20	3	2	0.04	0.04	0.81	0.15	0.49	0.01	0.50	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.12	0.00	0.11	0.00	0.85	0.26
60	6	20	3	2	0.02	0.04	0.91	0.15	0.49	0.02	0.12	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.05	0.01	0.06	0.00	0.94	0.08
40	2	10	4	2	0.13	0.02	0.99	0.06	1.00	0.01	1.00	0.00	1.00	0.05	1.00	0.04	0.03	0.00	0.97	0.00	0.79	0.00	0.99	0.33
60	6	10	4	2	0.03	0.02	1.00	0.07	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.04	0.01	0.00	0.95	0.01	0.81	0.00	1.00	0.14
40	2	20	4	2	0.14	0.04	0.98	0.11	1.00	0.01	1.00	0.01	1.00	0.05	1.00	0.04	0.00	0.00	0.29	0.00	0.74	0.00	0.90	0.26
60	6	20	4	2	0.05	0.04	0.99	0.10	1.00	0.02	0.54	0.00	1.00	0.05	1.00	0.04	0.00	0.00	0.17	0.00	0.78	0.00	0.96	0.08
Average Probability					0.14	0.03	0.94	0.12	0.73	0.02	0.56	0.00	1.00	0.06	1.00	0.05	0.00	0.00	0.44	0.00	0.45	0.00	0.68	0.28

Table A.6. Design matrix with detection and false alarm probabilities for regression outliers in multiple clouds at the centroid of x-space. A single cloud is placed δ_R off the regression surface. In the case of two clouds, one is placed at $+\delta_R$ and the other $-\delta_R$ off the regression plane.

APPENDIX A2: Simulation Study Tables

n	k	Cont. level	δ_2	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
					dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	1	3	0.15	0.03	0.54	0.14	0.48	0.01	0.50	0.01	0.77	0.06	0.70	0.06	0.25	0.00	0.51	0.01	0.12	0.00	0.81	0.35
60	6	10	1	3	0.12	0.03	0.41	0.15	0.45	0.02	0.23	0.00	0.70	0.06	0.58	0.06	0.17	0.00	0.43	0.02	0.04	0.00	0.70	0.15
40	2	20	1	3	0.12	0.04	0.28	0.21	0.43	0.02	0.33	0.01	0.61	0.10	0.53	0.09	0.00	0.00	0.24	0.02	0.03	0.00	0.56	0.36
60	6	20	1	3	0.03	0.04	0.12	0.18	0.28	0.06	0.07	0.01	0.42	0.11	0.36	0.10	0.08	0.02	0.13	0.04	0.00	0.00	0.27	0.19
40	2	10	1	4	0.42	0.03	0.80	0.12	0.80	0.01	0.84	0.01	0.95	0.06	0.90	0.07	0.25	0.00	0.73	0.00	0.42	0.00	0.88	0.34
60	6	10	1	4	0.35	0.03	0.64	0.14	0.80	0.02	0.64	0.00	0.93	0.06	0.81	0.08	0.17	0.00	0.65	0.01	0.25	0.00	0.83	0.15
40	2	20	1	4	0.33	0.04	0.52	0.20	0.79	0.01	0.63	0.01	0.85	0.12	0.77	0.14	0.00	0.00	0.36	0.01	0.16	0.00	0.65	0.36
60	6	20	1	4	0.08	0.04	0.18	0.19	0.67	0.05	0.20	0.02	0.62	0.15	0.56	0.15	0.08	0.00	0.22	0.03	0.01	0.00	0.33	0.20
40	2	10	1	5	0.70	0.03	0.94	0.10	0.96	0.01	0.98	0.00	1.00	0.06	0.98	0.08	0.25	0.00	0.86	0.00	0.80	0.00	0.89	0.34
60	6	10	1	5	0.72	0.03	0.81	0.14	0.96	0.02	0.91	0.00	0.99	0.06	0.94	0.10	0.33	0.00	0.81	0.00	0.75	0.00	0.89	0.14
40	2	20	1	5	0.65	0.04	0.73	0.18	0.96	0.01	0.87	0.01	0.96	0.13	0.91	0.19	0.00	0.00	0.45	0.00	0.55	0.00	0.67	0.36
60	6	20	1	5	0.25	0.04	0.28	0.19	0.91	0.03	0.41	0.03	0.78	0.19	0.73	0.20	0.00	0.00	0.30	0.02	0.41	0.00	0.36	0.21
40	2	10	2	3	0.13	0.02	0.60	0.12	0.49	0.01	0.51	0.01	0.80	0.06	0.74	0.06	0.25	0.00	0.53	0.01	0.13	0.00	0.91	0.35
60	6	10	2	3	0.10	0.02	0.58	0.13	0.47	0.02	0.26	0.00	0.77	0.06	0.68	0.05	0.17	0.00	0.47	0.02	0.06	0.00	0.80	0.15
40	2	20	2	3	0.07	0.03	0.37	0.17	0.45	0.01	0.38	0.01	0.68	0.09	0.59	0.09	0.13	0.00	0.29	0.01	0.02	0.00	0.61	0.34
60	6	20	2	3	0.02	0.03	0.27	0.17	0.40	0.04	0.14	0.01	0.57	0.10	0.49	0.09	0.08	0.00	0.21	0.02	0.00	0.00	0.46	0.15
40	2	10	2	4	0.32	0.02	0.85	0.10	0.81	0.01	0.85	0.00	0.96	0.06	0.93	0.07	0.25	0.00	0.73	0.00	0.44	0.00	0.96	0.35
60	6	10	2	4	0.26	0.02	0.81	0.12	0.81	0.02	0.65	0.00	0.96	0.06	0.90	0.07	0.17	0.00	0.67	0.01	0.29	0.00	0.94	0.14
40	2	20	2	4	0.22	0.02	0.64	0.16	0.79	0.01	0.70	0.01	0.91	0.10	0.83	0.13	0.13	0.00	0.39	0.00	0.17	0.00	0.69	0.32
60	6	20	2	4	0.12	0.03	0.43	0.17	0.77	0.03	0.37	0.01	0.80	0.12	0.72	0.14	0.08	0.00	0.33	0.01	0.06	0.00	0.57	0.14
40	2	10	2	5	0.58	0.02	0.95	0.07	0.96	0.01	0.98	0.00	1.00	0.06	0.99	0.08	0.25	0.00	0.86	0.00	0.81	0.00	0.97	0.34
60	6	10	2	5	0.55	0.02	0.93	0.10	0.96	0.02	0.92	0.00	1.00	0.05	0.98	0.09	0.33	0.00	0.82	0.00	0.77	0.00	0.98	0.14
40	2	20	2	5	0.49	0.02	0.83	0.14	0.96	0.01	0.92	0.01	0.99	0.10	0.95	0.18	0.13	0.00	0.47	0.00	0.59	0.00	0.72	0.32
60	6	20	2	5	0.27	0.03	0.60	0.17	0.95	0.02	0.63	0.01	0.94	0.14	0.88	0.19	0.08	0.00	0.42	0.00	0.45	0.00	0.66	0.14
Average Probabilities					0.29	0.02	0.58	0.14	0.72	0.02	0.58	0.00	0.83	0.09	0.76	0.10	0.15	0.00	0.49	0.01	0.30	0.00	0.71	0.25

Table A.7. Design matrix with detection and false alarm probabilities for regression outliers with the regressor variables for the outliers determined from the median of the first three clean observations for each variable. In the case of two clouds, the regressor variables for the second cloud are determined from the median of the last three clean observations

n	k	Cloud(s) Number	Cont. Level	δ_R	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
					dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
60	6	rand	15	10	0.71	0.01	0.99	0.05	1.00	0.02	1.00	0.01	1.00	0.05	1.00	0.35	0.11	0.00	1.00	0.00	1.00	0.00	0.99	0.11
60	6	rand	30	10	0.03	0.00	0.94	0.09	1.00	0.01	0.90	0.21	1.00	0.65	1.00	0.69	0.00	0.00	0.07	0.00	0.87	0.00	0.90	0.05
60	6	1	15	10	1.00	0.02	1.00	0.05	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.32	0.00	0.00	1.00	0.00	1.00	0.00	0.54	0.16
60	6	1	30	10	0.00	0.03	1.00	0.15	0.99	0.02	0.99	0.25	1.00	0.70	1.00	0.73	0.00	0.00	0.00	0.11	0.98	0.01	0.00	0.54
60	6	1	15	10	1.00	0.02	1.00	0.05	1.00	0.02	1.00	0.00	1.00	0.05	1.00	0.33	0.00	0.00	1.00	0.00	1.00	0.00	0.50	0.17
60	6	1	30	10	0.00	0.03	1.00	0.15	0.97	0.03	0.97	0.34	1.00	0.68	1.00	0.69	0.00	0.00	0.00	0.12	0.97	0.01	0.00	0.53
40	2	1	15	5	0.72	0.03	0.86	0.16	0.96	0.01	0.95	0.00	0.99	0.07	0.96	0.14	0.17	0.00	0.68	0.00	0.69	0.00	0.80	0.33
60	6	1	15	10	1.00	0.03	0.88	0.15	1.00	0.02	1.00	0.01	1.00	0.07	1.00	0.38	0.11	0.00	0.94	0.00	0.95	0.00	0.54	0.18
40	2	2	15	10	1.00	0.02	1.00	0.05	1.00	0.01	1.00	0.01	1.00	0.06	1.00	0.34	0.17	0.00	0.96	0.00	1.00	0.00	0.88	0.29
60	6	2	15	5	0.40	0.02	0.79	0.14	0.96	0.02	0.80	0.00	0.99	0.07	0.94	0.14	0.11	0.00	0.63	0.00	0.62	0.00	0.78	0.13
40	2	1	30	10	0.00	0.04	0.89	0.23	0.99	0.01	0.83	0.38	0.99	0.63	1.00	0.64	0.00	0.00	0.03	0.05	0.87	0.01	0.49	0.49
60	6	1	30	5	0.00	0.06	0.12	0.26	0.42	0.20	0.05	0.12	0.53	0.32	0.54	0.29	0.00	0.02	0.00	0.11	0.20	0.00	0.20	0.31
40	2	2	30	5	0.03	0.02	0.71	0.24	0.94	0.02	0.71	0.06	0.90	0.31	0.87	0.31	0.00	0.00	0.13	0.01	0.41	0.00	0.55	0.37
60	6	2	30	10	0.04	0.04	0.66	0.26	0.99	0.03	0.59	0.31	0.99	0.61	0.99	0.60	0.00	0.02	0.02	0.09	0.86	0.00	0.46	0.25
Average Probabilities					0.42	0.02	0.84	0.14	0.94	0.03	0.84	0.12	0.95	0.30	0.95	0.42	0.04	0.00	0.46	0.03	0.81	0.00	0.54	0.27

Table A.8. Design matrix with detection capability and false alarm rates for high magnitude, high contamination runs for regression outliers in the interior of x-space

n	k	Cont. Level	δ_L	δ_R	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
						dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	2	3	1	0.60	0.03	1.00	0.10	0.28	0.03	0.22	0.02	0.07	0.09	0.03	0.07	0.00	0.00	0.59	0.02	0.06	0.00	0.65	0.40
60	6	10	2	3	1	0.06	0.03	1.00	0.06	0.06	0.04	0.01	0.00	0.00	0.08	0.00	0.06	0.00	0.01	0.09	0.07	0.04	0.00	0.38	0.23
40	2	20	2	3	1	0.20	0.04	1.00	0.14	0.03	0.05	0.00	0.03	0.00	0.12	0.00	0.09	0.00	0.01	0.00	0.08	0.06	0.02	0.40	0.50
60	6	20	2	3	1	0.06	0.05	1.00	0.10	0.00	0.05	0.00	0.01	0.00	0.10	0.00	0.06	0.00	0.02	0.00	0.09	0.03	0.04	0.22	0.32
40	2	10	5	3	1	0.39	0.04	1.00	0.03	0.11	0.01	0.00	0.01	0.00	0.07	0.00	0.05	0.00	0.01	0.42	0.04	0.01	0.00	0.65	0.40
60	6	10	5	3	1	0.11	0.03	1.00	0.02	0.00	0.02	0.00	0.00	0.00	0.06	0.00	0.04	0.00	0.01	0.24	0.06	0.00	0.00	0.38	0.23
40	2	20	5	3	1	0.04	0.06	1.00	0.06	0.00	0.02	0.00	0.01	0.00	0.08	0.00	0.05	0.00	0.01	0.12	0.06	0.01	0.03	0.55	0.40
60	6	20	5	3	1	0.00	0.06	1.00	0.05	0.00	0.03	0.00	0.00	0.00	0.07	0.00	0.04	0.00	0.02	0.01	0.09	0.00	0.04	0.32	0.27
40	2	10	2	5	1	0.98	0.03	1.00	0.05	0.90	0.02	0.93	0.01	0.84	0.12	0.83	0.12	0.04	0.00	0.97	0.00	0.80	0.00	0.82	0.37
60	6	10	2	5	1	0.26	0.03	1.00	0.05	0.32	0.06	0.30	0.01	0.00	0.12	0.00	0.10	0.00	0.01	0.34	0.05	0.20	0.00	0.63	0.20
40	2	20	2	5	1	0.50	0.03	1.00	0.09	0.31	0.09	0.00	0.09	0.05	0.22	0.05	0.18	0.00	0.01	0.08	0.07	0.60	0.00	0.57	0.44
60	6	20	2	5	1	0.15	0.04	1.00	0.08	0.00	0.10	0.00	0.02	0.00	0.15	0.00	0.12	0.00	0.02	0.01	0.09	0.25	0.04	0.41	0.26
40	2	10	5	5	1	0.59	0.03	1.00	0.03	0.29	0.02	0.10	0.02	0.00	0.10	0.00	0.08	0.00	0.01	0.82	0.01	0.11	0.01	0.86	0.37
60	6	10	5	5	1	0.31	0.03	1.00	0.02	0.02	0.03	0.00	0.00	0.00	0.08	0.00	0.05	0.00	0.01	0.53	0.03	0.03	0.00	0.59	0.19
40	2	20	5	5	1	0.22	0.06	1.00	0.06	0.01	0.03	0.00	0.02	0.00	0.11	0.00	0.08	0.00	0.01	0.47	0.04	0.06	0.04	0.78	0.33
60	6	20	5	5	1	0.00	0.06	1.00	0.05	0.00	0.04	0.00	0.00	0.00	0.08	0.00	0.05	0.00	0.02	0.10	0.08	0.01	0.05	0.55	0.20
40	2	10	2	3	2	0.76	0.03	1.00	0.05	0.61	0.02	0.68	0.01	0.55	0.09	0.50	0.09	0.21	0.00	0.53	0.01	0.14	0.00	0.71	0.39
60	6	10	2	3	2	0.30	0.03	1.00	0.03	0.34	0.03	0.11	0.00	0.35	0.09	0.24	0.08	0.00	0.01	0.10	0.05	0.01	0.00	0.55	0.19
40	2	20	2	3	2	0.41	0.04	0.98	0.07	0.36	0.02	0.19	0.04	0.40	0.16	0.37	0.13	0.00	0.00	0.01	0.04	0.03	0.00	0.55	0.45
60	6	20	2	3	2	0.14	0.04	1.00	0.05	0.04	0.04	0.00	0.01	0.00	0.11	0.00	0.09	0.00	0.01	0.00	0.05	0.01	0.00	0.52	0.18
40	2	10	5	3	2	0.51	0.03	1.00	0.02	0.32	0.01	0.08	0.01	0.04	0.08	0.02	0.06	0.00	0.00	0.53	0.02	0.00	0.00	0.82	0.38
60	6	10	5	3	2	0.39	0.03	1.00	0.01	0.08	0.02	0.01	0.00	0.00	0.07	0.00	0.05	0.00	0.01	0.28	0.04	0.00	0.00	0.67	0.18
40	2	20	5	3	2	0.39	0.03	1.00	0.02	0.06	0.02	0.00	0.01	0.00	0.08	0.00	0.07	0.00	0.00	0.12	0.03	0.01	0.00	0.76	0.34
60	6	20	5	3	2	0.37	0.05	1.00	0.02	0.00	0.03	0.00	0.00	0.00	0.06	0.00	0.05	0.00	0.01	0.01	0.05	0.00	0.00	0.65	0.16
40	2	10	2	5	2	0.99	0.03	1.00	0.04	0.98	0.01	0.99	0.00	0.91	0.10	0.68	0.15	0.46	0.00	0.94	0.00	0.88	0.00	0.84	0.36
60	6	10	2	5	2	0.47	0.03	1.00	0.03	0.80	0.03	0.77	0.00	0.49	0.14	0.48	0.13	0.01	0.01	0.19	0.05	0.20	0.00	0.62	0.18
40	2	20	2	5	2	0.55	0.04	1.00	0.06	0.87	0.02	0.18	0.09	0.49	0.26	0.49	0.23	0.00	0.01	0.01	0.05	0.36	0.00	0.68	0.40
60	6	20	2	5	2	0.16	0.05	1.00	0.05	0.31	0.06	0.00	0.04	0.05	0.17	0.04	0.15	0.00	0.01	0.00	0.06	0.20	0.00	0.51	0.17
40	2	10	5	5	2	0.51	0.03	1.00	0.01	0.64	0.02	0.45	0.02	0.22	0.12	0.14	0.10	0.01	0.00	0.83	0.01	0.02	0.00	0.93	0.36
60	6	10	5	5	2	0.36	0.03	1.00	0.01	0.21	0.03	0.03	0.00	0.00	0.08	0.00	0.06	0.00	0.01	0.55	0.02	0.02	0.00	0.79	0.17
40	2	20	5	5	2	0.37	0.03	1.00	0.02	0.23	0.02	0.00	0.03	0.00	0.12	0.00	0.10	0.00	0.00	0.30	0.02	0.03	0.00	0.89	0.30
60	6	20	5	5	2	0.33	0.05	1.00	0.02	0.00	0.03	0.00	0.01	0.00	0.08	0.00	0.06	0.00	0.01	0.05	0.05	0.03	0.00	0.76	0.13
Average Probabilities						0.35	0.03	0.99	0.04	0.25	0.03	0.15	0.01	0.13	0.10	0.12	0.08	0.02	0.00	0.28	0.04	0.13	0.00	0.62	0.29

Table A.9. Design matrix with detection and false alarm probabilities for high leverage regression outliers

n	k	Cont. Level	δ_L	δ_R	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		C&M		RWLS		HS93		BR	
						dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	2	5	2	0.99	0.03	0.51	0.14	0.98	0.01	1.00	0.00	0.92	0.10	0.72	0.16	0.48	0.00	0.91	0.00	0.94	0.00	0.97	0.36
60	6	10	2	5	1	0.25	0.03	0.03	0.16	0.33	0.06	0.29	0.01	0.00	0.12	0.00	0.10	0.00	0.01	0.37	0.05	0.45	0.00	0.63	0.20
40	2	20	5	5	2	0.30	0.05	0.05	0.19	0.28	0.02	0.00	0.03	0.00	0.13	0.00	0.11	0.00	0.01	0.55	0.00	0.04	0.00	0.84	0.33
60	6	20	5	5	2	0.32	0.05	0.04	0.20	0.01	0.03	0.00	0.01	0.00	0.08	0.00	0.06	0.00	0.01	0.51	0.00	0.01	0.00	0.56	0.19
Average Probabilities						0.46	0.04	0.15	0.17	0.40	0.03	0.32	0.01	0.23	0.10	0.18	0.10	0.12	0.00	0.58	0.01	0.36	0.00	0.75	0.27

Table A.10. Design matrix with detection capability and false alarm probabilities for high leverage regression outliers when the response variable is not unusual in y-space for the outlying observations.

APPENDIX A2: Simulation Study Tables

n	k	Cont. Level	Cloud(s) Number	P&Y		SM&R		LMS		LTS		M		OLS		CM		RWLS		HS93		BR	
				dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr	dc	fr
40	2	10	2	0.92	0.01	1.00	0.00	0.60	0.01	0.41	0.01	0.45	0.08	0.83	0.06	0.35	0.00	0.71	0.00	0.42	0.02	0.82	0.36
40	2	20	2	0.93	0.02	0.98	0.01	0.48	0.01	0.36	0.01	0.47	0.08	0.93	0.06	0.28	0.00	0.51	0.00	0.47	0.02	0.79	0.33
60	6	10	2	0.89	0.01	1.00	0.01	0.57	0.02	0.41	0.00	0.49	0.07	0.89	0.07	0.38	0.00	0.72	0.00	0.46	0.01	0.76	0.17
60	6	20	2	0.84	0.01	0.98	0.02	0.46	0.03	0.39	0.00	0.51	0.08	0.96	0.10	0.30	0.00	0.51	0.00	0.48	0.01	0.68	0.13
40	2	10	1	0.44	0.02	1.00	0.01	0.41	0.01	0.11	0.01	0.06	0.09	0.04	0.07	0.01	0.00	0.75	0.01	0.09	0.03	0.82	0.36
40	2	20	1	0.20	0.02	1.00	0.03	0.12	0.02	0.01	0.01	0.05	0.09	0.04	0.07	0.01	0.00	0.41	0.02	0.05	0.03	0.81	0.31
60	6	10	1	0.12	0.02	1.00	0.02	0.28	0.03	0.05	0.00	0.06	0.08	0.05	0.06	0.02	0.01	0.66	0.02	0.10	0.02	0.74	0.16
60	6	20	1	0.04	0.02	1.00	0.03	0.06	0.03	0.00	0.00	0.05	0.08	0.04	0.06	0.01	0.01	0.16	0.05	0.04	0.02	0.66	0.13
Average Probabilities				0.54	0.01	0.99	0.01	0.37	0.02	0.21	0.00	0.26	0.08	0.47	0.06	0.17	0.00	0.55	0.01	0.26	0.02	0.76	0.24

Table A.11. Dash and Dat Configuration

```
##chose
# methodnumber=1  for running scenario1  (sim1)
# methodnumber=2    for running scenario2 (sim2)
# methodnumber=3    for running scenario3(sim3)

# At the following two lines change the index of the bacon.scc in a suitable form
source("d:/baconnew1/bacon.ssc")
dll.load("d:/baconnew1/bacon.dll",c("@bacon$qpdpit2t1t1t2t2t1t1t2"))
# distance must be "mahalanobis" or "euclid"
distance <- "mahalanobis"
# alpha must be (0<alpah<1 )
alpha <-.1
# multiplier  must be >= 3
multiplier <- 3
n<-100
p<-4
var <- matrix(0, 4, 4)
ort <- matrix(0,1,4)
colvar <- matrix(0, 4, 1)
radyantop <- matrix(0, 1, 4)
bradyantop <- matrix(0, 1, 4)
bir <- matrix(1, 1, 4)
e<-diag(bir,nrow=100,ncol=4)
ozd<-matrix(0,4,100)
bozd<-matrix(0,4,100)
ortkolvar<-matrix(0,1,4)
a<-t(c(4,3,2,1))
tozd<-t(matrix(a,p,n))
ts<-1
set.seed(999)
methodnumber<-scan()
if (methodnumber==1)
    {
####################
###SCENARİO 1 ###
####################
repeat{
sim1<-rmvnorm(100,mean=c(0,0,0,0),cov=matrix(c(4,0,0,0,0,3,0,0,0,0,2,0,0,0,0,1),4,4))
pcsim1<-princomp(sim1)
scoresim1<-pcsim1[["scores"]]
v<-scoresim1
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
```

```
out <- bacon(sim1,distance=distance,alpha=alpha,multiplier=multiplier)
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=4)
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim1<-var(sim1)
eigensim1<-matrix(c(eigen(varsim1,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim1
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}
else if (methodnumber==2)
{
########################
###SCENARİO2 #########
########################
repeat{
sim2a<-rmvnorm(10,mean=c(0,0,0,0),cov=matrix(c(36,0,0,0,0,27,0,0,0,0,18,0,0,0,0,9),4,4))
sim2b<-rmvnorm(90,mean=c(0,0,0,0),cov=matrix(c(4,0,0,0,0,3,0,0,0,0,2,0,0,0,0,1),4,4))
sim2<-rbind(sim2a[[]],sim2b[[]])
pcsim2<-princomp(sim2)
scoresim2<-pcsim2[["scores"]]
v<-scoresim2
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
out <- bacon(sim2,distance=distance,alpha=alpha,multiplier=multiplier)
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
```

```
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=4)
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim2<-var(sim2)
eigensim2<-matrix(c(eigen(varsim2,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim2
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}
else if (methodnumber==3)
{
################
###SCENARİO3#
################
repeat{
sim3a<-rmvnorm(10,mean=c(0,0,0,10),cov=matrix(c(4,0,0,0,0,3,0,0,0,0,2,0,0,0,0,1),4,4))
sim3b<-rmvnorm(90,mean=c(0,0,0,0),cov=matrix(c(4,0,0,0,0,3,0,0,0,0,2,0,0,0,0,1),4,4))
sim3<-rbind(sim3a[[]],sim3b[[]])
pcsim3<-princomp(sim3)
scoresim3<-pcsim3[["scores"]]
v<-scoresim3
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
out <- bacon(sim3,distance=distance,alpha=alpha,multiplier=multiplier)
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=4)
```

```
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim3<-var(sim3)
eigensim3<-matrix(c(eigen(varsim3,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim3
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}
else if (methodnumber>=4)
{
    break
}
ozdo<-rowMeans(ozd)
bozdo<-rowMeans(bozd)
ozdort<-t(matrix(ozdo,4,100))
bozdort<-t(matrix(bozdo,4,100))
ortozd<-(t(ozd)-tozd)*(t(ozd)-tozd)
bortozd<-(t(bozd)-tozd)*(t(bozd)-tozd)
mse<-colMeans(ortozd)
bmse<-colMeans(bortozd)
radyanort<-radyantop*(1/100)
bradyanort<-bradyantop*(1/100)
```

```
##chose
# methodnumber=4  for running Scenario 4 (sim4)
# methodnumber=5  for running Scenario 5 (sim5)
# methodnumber=6  for running Scenario 6 (sim6)

# At the following two lines change the index of the bacon.scc in a suitable form
source("d:/baconnew1/bacon.ssc")
dll.load("d:/baconnew1/bacon.dll",c("@bacon$qpdpit2t1t1t2t2t1t1t2"))
# distance must be "mahalanobis" or "euclid"
distance <- "mahalanobis"
# alpha must be (0<alpah<1 )
alpha <-.1
# multiplier  must be >= 3
multiplier <- 3;
n<-100
p<-20
colvar <- matrix(0, 20, 1)
radyantop <- matrix(0, 1, 20)
bradyantop <- matrix(0, 1, 20)
bir <- matrix(1, 1, 20)
e<-diag(bir,nrow=100,ncol=20)
bozd<-matrix(0,20,100)
ortkolvar<-matrix(0,1,20)
ozd<-matrix(0,20,100)
a<-t(c(5,4,3,2,1,0.15,0.14,0.13,0.12,0.11,0.10,0.09,0.08,0.07,0.06,0.05,0.04,0.03,0.02,0.01))
tozd<-t(matrix(a,p,n))
ts<-1
set.seed(999)
methodnumber<-scan()
if (methodnumber==1)
    {
###################
###SCENARİO 4 ###
###################
repeat{
sim4<-rmvnorm(100,mean=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),cov=diag(c(5,4,3,2,1,0.15,
    0.14,0.13,0.12,0.11,0.10,0.09,0.08,0.07,0.06,0.05,0.04,0.03,0.02,0.01)))
pcsim4<-princomp(sim4)
scoresim4<-pcsim4[["scores"]]
v<-scoresim4
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
out <- bacon(sim4,distance=distance,alpha=alpha,multiplier=multiplier)
```

```
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=20)
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim4<-var(sim4)
eigensim4<-matrix(c(eigen(varsim4,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim4
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}

else if (methodnumber==5)
{
##################
###SCENARİO5 ###
##################
repeat{
sim5a<-
    rmvnorm(10,mean=c(0,0,0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0),cov=diag(c(5,4,3,2,1,0.15,0.14,
    0.13,0.12,0.11,0.10,0.09,0.08,0.07,0.06,0.05,0.04,0.03,0.02,0.01)))
sim5b<rmvnorm(90,mean=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),cov=diag(c(5,4,3,2,1,0.15,
0.14, 0.13,0.12,0.11,0.10,0.09,0.08,0.07,0.06,0.05,0.04,0.03,0.02,0.01)))
sim5<-rbind(sim5a[[]],sim5b[[]])
pcsim5<-princomp(sim5)
scoresim5<-pcsim5[["scores"]]
v<-scoresim5
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
out <- bacon(sim5,distance=distance,alpha=alpha,multiplier=multiplier)
```

```
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=20)
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim5<-var(sim5)
eigensim5<-matrix(c(eigen(varsim5,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim5
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}

else if (methodnumber==6)
{
###################
###SCENARİO 6#####
###################
repeat{
sim6a<-
    rmvnorm(10,mean=c(0,0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),cov=diag(c(5/20,4/20,3/20,2/20,
    1/20,(0.15)/20,(0.14)/20,(0.13)/20,(0.12)/20,(0.11)/20,(0.10)/20,(0.09)/20,(0.08)/20,(0.07)/20,
    (0.06)/20,(0.05)/20,(0.04)/20,(0.03)/20,(0.02)/20,(0.01)/20)))
sim6b<-rmvnorm(90,mean=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),cov=diag(c(5,4,3,2,1,0.15,
0.14,0.13,0.12,0.11,0.10,0.09,0.08,0.07,0.06,0.05,0.04,0.03,0.02,0.01)))
sim6<-rbind(sim6a[[]],sim6b[[]])
pcsim6<-princomp(sim6)
scoresim6<-pcsim6[["scores"]]
v<-scoresim6
vxe<-v*e
inorm<-1/sqrt(colSums(v*v))
cvxe<-colSums(vxe)
cosradyan<-cvxe*inorm
radyan<-acos(cosradyan)
radyantop<-radyan+radyantop
```

```
out <- bacon(sim6,distance=distance,alpha=alpha,multiplier=multiplier)
TX <- out[[1]]
a<-dim(TX)
pp<-a[[1]]
vTX<-var(TX)
ozvek<-matrix(c(eigen(vTX,only.values=F)$vectors),ncol=p)
ytb<-TX%*%ozvek
ee<-diag(bir,nrow=pp,ncol=20)
ytbxe<-ytb*ee
binorm<-1/(sqrt(colSums(ytb*ytb)))
ytbxet<-colSums(ytbxe)
bcosradyan<-ytbxet*binorm
bradyan<-acos(bcosradyan)
bradyantop<-bradyantop+bradyan
varsim6<-var(sim6)
eigensim6<-matrix(c(eigen(varsim6,only.values=T)$values),ncol=1)
ozd[,ts]<-eigensim6
vytb<-var(ytb)
eigenytb<-matrix(c(eigen(vytb,only.values=T)$values),ncol=1)
bozd[,ts]<-eigenytb
ts<-ts+1
if (ts>100)
break
}
}
else if (methodnumber>=7||methodnumber<=3)
{
    break
}
ozdo<-rowMeans(ozd)
bozdo<-rowMeans(bozd)
ozdort<-t(matrix(ozdo,20,100))
bozdort<-t(matrix(bozdo,20,100))
ortozd<-(t(ozd)-tozd)*(t(ozd)-tozd)
bortozd<-(t(bozd)-tozd)*(t(bozd)-tozd)
mse<-colMeans(ortozd)
bmse<-colMeans(bortozd)
radyanort<-radyantop*(1/100)
bradyanort<-bradyantop*(1/100)
```

```
#####################################################################
####################GENERATE DATA ###################################
#####################################################################

gentable6k6<-function(out1,out2,outshft11,outshft12,outshft13,outshft14,outshft15,
outshft16, outshft21,outshft22,outshft23,outshft24,outshft25, outshft26,yshift1,yshift2,n,k,x)
        {
            {
                outs<-out1+out2 # the total planted outliers
                first<-n-outs+1
                last<-n-outs
                nn<-n-out2+1
                nnn<-n-out2
                shift11<-7.5 + outshft11*4
                shift12<-7.5 + outshft12*4
                shift13<-7.5 + outshft13*4
                shift14<-7.5 + outshft14*4
                shift15<-7.5 + outshft15*4
                shift16<-7.5 + outshft16*4
                #   one<-rep(1,n)
                jin<-matrix(morm(last*6,7.5,4.0),ncol=6)
                yin<-apply(5*jin,1,sum) + matrix(morm(last),ncol=1)
                yin<-matrix(yin,ncol=1)
                joutl1<-matrix(shift11+runif(outs,0.0,0.25),ncol=1)
                joutl2<-matrix(shift12+runif(outs,0.0,0.25),ncol=1)
                joutl3<-matrix(shift13+runif(outs,0.0,0.25),ncol=1)
                joutl4<-matrix(shift14+runif(outs,0.0,0.25),ncol=1)
                joutl5<-matrix(shift15+runif(outs,0.0,0.25),ncol=1)
                joutl6<-matrix(shift16+runif(outs,0.0,0.25),ncol=1)
                j1<-cbind(joutl1,joutl2,joutl3,joutl4,joutl5,joutl6)
                x<-rbind(jin,j1) # the x values
                x<-as.matrix(x)
                j11<-x[first:nnn,]
                j12<-x[nn:n,]
                y1<-apply(5*j11,1,sum)+ yshift1
                y1<-matrix(y1,ncol=1)
                y2<-apply(5*j12,1,sum)+ yshift2
                y2<-matrix(y2,ncol=1)
                y<-rbind(yin,y1,y2)
                y<-matrix(y,ncol=1)
            }
                return(x,y)
        }
```

```
gentable6k2<-function(out1,out2,outshft11,outshft12,outshft21,outshft22,yshift1, yshift2,
n,k,x)
        {
            {
                outs<-out1+out2 # the total planted outliers
                first<-n-outs+1
                last<-n-outs
                nn<-n-out2+1
                nnn<-n-out2
                shift11<-7.5 + outshft11*4
                shift12<-7.5 + outshft12*4
                #   one<-rep(1,n)
                jin<-matrix(rnorm(last*2,7.5,4.0),ncol=2)
                yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
                yin<-matrix(yin,ncol=1)
                jout11<-matrix(shift11+runif(outs,0.0,0.25),ncol=1)
                jout12<-matrix(shift12+runif(outs,0.0,0.25),ncol=1)
                j1<-cbind(jout11,jout12)
                x<-rbind(jin,j1) # the x values
                x<-as.matrix(x)
                j11<-x[first:nnn,]
                j12<-x[nn:n,]
                y1<-apply(5*j11,1,sum)+ yshift1
                y1<-matrix(y1,ncol=1)
                y2<-apply(5*j12,1,sum)+ yshift2
                y2<-matrix(y2,ncol=1)
                y<-rbind(yin,y1,y2)
                y<-matrix(y,ncol=1)
            }
                return(x,y)
        }
gentable2a<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
        {
            {
                outs<-out1+out2 # the total planted outliers
                first<-n-outs+1 # observation number of first planted outlier
                last<-n-outs# observation number of the last clean case
                kmx<-k-x
                x1<-matrix(rnorm(k*last,7.5,4.0),ncol=k)
                x2<-matrix(runif(k*outs,7.375,7.625),ncol=k)
                x2<-x2+0.5
                x<-rbind(x1,x2)
                x<-as.matrix(x)
                y1<-apply(5*x[1:last,],1,sum) + matrix(rnorm(last),ncol=1)
                y1<-matrix(y1,ncol=1)
                y2<-apply(5*x[(first:(n-out2)),],1,sum)+yshift1 # responses for the
```

```
                                                                    #second cloud
                    y2<-matrix(y2,ncol=1)
                    nn<-n-out2+1
                    y3<-apply(5*x[nn:n,],1,sum)+yshift2 # responses for the second cloud
                    y3<-matrix(y3,ncol=1)
                    y<-rbind(y1,y2,y3)
                    y<-matrix(y,ncol=1)
                    xy<-cbind(x,y)
                }
                return(x,y,xy)
            }
    gentable5<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
                {
                    {
                    outs<-out1+out2 # the total planted outliers
                    first<-n-outs+1 # observation number of first planted outlier
                    last<-n-outs# observation number of the last clean case
                    kmx<-k-x
                    shift1<-7.5 + outshft1*4   # place cloud 1 at this location
                    shift2<-7.5 + outshft2*4   # place cloud 2 at this location
                    #   one<-rep(1,n)                    # some procedures need an intercept
                    jin<-matrix(rnorm(last*k,7.5,4.0),ncol=k) # predictors for clean cases
                    yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
                    yin<-matrix(yin,ncol=1)        # clean response values
                    j1<-matrix(shift1+runif(out1*k,0.0,0.25),ncol=k) # cloud 1 x values
                    j2<-matrix(shift2+runif(out2*k,0.0,0.25),ncol=k) # cloud 2 x values
                    x<-rbind(jin,j1,j2) # the x values
                    x<-as.matrix(x)
                    y1<-apply(5*j1,1,sum)+ yshift1 # responses for the first cloud
                    y1<-matrix(y1,ncol=1)
                    y2<-apply(5*j2,1,sum)+yshift2     # responses for the second cloud
                    y2<-matrix(y2,ncol=1)
                    y<-rbind(yin,y1,y2)
                    y<-matrix(y,ncol=1)
                }
                return(x,y)
            }
gentable2<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
                {
                    {
                    outs<-out1+out2 # the total planted outliers
                    first<-n-outs+1 # observation number of first planted outlier
                    last<-n-outs# observation number of the last clean case
                    kmx<-k-x
                    x1<-matrix(rnorm(k*last,7.5,4.0),ncol=k)
                    x2<-matrix(runif(k*outs,7.375,7.625),ncol=k)
                    x<-rbind(x1,x2)
```

```
            x<-as.matrix(x)
            y1<-apply(5*x[1:last,],1,sum) + matrix(rnorm(last),ncol=1)
            y1<-matrix(y1,ncol=1)
            y2<-apply(5*x[(first:(n-out2)),],1,sum)+yshift1   # responses for the
                                                              #second cloud
            y2<-matrix(y2,ncol=1)
            nn<-n-out2+1
            y3<-apply(5*x[nn:n,],1,sum)+yshift2 # responses for the second cloud
            y3<-matrix(y3,ncol=1)
            y<-rbind(y1,y2,y3)
            y<-matrix(y,ncol=1)
            xy<-cbind(x,y)
    }
        return(x,y,xy)
}
gendatamed1<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
    {
        {
            outs<-out1+out2
            last<-n-outs
            first<-n-outs+1
            lastm2<-last-2
            xin<-matrix(rnorm(last*k,7.5,4.0),ncol=k)
            xmed<-apply(xin[lastm2:last,],2,median)
            temp<-matrix(rnorm(outs*k,0,.05),ncol=k)
            xmedm<-xmed+temp
            xmedm<-matrix(xmedm,ncol=k,byrow=T)
            x<-rbind(xin,xmedm)
            yin<-apply(5*xin,1,sum) + matrix(rnorm(last),ncol=1)
            yout<-apply(5*xmedm,1,sum)+ matrix(rnorm(outs)+yshift1,ncol=1)
            y<-rbind(yin,yout)
        }
            return(x,y,xmed)
    }

gendatamed2<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
    {
            outs<-out1+out2
            last<-n-outs
            first<-n-outs+1
            lastm2<-last-2
            xin<-matrix(rnorm(last*k,7.5,4.0),ncol=k)
            xmed1<-apply(xin[lastm2:last,],2,median)
            temp<-matrix(rnorm(out1*k,0,.05),ncol=k)
            xmedm1<-xmed1+temp
            xmedm1<-matrix(xmedm1,ncol=k,byrow=T)
```

```
                xmed2<-apply(xin[1:3,],2,median)
                temp<-matrix(rnorm(out2*k,0,.05),ncol=k)
                xmedm2<-xmed2+temp
                xmedm2<-matrix(xmedm2,ncol=k,byrow=T)
                x<-rbind(xin,xmedm1,xmedm2)
                yin<-apply(5*xin,1,sum) + matrix(rnorm(last),ncol=1)
                yout1<-apply(5*xmedm1,1,sum)+matrix(rnorm(out1)+yshift1,ncol=1)
                yout2<-apply(5*xmedm2,1,sum)+matrix(rnorm(out2)+yshift2,ncol=1)
                y<-rbind(yin,yout1,yout2)
            }
                return(x,y,xmed1,xmed2)
        }

        genrand<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
        {
            {
                outs<-out1+out2
                first<-n-outs+1
                last<-n-outs
                one<-rep(1,n)
                x<-matrix(rnorm(k*n,7.5,4.0),ncol=k)
                yin<-apply(5*x[1:last,],1,sum) + matrix(rnorm(last),ncol=1)
                yin<-matrix(yin,ncol=1)
                x<-as.matrix(x)
                yout<-apply(5*x[first:n,],1,sum)+ yshift1
                yout<-as.matrix(yout,ncol=1)
                y<-rbind(yin,yout)
                y<-matrix(y,ncol=1)
            }
                return(x,y)
        }
dashdatdoubleclouds<-function(n,k,contamination){
                    {

                conobsnumber<-round(n*contamination)
                cleanobsnumber<-(n-conobsnumber)
                firstoutobsnumber<-n-conobsnumber+1
                x11<-matrix(runif(cleanobsnumber,-3,3),cleanobsnumber,1)
                x12<-matrix(runif(conobsnumber,19,20),conobsnumber,1)
                x1<-rbind(x11,x12)
                x2<-matrix(rnorm(n*(k-1)),nrow=n)
                x<-cbind(x1,x2)
                x<-as.matrix(x)
                y<-matrix(rnorm(n*1),nrow=n)
                t<-runif(n, min=0, max=1)
                for ( i in firstoutobsnumber:n)
                {
```

```
                    if (t[i]<0.5) {y[i]<-y[i]+6} else {y[i]<-y[i]-6}
                }
                    yx<-cbind(y,x)
    }
    return(x,y,yx)
                }

dashdatsinglecloud<-function(n,k,contamination){
                    {

                conobsnumber<-round(n*contamination)
                cleanobsnumber<-(n-conobsnumber)
                firstoutobsnumber<-n-conobsnumber+1
                x11<-matrix(runif(cleanobsnumber,-3,3),cleanobsnumber,1)
                x12<-matrix(runif(conobsnumber,19,20),conobsnumber,1)
                x1<-rbind(x11,x12)
                x2<-matrix(rnorm(n*(k-1)),nrow=n)
                x<-cbind(x1,x2)
                x<-as.matrix(x)
                y<-matrix(rnorm(n*1),nrow=n)
                firstoutobsnumber<-n-conobsnumber+1
                for (i in firstoutobsnumber:n)
                {y[i]<-y[i]+6}
                yx<-cbind(y,x)
    }
    return(x,y,yx)

                }
```

```
############################################################
############################################################
# methodnumber=1   for running   Chatterjee and Hadi
# methodnumber=2   for running   Hadi and Simonoff 1993
# methodnumber=3   for running   RWLS
# methodnumber=4   for runnnig   BACON regression
############################################################
############################################################

methodnumber<-scan()

if (methodnumber==1)
    {

  ########################################
  #####Chatterjeee and Machler Prog.  Code #######
  ########################################

    pppo<-function(N,out1,out2,xs1,xs2,yshift1,yshift2,n,k,xx)
        {
            {
                outs<-out1+out2
                first<-n-outs+1
                last<-n-outs
                sumfalse<-0
                sumdetect<-0
                for(i in 1:N)
                {
                    outliers<-matrix(0,n,1)
                    set.seed(i)
                    data<-gentable5(out1,out2,xs1,xs2,yshift1,yshift2,n,k,xx)
                y<-data$y
                y<-as.matrix(y)
                x<-data$x
                x<-as.matrix(x)
                hi<-hat(x)
                    hi<-as.matrix(hi)
                    hii<-hi
                    x<-cbind(1,x)
                    x<-as.matrix(x)
                    n<-nrow(x)
                    p<-ncol(x)
                    k<-p-1
                    B<-solve(t(x)%*%x)%*%(t(x)%*%y)
                    e1<-y-(x%*%B)
```

```
sse0<-colSums(e1*e1)
md<-p/n
md<-matrix(md,n,1)
maxx<-matrix(0,n,1)
for(i in 1:n)
        {
            maxx[i]<-max(hi[i],md[i])

        }

wi<-1/maxx
tol<-0.0001
maxiter<-25
iterRwls<-0
arc<-tol+1
ssej<-sse0
while ((iterRwls < maxiter) && (arc >tol) )
        {
            iterRwls<-iterRwls+1

            wii<-diag(n)# make nxn identity matrix
            wii<-as.matrix(wii)
            for (i in 1:n)
                    {
                        wii[i,i]<-wi[[i]]
                    }

            Bhat<-solve(t(x)%*%wii%*%x)%*%(t(x)%*%wii%*%y)
            yhat<-x%*%Bhat
            e<-y-yhat
            e<-as.matrix(e)
            ee<-e
            ssek<-colSums(e*e)
            abse<-abs(e)
            md<-median(abse)
            md<-matrix(md,n,1)
            maxx<-matrix(0,n,1)
            for (i in 1:n)
                        {
                            maxx[i]<-max(abse[i],md[i])
                        }
            wi<-(1-hii)*(1-hii)*(1/maxx)
            arc<-abs((ssek-ssej)/ssej)
            ssej<-ssek
    }#end while
            corfac<-1.38373
            mse<-corfac*sqrt((colSums(ee*ee))/(n-k-1))
```

```
                        ei<-ee*(1/(mse*(sqrt(1-hii))))
                        outliers<-matrix(0,n,1)
                            for (m in 1:n)
                                {
                                    alpha<-0.05
                                    if (abs(ei[m])>abs(qt(1 - alpha/2, n - k-1)))
                                        {
                                            outliers[m]<-1
                                        }
                                }
                        false<-sum(outliers[1:last])
                        sumfalse<-sumfalse+false
                        detect<-sum(outliers[first:n])
                        sumdetect<-sumdetect+detect
                        }#end for

        }
                pp<-sumdetect/(N*outs)
                po<-sumfalse/(N*last)
                return(pp,po)
            }
}

else if (methodnumber==2)
{
##########################################
######HADI and SIMONOFF prog. Code ######
##########################################

hsmeth1 <- function(X, y, initial="resid", alpha = 0.05)
{
# This S-Plus function computes estimates based on the procedure Method
# 1 of Hadi and Simonoff (1993) and finds the corresponding set of
# outliers.
#
# This code requires the Matrix library of S-Plus; before calling it,
# enter the command
#       library(Matrix)
# Input parameters
# y is an nx1 response vector
# X is an nxp matrix of explantatory variables (no column of 1's assumed in input)
# initial determines the diagnostic used to create the initial subset; choices
#   are "hat", "cook", or "resid"
# alpha is the significance level for the test in step 3.a. of the Main Algorithm
# Output parameters
# outliers: the set of outlier observations
# regline: lm object of rejection-plus-least squares regression estimate
```

```
# Reference: A.S. Hadi and J.S. Simonoff, "Procedures for the Identification of
#    Multiple Outliers in Linear Models" (1993), Journal of the American
#    Statistical Association, 88, 1264-1272.
    h <- hat(X)
    X <- cbind(1, X)
    n <- nrow(X)
    p <- ncol(X)
    beta <- solve(t(X) %*% X) %*% t(X) %*% y
    t1 <- rep(0, length = n)
    tt <- rep(0, length = n)
    if (initial=="hat") Ind <- order(h) else {
        e <- y - X %*% beta
        if (initial=="resid") Ind <- order(abs(e)) else {
            d <- (e * e * h)/((1 - h)^2)
            Ind <- order(d)}}
    medio <- floor((n + p - 1)/2)
    j <- p + 1
    bdet<-0
    while(bdet==0){
        Jnd <- Ind[1:j]
        Xs <- X[Jnd, ]
        ddet<-det(t(Xs) %*% Xs,logarithm=F)$modulus
        if (abs(ddet<1.e-13))  j<-(j+1) else bdet<-1}
# Beginning of while loop to find initial subset M
    while(j <= medio) {
# beginning of while loop for finding subset M
        Jnd <- Ind[1:j]
        Xs <- X[Jnd, ]
        ys <- y[Jnd]
        beta <- solve(t(Xs) %*% Xs) %*% t(Xs) %*% ys
        e <- y - X %*% beta
        sind <- sqrt(t(e[Jnd]) %*% e[Jnd]/(length(Jnd) - p))
        iXs <- solve(t(Xs) %*% Xs)
        h <- diag(X %*% iXs %*% t(X))
        t1[Ind[1:j]] <- e[Ind[1:j]]/sqrt(1 - h[Ind[1:j]])
          t1[Ind[(j+1):n]]<- e[Ind[(j + 1):n]]/sqrt(1 + h[Ind[(j + 1):n]])
        Ind <- order(abs(t1))
        j <- (j + 1)
    }
# end of while loop for finding subset M
    rr <- c(0)
    while((rr == 0) & (j <= (n - 1))) {
# The loop for The Main Algorithm
        Jnd <- Ind[1:j]
        Xs <- X[Jnd, ]
        ys <- y[Jnd]
        beta <- solve(t(Xs) %*% Xs) %*% t(Xs) %*% ys
```

```
        e <- y - X %*% beta
        sind <- sqrt(t(e[Jnd]) %*% e[Jnd]/(length(Jnd) - p))
        iXs <- solve(t(Xs) %*% Xs)
        for(i in (j + 1):n) {
            h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
            t1[Ind[i]] <- e[Ind[i]]/sqrt(1 + h[Ind[i]])
        }
        for(i in 1:j) {
            h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
            t1[Ind[i]] <- e[Ind[i]]/sqrt(1 - h[Ind[i]])
        }
        tt[Ind] <- t1[Ind]/sind
        Ind <- order(abs(tt))
        u <- sort(abs(tt))
        if(u[j + 1] > qt(1 - alpha/(2 * (j + 1)), j - p)) {
            rr <- 1
        }
        else {
            j <- j + 1
        }
    }
# End of the loop for The Main Algorithm step
    K <- (1:n)[abs(tt) > qt(1 - alpha/(2 * (j + 1)), j - p)]
    jim<-ifelse(abs(tt) > qt(1 - alpha/(2 * (j + 1)), j - p),1,0)
    y[K] <- NA
    X <- X[, 2:ncol(X)]
    out <- lm(y ~ X, na.action = na.omit)
    return(list(outliers = K, jim=jim, regline = out))
}
# Auxiliary function
if(exists("det", mode="function", where=1))
   rm(det)# just in case
  library(Matrix) #-- defines 'det'
  det.default <- function(x,...) det(as.Matrix(x),...)
pppo<-function(N,out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x,seedy)
{
    {
    outs<-out1+out2
    first<-n-outs+1
    last<-n-outs
    i<-1
    sumhs93<-0 # total number of planted outliers detected with hs97
    sumhs93f<-0 # total number of false alarms
    for (r in 1:N){
        ii<-seedy+i
        set.seed(ii)
        data<-gendatamed1(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
```

```
        j<-hsmeth1(data$x,data$y)
        hsout<-sum(j$jim[first:n])
        sumhs93<-sumhs93 + hsout
        hsfalse<-sum(j$jim[1:last])
        sumhs93f<-sumhs93f + hsfalse
        i<-i+1
    }   #end for
        pp<-sumhs93/(N*outs)
        po<-sumhs93f/(N*(n-outs))
        }
    return(pp,po)
}
}
else if (methodnumber==3)
{
    ##########################################
    ##############RWLS  Prog. Code ##############
    ##########################################
        # At the following two lines change the index of the bacon.scc in a suitable form
        source("d:/baconnew1/bacon.ssc")
        dll.load("d:/baconnew1/bacon.dll",c("@bacon$qpdpit2t1t1t2t2t1t1t2"))
        # distance must be "mahalanobis" or "euclid"
        distance <- "mahalanobis"
        # alpha must be (0<alpah<1 )
        alpha <-.1
        # multiplier  must be >= 3
        multiplier <-5
      pppo<-function(N,out1,out2,xs1,xs2,yshift1,yshift2,n,k,xx)
      {
        {
          outs<-out1+out2
          first<-n-outs+1
          last<-n-outs
          sumfalse1<-0
          sumdetect1<-0
          sumfalse2<-0
          sumdetect2<-0
          for(i in 1:N)
          {
            outliers1<-matrix(0,n,1)
            outliers2<-matrix(0,n,1)
            set.seed(i)
            cat("iteration ",i, "  ")
            data<-gendatamed1(out1,out2,xs1,xs2,yshift1,yshift2,n,k,xx)
            y<-data$y
            y<-as.matrix(y)
            x<-data$x
```

```
x<-as.matrix(x)
out <- bacon(x,distance=distance,alpha=alpha,multiplier=multiplier)
di<-out[["distance"]]
di<-as.matrix(di)
di<-di*di
x<-cbind(1,x)
B<-solve(t(x)%*%x)%*%(t(x)%*%y)
e1<-y-(x%*%B)
sse0<-colSums(e1*e1)
di<-di/colSums(di)
dii<-di #will use at the graph
md<-median(di)
md<-matrix(md,n,1)
maxx<-matrix(0,n,1)
for(i in 1:n)
        {
            maxx[i]<-max(di[i],md[i])
        }
di<-1/maxx
colsums<-colSums(di)
di<-di/colsums
wi<-di
tol<-0.0001
maxiter<-25
iterRwls<-0
arc<-tol+1
ssej<-sse0
while ((iterRwls < maxiter) && (arc >tol) )
        {
            iterRwls<-iterRwls+1
            wii<-diag(n)# make nxn identity matrix
            wii<-as.matrix(wii)
            for (i in 1:n)
                    {
                        wii[i,i]<-wi[[i]]
                    }
            Bhat<-solve(t(x)%*%wii%*%x)%*%(t(x)%*%wii%*%y)
            yhat<-x%*%Bhat
            e<-y-yhat
            e<-as.matrix(e)
            ee<-e
            e<-(e*e)/colSums(e*e)
            ssek<-colSums(e*e)
            abse<-abs(e)
            md<-median(e)
            md<-matrix(md,n,1)
            maxx<-matrix(0,n,1)
```

```
                    for (i in 1:n)
                        {
                            maxx[i]<-max(abse[i],md[i])
                        }
                    ai<-(1-di)*(1/maxx)
                    wi<-(ai*ai)/colSums(ai*ai)

                    arc<-abs((ssek-ssej)/ssej)
                    ssej<-ssek
                }#end while
                    p<-k+1
                cofac1<-2*(n+2*p)/(n-2*p)
                mse1<-sqrt(cofac1)*sqrt((colSums(ee*ee))/(n-p))
                hii<-hat(x)
                ei1<-ee*(1/(mse1*(sqrt(1-hii))))
                    for (m in 1:n)
                        {
                            alpha<-0.05
                            if (abs(ei1[m])>abs(qt(1 - alpha/2, n - k-1)))

                                {

                                    outliers1[m]<-1
                                }
                        }
                    false1<-sum(outliers1[1:last])
                    sumfalse1<-sumfalse1+false1
                    detect1<-sum(outliers1[first:n])
                    sumdetect1<-sumdetect1+detect1
                    }#end for
        }
            pp1<-sumdetect1/(N*outs)
            po1<-sumfalse1/(N*last)
        return(pp1,po1)
        }
}
else if (methodnumber==4)
{
########################################
####BACON Regression Prog. Code #########
########################################

#LINK BACON PROGRAM
# At the following two lines change the index of the bacon.scc in a suitable form
source("d:/baconnew1/bacon.ssc")
dll.load("d:/baconnew1/bacon.dll",c("@bacon$qpdpit2t1t1t2t2t1t1t2"))
# distance must be "mahalanobis" or "euclid"
```

```
distance <- "mahalanobis"
# alpha must be (0<alpah<1 )
alpha <-.1
# multiplier  must be >= 3
multiplier <- 3
####calculating the BACON distances "di"
hsbmh1 <- function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x,alpha=0.05,i,seedy)
{
# This S-Plus function computes estimates based on the procedure Method
# 1 of Hadi and Simonoff (1993) and finds the corresponding set of
# outliers.
#
# This code requires the Matrix library of S-Plus; before calling it,
# enter the command
#       library(Matrix)
# If you do not have access to the Matrix library, you will need to make a few
# changes. First, create the following determinant function:
# det <- function(M) Re(prod(eigen(M, only.values = T)$values))
# You will need to change one line in the hsbmh1 function:
# ddet <- det(t(Xs) %*% Xs, logarithm = F)$modulus
# to
# ddet <- det(t(Xs) %*% Xs)
# Input parameters
# y is an nx1 response vector
# X is an nxp matrix of explantatory variables (no column of 1's assumed in input)
# initial determines the diagnostic used to create the initial subset; choices
#   are "hat", "cook", or "resid"
# alpha is the significance level for the test in step 3.a. of the Main Algorithm
#
# Output parameters
# outliers: the set of outlier observations
# regline: lm object of rejection-plus-least squares regression estimate
#
# Reference: A.S. Hadi and J.S. Simonoff, "Procedures for the Identification of
#   Multiple Outliers in Linear Models" (1993), Journal of the American
#   Statistical Association, 88, 1264-1272.
#
                ii<-seedy+i
                set.seed(ii)
                data<-gentable7k2(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
                y<-data$y
                y<-as.matrix(y)
                x<-data$x
                x<-as.matrix(x)
                out <- bacon(x,distance=distance,alpha=alpha,multiplier=multiplier)
                di<-out[["distance"]]
                Xs<-out[["xclean"]]
```

```
            di<-as.matrix(di)
            Xs<-as.matrix(Xs)
            Ind<-order(di)
            X<-cbind(1,x)
            n <- nrow(X)
            p <- ncol(X)
            t1 <- rep(0, length = n)
            ti <- rep(0, length = n)
            tt <- rep(0, length = n)
            Xs<-cbind(1,Xs)
            j<- nrow(Xs)
            bdet<-0
            while (bdet==0 & j<=n)
                {
                    Jnd <- Ind[1:j]
                    Xs <- X[Jnd, ]
                    ddet<-det(t(Xs) %*% Xs,logarithm=F)$modulus
                    if (abs(ddet<1.e-13))  {j<-(j+1)} else {bdet<-1}
                }
                if ( bdet==0 && j==n )
                    {
                            i<-i+1
                            hsbmh11<-hsbmh1(out1,out2, outshft1, outshft2, yshift1, yshift2,
                                                                        n,k,x,alpha=0.05,i)
                    }
            ys<-y[Ind[1:j]]
            beta <- ginverse(t(Xs) %*% Xs) %*% t(Xs) %*% ys
            e <- y - X %*% beta
            h<-matrix(0,p,1)
                iXs <-ginverse(t(Xs) %*% Xs)
            for(i in (j + 1):n)
                {
                    h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
                    t1[Ind[i]] <- e[Ind[i]]/sqrt(1 + h[Ind[i]])
                }
            for(i in 1:j)
                {
                    h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
                    t1[Ind[i]] <- e[Ind[i]]/sqrt(1 - h[Ind[i]])
                }
        sind <- sqrt(t(e[Jnd]) %*% e[Jnd]/(length(Jnd) - p))
                    t1[Ind] <- t1[Ind]/sind
            Ind<-order(abs(t1))
            medio<-p*5
            j<-p+1
   while(j <= medio)
{
```

```
            # beginning of while loop for finding subset M
            bdet<-0
            while(bdet==0)
                {
                    Jnd <- Ind[1:j]
                    Xs <- X[Jnd, ]
                    ddet<-det(t(Xs) %*% Xs,logarithm=F)$modulus
                    if (abs(ddet<1.e-13))  j<-(j+1) else bdet<-1
                }

        ys <- y[Jnd]
        beta <- ginverse(t(Xs) %*% Xs) %*% t(Xs) %*% ys
        e <- y - X %*% beta
        iXs <- ginverse(t(Xs) %*% Xs)
        h <- diag(X %*% iXs %*% t(X))
        t1[Ind[1:j]] <- e[Ind[1:j]]/sqrt(1 - h[Ind[1:j]])
        t1[Ind[(j+1):n]]<- e[Ind[(j + 1):n]]/sqrt(1 + h[Ind[(j + 1):n]])
        sind <- sqrt(t(e[Jnd]) %*% e[Jnd]/(length(Jnd) - p))
        ti[Ind] <- t1[Ind]/sind

            Ind <- order(abs(ti))
            j <- (j + 1)
}
        j<-j-1
        Jnd<-Ind[1:j]
        longueur<-as.matrix(rep(c(0,1),n))
        jj<-2
        while(longueur[jj]!=longueur[jj-1])
        {
                bdet<-0
                while(bdet==0)
                        {
                            Jnd <- Ind[1:j]
                            Xs <- X[Jnd, ]
                            ddet<-det(t(Xs) %*% Xs,logarithm=F)$modulus
                            if (abs(ddet<1.e-13))  j<-(j+1) else bdet<-1

                        }
                    ys <- y[Jnd]
                    beta <- ginverse(t(Xs) %*% Xs) %*% t(Xs) %*% ys
                    e <- y - X %*% beta

                        iXs <-ginverse(t(Xs) %*% Xs)
                for(i in (j + 1):n)
                    {
                        h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
```

```
            t1[Ind[i]] <- e[Ind[i]]/sqrt(1 + h[Ind[i]])
        }
    for(i in 1:j)
        {
            h[Ind[i]] <- X[Ind[i], ] %*% iXs %*% X[Ind[i], ]
            t1[Ind[i]] <- e[Ind[i]]/sqrt(1 - h[Ind[i]])
        }
sind <- sqrt(t(e[Jnd]) %*% e[Jnd]/(length(Jnd) - p))

            ti[Ind] <- t1[Ind]/sind

            obs<-as.matrix(c(1:nrow(X)))
            mat<-cbind(y,X,obs,ti)
            mat.ord<-mat[order(mat[,ncol(mat)]),1:ncol(mat)]
            r<-nrow(Xs)
            #alpha<-0.05
            critique<-qt(alpha/2*(j+1),j-p)
            dif<-sign(critique-mat.ord[,p+3])
            if(sum(dif)==n)
            {i<-n}
                else
                {
                    i<-table(dif,dif)[2,2]}
                    Xs<-mat.ord[1:i,2:p]
                    ys<-mat.ord[1:i,1]
                    jj<-jj+1
                    longueur[jj]<-nrow(Xs)
                    if(sum(dif)==n)
                    {
                        outliers<-c("BACON detect no outliers")
                        nbr.out<-0
                    }
                        else
                        {
                            i<-table(dif,dif)[nrow(table(dif,dif)),ncol(table(dif,dif))]
                            outliers<-mat.ord[(i+1):n,p+2]
                            nbr.out<-length(outliers)
                        }
        }

    return(outliers,nbr.out,ti,Xs,ys,ii,p)
}
# Auxiliary function; omit these lines if you do not have access to the Matrix
# library, and create your own det() function
if(exists("det", mode="function", where=1))
   rm(det)# just in case
 library(Matrix) #-- defines 'det'
```

```
 det.default <- function(x,...) det(as.Matrix(x),...)
pppo<-function(N,out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x,seedy)
{
    {
        outs<-out1+out2
        first<-n-outs+1
        last<-n-outs
        sumfalse<-0
        sumdetect<-0
        i<-1
        for(r in 1:N)
        {
        outliers1<-matrix(0,n,1)
        hsbmh11<-hsbmh1(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x,alpha=0.05,i,seedy)
        ii<-hsbmh11[["ii"]]
        ti<-hsbmh11[["ti"]]
        ti<-as.matrix(ti)
        nbr.out<-hsbmh11[["nbr.out"]]
        nbr.out<-as.matrix(nbr.out)
        nbr.clean<-n-nbr.out
        p<-hsbmh11[["p"]]
        for (m in 1:n)
            {
                alpha<-0.05
                if (abs(ti[m])>abs(qt(1 - alpha/(2 * (nbr.clean + 1)), nbr.clean - p)))
                {
                outliers1[m]<-1
                }
            }
        false<-sum(outliers1[1:last])
        sumfalse<-sumfalse+false
        detect<-sum(outliers1[first:n])
        sumdetect<-sumdetect+detect
        i<-i+1
        } #end for
    }
    pp<-sumdetect/(N*outs)
    po<-sumfalse/(N*last)
    return(pp,po)
    }}
if (methodnumber>=5)
    {break
    }
```

```
############################
#configuration of Table 5.4.
############################
##pppo1<-pppo(500,2,2,0,0,3,0,40,2,2)#gendatamed1
##pppo2<-pppo(500,3,3,0,0,3,0,60,6,6) #gendatamed1
##pppo3<-pppo(500,4,4,0,0,3,0,40,2,2) #gendatamed1
##pppo4<-pppo(500,6,6,0,0,3,0,60,6,6) #gendatamed1
##pppo5<-pppo(500,2,2,0,0,4,0,40,2,2) #gendatamed1
##pppo6<-pppo(500,3,3,0,0,4,0,60,6,6) #gendatamed1
##pppo7<-pppo(500,4,4,0,0,4,0,40,2,2) #gendatamed1
##pppo8<-pppo(500,6,6,0,0,4,0,60,6,6) #gendatamed1
##pppo9<-pppo(500,2,2,0,0,5,0,40,2,2)#gendatamed1
##pppo10<-pppo(500,3,3,0,0,5,0,60,6,6) #gendatamed1
##pppo11<-pppo(500,4,4,0,0,5,0,40,2,2) #gendatamed1
##pppo12<-pppo(500,6,6,0,0,5,0,60,6,6) #gendatamed1
##pppo13<-pppo(500,2,2,0,0,3,3,40,2,2) #gendatamed2
##pppo14<-pppo(500,3,3,0,0,3,3,60,6,6) #gendatamed2
##pppo15<-pppo(500,4,4,0,0,3,3,40,2,2) #gendatamed2
##pppo16<-pppo(500,6,6,0,0,3,3,60,6,6) #gendatamed2
##pppo17<-pppo(500,2,2,0,0,4,4,40,2,2) #gendatamed2
##pppo18<-pppo(500,3,3,0,0,4,4,60,6,6) #gendatamed2
##pppo19<-pppo(500,4,4,0,0,4,4,40,2,2) #gendatamed2
##pppo20<-pppo(500,6,6,0,0,4,4,60,6,6) #gendatamed2
##pppo21<-pppo(500,2,2,0,0,5,5,40,2,2) #gendatamed2
##pppo22<-pppo(500,3,3,0,0,5,5,60,6,6) #gendatamed2
##pppo23<-pppo(500,4,4,0,0,5,5,40,2,2) #gendatamed2
##pppo24<-pppo(500,6,6,0,0,5,5,60,6,6) #gendatamed2

############################
#Configuration of Table 5.5.
############################
#pppo25<-pppo(500,4,5,0,0,10,0,60,6,6)#genrand
##pppo26<-pppo(500,9,9,0,0,10,0,60,6,6)#genrand
#pppo26a<-pppo(500,5,4,0,0,10,10,60,6,6)#gentable2
#pppo26b<-pppo(500,9,9,0,0,10,10,60,6,6)#gentable2
#pppo26c<-pppo(500,5,4,0,0,10,10,60,6,6)#gentable2a
#pppo26d<-pppo(500,9,9,0,0,10,10,60,6,6)#gentable2a
##pppo27<-pppo(500,3,3,0,0,5,0,40,2,2)#gendatamed1
##pppo28<-pppo(500,4,5,0,0,10,0,60,6,6)#gendatamed1
##pppo29<-pppo(500,3,3,0,0,10,10,40,2,2)#gendatamed2
##pppo30<-pppo(500,4,5,0,0,5,5,60,6,6)#gendatamed2
##pppo31<-pppo(500,6,6,0,0,10,0,40,2,2)#gendatamed1
##pppo32<-pppo(500,9,9,0,0,5,0,60,6,6)#gendatamed1
##pppo33<-pppo(500,6,6,0,0,5,5,40,2,2)#gendatamed2
##pppo34<-pppo(500,9,9,0,0,10,10,60,6,6)#gendatamed2
```

```
############################
#Configuration of Table 5.6.
############################
#Single cloud scenarious
 # #pppo35<-pppo(500,2,2,2,2,3,3,40,2,2)#gentable5
  # #pppo36<-pppo(500,3,3,2,2,3,3,60,6,6)#gentable5
  ##pppo37<-pppo(500,4,4,2,2,3,3,40,2,2)#gentable5
  # #pppo38<-pppo(500,6,6,2,2,3,3,60,6,6)#gentable5
   ##pppo39<-pppo(500,2,2,5,5,3,3,40,2,2)#gentable5
  # #pppo40<-pppo(500,3,3,5,5,3,3,60,6,6)#gentable5
  # #pppo41<-pppo(500,4,4,5,5,3,3,40,2,2)#gentable5
  # #pppo42<-pppo(500,6,6,5,5,3,3,60,6,6)#gentable5
#   pppo43<-pppo(500,2,2,2,2,5,5,40,2,2)#gentable5
#   pppo44<-pppo(500,3,3,2,2,5,5,60,6,6)#gentable5
#   pppo45<-pppo(500,4,4,2,2,5,5,40,2,2)#gentable5
 #  pppo46<-pppo(500,6,6,2,2,5,5,60,6,6)#gentable5
#   #pppo47<-pppo(500,2,2,5,5,5,5,40,2,2)#gentable5
#   #pppo48<-pppo(500,3,3,5,5,5,5,60,6,6)#gentable5
#   #pppo49<-pppo(500,4,4,5,5,5,5,40,2,2)#gentable5
 # #pppo50<-pppo(500,6,6,5,5,5,5,60,6,6)#gentable5

#two clouds scenarious

#  #pppo51<-pppo(500,2,2,2,2,3,5,40,2,2)#gentable5
#  #pppo52<-pppo(500,3,3,2,2,3,5,60,6,6)#gentable5
 # #pppo53<-pppo(500,4,4,2,2,3,5,40,2,2)#gentable5
 #  #pppo54<-pppo(500,6,6,2,2,3,5,60,6,6)#gentable5
 # #pppo55<-pppo(500,2,2,5,5,3,5,40,2,2)#gentable5
  # #pppo56<-pppo(500,3,3,5,5,3,5,60,6,6)#gentable5
   ##pppo57<-pppo(500,4,4,5,5,3,5,40,2,2)#gentable5
  # ppo58<-pppo(500,6,6,5,5,3,5,60,6,6)#gentable5
  # #pppo59<-pppo(500,2,2,2,2,5,7,40,2,2)#gentable5
  # #pppo60<-pppo(500,3,3,2,2,5,7,60,6,6)#gentable5
  # #pppo61<-pppo(500,4,4,2,2,5,7,40,2,2)#gentable5
  # #pppo62<-pppo(500,6,6,2,2,5,7,60,6,6)#gentable5
 #  #pppo63<-pppo(500,2,2,5,5,5,7,40,2,2)#gentable5
  # #pppo64<-pppo(500,3,3,5,5,5,7,60,6,6)#gentable5
 #  #pppo65<-pppo(500,4,4,5,5,5,7,40,2,2)#gentable5
  # #pppo66<-pppo(500,6,6,5,5,5,7,60,6,6)#gentable5

############################
#Configuration of Table 5.2.
############################
##pppo83<-pppo(500,2,2,0,0,3,0,40,2,2)#genrand
#pppo84<-pppo(500,3,3,0,0,3,0,60,6,6)#genrand
##pppo85<-pppo(500,4,4,0,0,3,0,40,2,2)#genrand
```

```
##pppo86<-pppo(500,6,6,0,0,3,0,60,6,6)#genrand
#pppo87<-pppo(500,2,2,0,0,4,0,40,2,2)#genrand
##pppo88<-pppo(500,3,3,0,0,4,0,60,6,6)#genrand
##pppo89<-pppo(500,4,4,0,0,4,0,40,2,2)#genrand
##pppo90<-pppo(500,6,6,0,0,4,0,60,6,6)#genrand
##pppo91<-pppo(500,2,2,0,0,5,0,40,2,2)#genrand
##pppo92<-pppo(500,3,3,0,0,5,0,60,6,6)#genrand
##pppo93<-pppo(500,4,4,0,0,5,0,40,2,2)#genrand
##pppo94<-pppo(500,6,6,0,0,5,0,60,6,6)#genrand

#########################
#Configuration of Table 5.3.
#########################
#pppo95<-pppo(500,2,2,0,0,3,3,40,2,2)#gentable2
#pppo96<-pppo(500,3,3,0,0,3,3,60,6,6)#gentable2
#pppo97<-pppo(500,4,4,0,0,3,3,40,2,2)#gentable2
#pppo98<-pppo(500,6,6,0,0,3,3,60,6,6)#gentable2
#pppo99<-pppo(500,2,2,0,0,4,4,40,2,2)#gentable2
#pppo100<-pppo(500,3,3,0,0,4,4,60,6,6)#gentable2
#pppo101<-pppo(500,4,4,0,0,4,4,40,2,2)#gentable2
#pppo102<-pppo(500,6,6,0,0,4,4,60,6,6)#gentable2
#pppo103<-pppo(500,2,2,0,0,3,-3,40,2,2)#gentable2
#pppo104<-pppo(500,3,3,0,0,3,-3,60,6,6)#gentable2
#pppo105<-pppo(500,4,4,0,0,3,-3,40,2,2)#gentable2
#pppo106<-pppo(500,6,6,0,0,3,-3,60,6,6)#gentable2
#pppo107<-pppo(500,2,2,0,0,4,-4,40,2,2)#gentable2
#pppo108<-pppo(500,3,3,0,0,4,-4,60,6,6)#gentable2
#pppo109<-pppo(500,4,4,0,0,4,-4,40,2,2)#gentable2
#pppo110<-pppo(500,6,6,0,0,4,-4,60,6,6)#gentable2

#########################
#Configuration of Table 5.7.
#########################
#pppo111<-pppo(500,2,2,2,-2,2,-2,5,-5,40,2,2)#gentable6k2
#pppo112<-pppo(500,3,3,2,-2,2,-2,2,-2,2,-2,2,-2,2,-2,5,5,60,6,6)#gentable6k6
#pppo113<-pppo(500,4,4,5,-5,5,-5,5,-5,40,2,2)#gentable6k2
#pppo114<-pppo(500,6,6,5,-5,5,-5,5,-5,5,-5,5,-5,5,-5,5,-5,60,6,6)#gentable6k6
```