CONTRIBUTIONS TO OUTLIER DETECTION METHODS:

SOME THEORY AND APPLICATIONS

by

YINAZE HERVE DOVOEDO

SUBHABRATA CHAKRABORTI, COMMITTEE CHAIR
B. MICHAEL ADAMS
BRUCE BARRETT
GILES D'SOUZA
JUNSOO LEE

A DISSERTATION

Submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in the Department of
Information Systems, Statistics, and Management
Science in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2011

ABSTRACT

Tukey's traditional boxplot (Tukey, 1977) is a widely used Exploratory Data Analysis (EDA) tools often used for outlier detection with univariate data. In this dissertation, a modification of Tukey's boxplot is proposed in which the probability of at least one false alarm is controlled, as in Sim et al. 2005. The *exact expression* for that probability is derived and is used to find the fence constants, for observations from any specified location-scale distribution. The proposed procedure is compared with that of Sim et al., 2005 in a simulation study.

Outlier detection and control charting are closely related. Using the preceding procedure, one- and two-sided boxplot-based Phase I control charts for individual observations are proposed for data from an exponential distribution, while controlling the overall false alarm rate. The proposed charts are compared with the charts by Jones and Champ, 2002, in a simulation study.

Sometimes, the practitioner is unable or unwilling to make an assumption about the form of the underlying distribution but is confident that the distribution is skewed. In that case, it is well documented that the application of Tukey's boxplot for outlier detection results in increased number of false alarms. To this end, in this dissertation, a modification of the so-called adjusted boxplot for skewed distributions by Hubert and Vandervieren, 2008, is proposed. The proposed procedure is compared to the adjusted boxplot and Tukey's procedure in a simulation study.

In practice, the data are often multivariate. The concept of a (statistical) depth (or equivalently outlyingness) function provides a natural, nonparametric, "center-outward" ordering of a multivariate data point with respect to data cloud. The deeper a point, the less outlying it is. It is then natural to use some outlyingness functions as outlier identifiers. A simulation study is

performed to compare the outlier detection capabilities of selected outlyingness functions available in the literature for multivariate skewed data. Recommendations are provided.

ACKNOWLEDGMENTS

I would like to extend my grateful thanks to my dissertation committee, specially the chair, as well as other professors, for their help and time. I also thank everyone who gave me emotional support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Brief Overview of Outlier Detection

As pointed out by Beckman and Cook (1983), no observation can be guaranteed to be a totally dependable manifestation of a phenomenon under consideration. However, observations which stand apart from the bulk of the data warrant attention. Many terms have been used in the literature to refer to such observations; some of them listed in Beckman and Cook (1983) are "outliers", "discordant observations", "rogue values", "contaminants", "surprising values", "mavericks", "dirty". The notion of outliers has been somewhat vague throughout the years. Beckman and Cook (1983) reported how Edgeworth (1887) defined such observations: "discordant observations may be defined as those which present the appearance of differing in respect of their law of frequency from other observations with which they are combined". Outliers undeniably have a long history.

It is apparent that if one could identify the "faulty" observations in a dataset, then one could better understand the phenomenon under study; however, if one could better understand the phenomenon under study, discordant observations could easily be identified and hence inference and predictions for the future could be improved. Thus, the study or the detection of outliers is like the "chicken-and-egg" problem as Hadi, Imon, and Werner (2009) put it. The following citation is due to Francis Bacon (1620), and was taken from Billor, Hadi and Velleman (2000); also cited in Serfling (2007):

"Whoever knows the ways of Nature will more easily notice her deviations; and, on the other hand, whoever knows her deviations will more accurately describe her way."

– Francis Bacon, 1620

Needless to say that the study of outliers has received considerable attention (see for example Tukey, 1977; Barnett, 1978; Hawkins, 1980; Davies and Gather, 1993; Barnett and Lewis, 1994; Schwertman, Owens, and Adnan, 2004; Hawkins, 2006; Schwertman and de Silva 2007; Cerioli 2010; Dang and Serfling, 2010) in the context of data analysis. This is simply because their non-identification or mis-identification can substantially influence the data analysis leading to distortion and possibly inaccurate conclusions. In some circumstances, however, the outliers themselves are of interest, as they may lead to new knowledge or discovery.

1.2 Univariate Outlier Detection

Among the more routinely used univariate outlier detection techniques, the boxplot is the most popular with practitioners. The boxplot is a versatile exploratory data analysis (EDA) tool, used for displaying and summarizing univariate data. It helps visualize the location, spread, and skewness of data distributions, along with unusual values or outliers. The most commonly used version of the boxplot is the one by Tukey (1977) and is referred to as the traditional boxplot. Let $X_i\ , i = 1,2,..n$ denote observations from a sample of size n and $X_{(i)}\ , i = 1,2,..n$ denote the corresponding order statistics. The lower and upper fences of the traditional boxplot are given by $LF = Q_1 - k(Q_3 - Q_1)$ and $UF = Q_3 + k(Q_3 - Q_1)$, where Q_1 and Q_3 are the so-called fourths (or quartiles) originally defined as follows: $Q_1 = X_{(f)}\ and\ Q_3 = X_{(n-f+1)}$ with $f = \frac{1}{2}\left[\frac{(n+3)}{2}\right]$ where $[.]$ is the greatest integer function, and n is the sample size. The fence constant k is commonly chosen to be either 1.5 or 3. Observations that fall either below the lower fence or above the upper fence are labeled outlying observations (or potential outliers).

Along another line of outlier detection, Rosner (1983) proposed the Generalized Extreme Studentized Deviate (GESD) procedure, when the underlying distribution of the data is normal. It is an iterative procedure that requires the user to specify the maximum number of outliers suspected. It consists of n tests with test statistics $R_1, R_2, \ldots, R_n$ where n is the suspected number of outliers. Brant (1990) compared this procedure with the traditional boxplot and concluded that suitably chosen versions of these rules perform comparably.

In this dissertation, the focus is on the traditional boxplot and its various modifications.

1.3 Higher Dimensional Outlier Detection

While univariate outlier detection methodologies are useful, in practice, often, observations are collected on more than one variable, which gives rise to the multivariate data, data that are correlated. The detection of outliers in multivariate data is a more challenging problem. This is more so since the data can be visualized in case of univariate, bivariate and perhaps trivariate data, there is virtually no visual support currently available when the dimension of the data is higher than three. Barnett and Lewis (1994) stated "As Gnanadesikan and Kettenring (1972) remark, a multivariate outlier no longer has a simple manifestation as an observation which 'sticks out at the end' of the sample. The sample has no 'end'! But, notably in bivariate data, we may still perceive an observation as suspiciously aberrant from the data mass ..."

Because of the practical importance of the problem, many procedures to detect unusual observations in multivariate (normal) data have been developed. An approach is to check for outliers in each variable (marginal distribution) using any univariate outlier detection method. However, it is easy to see that if such an approach is used, when the variables are correlated,

some unusual observations may be missed. To illustrate this, consider a set of bivariate data that consists of heights and shoe sizes and suppose that a case (a person) has a relatively low height but a relatively high shoe size. Such a case should be declared as an outlier when heights and shoe sizes are monitored jointly, but may fail to be detected if we were to check for outliers in the variables height and shoe size separately. Another problem with checking for outliers in the two variables separately is that we will declare a case with both a relatively large height and relatively large shoe size as an outlier while it is not one. Thus, this approach will tend to lead to too many false decisions. This is because if we monitor each variable separately, we do not take into account some of the overall structure of the data, in case of this example, the correlation between height and shoe size. Some outlier methodologies have been developed to solve this problem. Examples are the Relplot and Quelplot by Goldberg and Iglewicz (1992), in case of bivariate data.

1.4 Focus of Dissertation

1.4.1 A New Boxplot Procedure for Oulier Detection in Location-Scale Data

A closer examination of the traditional boxplot suggests that it is more appropriate for situations in which the underlying data follow a symmetric distribution. When the data are from a skewed distribution, the traditional boxplot with $k = 1.5$ will wrongly flag too many observations as outliers. For example for the Exponential (1) distribution, the probability that an observation will fall outside the fences is about 4.81%, which may be viewed as high. On the other hand, the traditional boxplot with $k = 3$ may fail to identify many outliers in most cases.

Sim, Gan and Chang (2005) proposed a boxplot-like procedure for identifying outliers in location-scale data. Their boxplot is similar to the traditional boxplot, but the fence constants $k's$

are chosen to control what is called *the some-outside rate per sample* (hereafter SORS) at a small prescribed level α. The SORS is the probability that at least one observation in an uncontaminated sample is (falsely) classified as outlier. This criterion was introduced by Hoaglin, Iglewicz, and Tukey (1986). Carling (2000) suggested that, in non-Gaussian cases, there might be some advantages in constructing the boxplot fences from the median (see literature review). He proposed a boxplot with fences of the form $LF = Q_2 - k(Q_3 - Q_1)$, $UF = Q_2 + k(Q_3 - Q_1)$ where k is a sample size dependent constant, and Q_2 is the median. Carling (2000) also showed that the right skewness increases the variance of the sample third quartile more than that of the median. Along this line Schwertman, Owens and Adnan (2004; hereafter SOA) suggested the use of the semi-interquartile ranges (SIQRs), as in Kimber (1990), instead of the interquartile range, in order to account for skewness. Specifically, Kimber (1990)'s fences are of the form $LF = Q_1 - kSIQR_L$ and, $UF = Q_3 + kSIQR_U$. SOA's are of the form: $= Q_2 - k_1SIQR_L,\ UF = Q_2 + k_1SIQR_U$, where k and k_1 are fence constants and $SIQR_L = Q_2 - Q_1,\ SIQR_U = Q_3 - Q_2$.

The boxplot proposed in this dissertation is similar to the one considered by SOA. In other words, the lower (respectively upper) fence is obtained by subtracting (adding) some multiple of the lower (upper) semi-interquartile range from (to) the median. However, while SOA found the fence constant by controlling *the outside rate per observation* (hereafter ORO, probability that one observation from an uncontaminated data is (falsely) classified as outlier), we control the SORS. This seems more meaningful because the practitioner is often interested in determining if there is at least one outlier, not if there is exactly one outlier. Also, SOA's mathematical development for finding the fence constants is approximate while we derive an exact expression that can be used to find the fence constants, following Sim, Gan and Chang

(2005). In addition, while SOA's procedure can be applied for normal or "near normal" data, the proposed procedure is applicable to data from any location-scale distribution, a further generalization. A simulation study is performed to compare the performance of the proposed procedure and Sim, Gan and Chang (2005)'s boxplot procedure for the Exponential distribution.

Next, the proposed boxplot procedure is used to develop phase I control charts for location with individual observations, when it is known that the data arise from a location-scale distribution. The resulting control charts can be used as tools for detection of out-of-control conditions, particularly with data from skewed distributions and are expected to be more robust than existing control charts based on the sample mean.

1.4.2 Application: Boxplot Based Phase I Control Charts for Individual Observations

One-sided and two-sided control charts for individual observations from location-scale distributions are derived from the boxplot procedure discussed in section 1.4.1. The performance of these control charts is investigated for robustness and shift detection capability.

Assuming that we have data from a process that is in-control, the center line (CL) of the control chart will be the median of the sample, since the fences of the proposed boxplot procedure are constructed from the median. The upper and lower control limits (UCL and LCL) of the two-sided chart will be respectively the upper and lower boxplot fences described above and given in Dovoedo and Chakraborti (2009). Because we control the SORS at some value α in the proposed boxplot, the "overall" *false alarm rate* (also called the *false alarm probability*) for the corresponding Phase I control chart, when the underlying parameters of the distribution are known, is α. Then, the "individual" false alarm rate is given by $\alpha_0 = 1 - (1 - \alpha)^m$ where m is the number of independent observations. This derived control chart can be applied to data

arising from any location-scale distribution including the popular normal and the exponential. Note that for skewed distributions, the lower and upper control limits are asymmetrically placed with respect to the center because of the use of semi-interquartile ranges and the fact that the lower and upper fence constants are allowed to be different. Jones and Champs (2002) proposed phase I individual control charts for exponential data. The performance of our boxplot-based control charts is investigated and compared to their charts.

1.4.3 A Modified Adjusted Boxplot (Distribution-free) Procedure for Skewed Distributions

Hubert and Vandervieren (2008) argued that most of the existing boxplot procedures mentioned above have some drawbacks. They showed through examples that the traditional boxplot is not appropriate for skewed data. Also, while it seems interesting (or useful) to take advantage of the knowledge about the underlying distribution when constructing boxplot procedures, it may be the case that such information is not available to the practitioners; all they know is the presence of skewness given the type of the problem being encountered. For that situation, Hubert and Vandervieren (2008) proposed an adjustment to the traditional boxplot (with fence constant $k = 1.5$) that includes a robust measure of skewness, the so-called *medcouple* (MC) – see more details in chapter 2 and chapter 5 –, in the determination of the whiskers (or equivalently, the fences). Their proposed boxplot procedure does not require knowledge of the form of the underlying distribution. The fences are of form: $[Q_1 - h_l(MC)IQR, Q_3 + h_u(MC)IQR]$ where $h_l(MC)$ and $h_u(MC)$ are two appropriately determined functions of the *medcouple*, MC, and IQR is the interquartile range. For reasons given in Carling (2000) and previously briefly mentioned, we consider a natural competitor to Hubert and Vandervieren (2008)'s boxplot, one with the fences given by:

$[Q_2 - k_l(MC)SIQR_l,\ Q_2 + k_u(MC)SIQR_u]$ where $k_l(MC)\ and\ k_u(MC)$ are two different and appropriately determined functions of the *medcouple*. This set of fences is also of interest for reasons mentioned later (see literature review, section 2.1.2). This new modified adjusted boxplot is proposed and its performance is studied in comparison with that of the adjusted boxplot by Hubert and Vandervieren (2008) both for contaminated (with outliers) and uncontaminated (without outliers) data.

1.4.4 Outlier Detection for Multivariate Skewed Data: A Comparative Study

As mentioned previously, multivariate outlier detection is a difficult problem. Some univariate outlier detection methods have been extended to the bivariate or multivariate outlier detection problems. See for example, Becketti and Gould (1987), Goldberg and Iglewicz (1992), Rousseeuw, Ruts and Tukey (1999). It is usually assumed that the data come from an elliptical distribution like the multivariate normal distribution. Exceptions exist, however. Hubert and Van der Veeken (2008) studied outlier detection in multivariate skewed data.

They applied the adjusted boxplot by Hubert and Vandervieren (2008) to some modification of the Stahel-Donoho outlyingness (see more details in section 2.2.3). Each multivariate observation is transformed into a univariate index called outlyingness that measures how outlying the observation is with respect to the data cloud. Observations with large outlyingness values correspond to potential outliers. A boxplot could be used on the outlyingness values to detect such observations.

As mentioned earlier, a modification of the adjusted boxplot by Hubert and Vandervieren (2008) is proposed, which we call the modified adjusted boxplot. Specifically, the boxplot by Hubert and Vandervieren (2008) have fences given by:

$$\begin{cases} [Q_1 - 1.5e^{-4MC}IQR, Q_3 + 1.5e^{3MC}IQR] \text{ if } MC > 0 \\ [Q_1 - 1.5e^{-3MC}IQR, Q_3 + 1.5e^{4MC}IQR] \text{ if } MC \leq 0 \end{cases}$$

In a simulation study, we could apply both the adjusted boxplot by Hubert and Vandervieren (2008) and our modified adjusted boxplot to the outlyingness values corresponding to various outlyingness functions. An outlyingness function is a function that maps each multivariate observation to its outlyingness index. A goal in this dissertation is to compare the outlier detection capabilities of some selected outlyingness functions in the existing literature in the context of multivariate skew-normal data. A p-dimensional random variable X is said to have a multivariate skew-normal distribution (Azzalini and Dalla Valle, 1996) with a vector of shape parameters $\boldsymbol{\alpha}$ with dependence parameter $\boldsymbol{\Omega}$, written $X \sim \mathcal{SN}_p(\alpha, \boldsymbol{\Omega})$, when its probability density function is of the form

$$f_p(\boldsymbol{x}) = 2\phi_p(\boldsymbol{x}, \boldsymbol{\Omega})\Phi(\alpha^T \boldsymbol{x}) ,$$

where $\phi_p(\boldsymbol{x}, \boldsymbol{\Omega})$ is the p-dimensional normal density with mean zero and correlation matrix $\boldsymbol{\Omega}$, and Φ is the standard normal distribution function.

Observe that when the vector of shape parameters $\boldsymbol{\alpha} = \mathbf{0}$ and dependence parameter $\boldsymbol{\Omega} = \boldsymbol{I_p}$ then the skew-normal distribution reduces to the multivariate standard normal distribution. Therefore, the term skew-normal distribution refers to a class of distribution that includes the standard normal distribution.

Random variates from skew-normal distributions are generated using the R package "sn" by Azzalini (2010).

This approach (of applying boxplots to outlyingness values) proves unproductive, so the $(1-\alpha)100^{th}$ perccentile of the outlyingness values for a specified small α is obtained from simulated, uncontaminated (without outliers) data from the skew-normal distribution under consideration. That percentile is subsequently used for outlier detection in contaminated (with

outliers) data: If an observation's outlyingness value falls above that percentile, that observation will be declared as potential outlier

1.5 Organization of the Dissertation

The dissertation is organized as follows. We present a brief review of univariate and multivariate outlier detection procedures in chapter 2. In chapter 3, we study a new set of boxplot fences for the location-scale family. An application of this boxplot procedure to construct a one-sided and a two-sided phase I control charts for individual observations from location-scale distributions is presented in chapter 4. A new skewness adjusted (distribution-free) boxplot procedure for univariate outlier detection is considered and studied in chapter 5. In chapter 6, a comparative study of outlier detection in multivariate skewed data is conducted and the results are presented. Finally, conclusions and some directions for future research are presented in chapter 7.

CHAPTER 2

LITERATURE REVIEW

2.1 Univariate Outlier Detection

The traditional boxplot (Tukey, 1977) is a very useful tool for many purposes. Its fences are defined by

$LF = Q_1 - k(Q_3 - Q_1)$ and $UF = Q_3 + k(Q_3 - Q_1)$ where Q_1 and Q_3 are the quartiles (see definition previously). The constant k is commonly chosen to be either 1.5 or 3.

Some authors have subsequently defined the quartiles in many ways. Frigge, Hoaglin, and Iglewicz (1989) compiled eight definitions of the lower quartile Q_1 in terms of the ordered observations and warned that users should keep the differences in mind when computing the quartiles in small samples.

The traditional boxplot can be used to visualize some characteristics (location, spread, skewness) of the data as well as for outlier detection. Its popularity stems from its simplicity. In addition, it does not use the few extreme observations in the construction of the fences, which makes the boxplot not suffer from the well-known phenomenon of "masking". Masking is the phenomenon by which the presence of some outliers makes each outlier difficult to detect. The traditional boxplot has a breakdown point of 25%. The breakdown point can be defined as the fraction of outliers in a sample that an estimator (or a procedure) can cope with or withstand (Hubert, 1981). In case of estimators, the breakdown point can also be defined as the smallest amount of contamination in the data that may cause an estimator to take arbitrarily large aberrant

values (Donoho and Huber, 1983). The use of the fence constant $k = 1.5$ or $k = 3$ in the traditional boxplot has been questioned by many authors. Some authors have shown that even when the data are normal, for some sample sizes, the probability of falsely identifying at least one outlier in an uncontaminated sample may be high. Others have pointed out that it is inappropriate to use the traditional boxplot (with $k = 1.5$) when the distribution of the data is skewed. This is because, as pointed out earlier (in chapter 1), this may result in too many false alarms.

2.1.1 Controlling False Alarm in the Traditional Boxplot

One objection to the traditional boxplot is that the fence constant $k = 1.5$ should not be a constant but should depend on the sample size. In other words, as Schwertman and de Silva (2007) pointed out, "a value that is judged extreme in a small data set might be expected in a much larger sample". Hoaglin, Iglewicz, and Tukey (1986) studied the performance of the traditional boxplot ($k = 1.5$) in a simulation study and found that the sample SORS α, which is the proportion of times there is at least one outlier, can be as high as 50% for uncontaminated Gaussian samples of size $n = 75$. As explained earlier, instead of controlling the ORO α_n, it might be more meaningful to control the SORS α. Along this line, for the traditional boxplot, Hoaglin and Iglewicz (1987) found approximate values of the fence constants *k* for three different definitions of quartiles when α is maintained at levels of 0.1 and 0.05, respectively. For this traditional boxplot, the fences are given by $LF = Q_1 - k_l(Q_3 - Q_1)$ and $UF = Q_3 + k_u(Q_3 - Q_1)$, where Q_1, Q_2 and Q_3 are defined previously. Sim Gan and Chan (2005) derived an exact expression that can be used to find the fence constants k_l and k_u for a given small α, and the sample size n of location-scale data. Following the same idea of controlling α, Barnerjee and

Iglewicz (2007) obtained the fence constants, for the traditional boxplot, appropriate for large samples (larger than 2000) for selected distributions: The normal, exponential, gamma, weibull and t distributions. The motivation behind the Banerjee and Iglewicz large sample procedure is that far larger data sets are encountered nowadays in a variety of fields like genetic, commercial research, and medical research to name few. The procedure is approximate. They also proposed approximate formulae for finding adjustment constants that should multiply the large sample fence constants to make them appropriate for smaller sample sizes data.

2.1.2 Accounting for Skewness in the Boxplot Procedures

Another objection to the traditional boxplot is that, while not specifically assumed, the boxplot appears to have been designed for situations in which the underlying distribution is more or less symmetric. When the underlying distribution is skewed, applying the traditional boxplot with fence constant $k = 1.5$ may result in increased false alarms. For example, as pointed out before, the probability of exceedance for a normal population is approximately 0.7% (0.035% for each tail). The probability of lower (respectively upper) exceedance is the probability that a random observation will fall either below (respectively above) the lower (respectively upper) fence in an uncontaminated population. The probability of exceedance is the sum of the previous two probabilities. For the $Exponential(1)$ and $Chi\text{-}square(1)$ populations, the probability of lower exceedance is the same, zero, but the probabilities of upper exceedance are respectively approximately 4.81% and 7.56%. To allow the traditional boxplot to accommodate skewed distributions, boxplot with fences $LF = Q_1 - 2kSIQR_l, UF = Q_3 + 2kSIQR_u$, where $SIQR_l = Q_2 - Q_1$ and $SIQR_u = Q_3 - Q_2$, with $k = 1.5$ has been suggested by Kimber (1990).

For distributions other than the normal (the non-Gaussian case), Carling (2000), compared the traditional boxplot which yields the "Tukey's rule", with the upper fence: $c_1^U = Q_3 + k(Q_3 - Q_1)$ with the following median based boxplot proposed by Carling (2000), which yields the "median rule", with upper fence: $c_2^U = Q_2 + k(Q_3 - Q_1)$. Notice that the difference between the two fences is that one is constructed from the sample median while the other is constructed from the third sample quartile. Carling (2000) reached the conclusion that with respect to the two criteria, "resistance" and "efficiency", the median rule performs better than Tukey's rule. Roughly speaking, a rule is said to be resistant when it does not suffer from masking. The class of the Generalized Lambda Distribution, which allows the skewness and kurtosis to be varied one at a time, was used to compare the resistances of the two rules. An operational definition of resistance given by Carling (2000) is $|c^U(r_\infty, p) - c^U(r_\infty, p = 0)|$ where $c^U(r_\infty, p)$ is the population upper fence for a p-contaminated (where p is a small percentage) large sample with the fence constant chosen such that the large sample outside rates per sample are r_∞ $(= 0.01, 0.02, 0.03, \dots, 0.2)$. Thus, roughly speaking, resistance is the deviation in upper fences obtained from a p-contaminated sample with respect to a non-contaminated counterpart. Carling (2000) found that the deviation for Tukey's rule is higher than that for the median rule, which means that the median rule is more resistant. Carling (2000) also studied the Asymptotic Relative Efficiency, $ARE = V(c_2^U)/V(c_1^U)$ of the cutoffs (or fences) of the two rules, where $V(c_1^U)$ and $V(c_2^U)$ are the asymptotic variances of the fences of Tukey's rule and the median rule respectively. The ARE was derived using a result that gives the distribution of the sample vector $(Q_1, Q_2, Q_3)'$ provided in David (1981). Carling (2000) showed that the relative efficiency is less than one (in almost all cases) and it decreases with higher amount of skewness in the distribution. This suggests, as Carling (2000) pointed out, that the

right skewness increases the sample variation for the third quartile more than that for the median. One of Carling (2000)'s conclusions was that in almost all cases (except for symmetric distributions with low kurtosis), the median rule has an equal or smaller variance than the Tukey rule and hence should be preferable.

The preceding discussion suggests the consideration of the boxplot with fences: $LF = Q_2 - k_l SIQR_l$ and $UF = Q_2 + k_u SIQR_u$. In this dissertation, we study such fences. The fence constants are obtained by controlling the SORS at some small value, say, α. An exact expression for the SORS is derived for the location-scale family of distributions. This family is rich and includes the popular normal, exponential, and logistic distributions among others. It should be noted that these fences have been studied by Schwertman, Owen and Adnan (2004) with $k_l = k_u = k$. However, (i) their theoretical development is approximate and covers only normal and "near normal" distribution and (ii) they control the ORO.

2.1.3 Adjusting for Skewness in the Boxplot Procedures: Another Approach

2.1.3.1 A New Measure of Skewness: The Medcouple

As noted earlier, detection of outliers in data from skewed distributions is more challenging. The classical skewness coefficient b_1 of a univariate data set $X_n = \{x_1, x_2, \dots, x_3\}$ is defined by $b_1 = m_3(X_n)/m_2(X_n)^{3/2}$, where $m_j(X_n)$ represents the jth sample central moment. The coefficient b_1 has a breakdown point of zero since it is strongly affected by even a single outlier.

Brys, Hubert, and Struyf (2003) considered a new measure of skewness, the medcouple (MC), that is robust to outliers. Assume that $X_n = \{x_{(1)}, x_{(2)}, \dots, x_{(n)}\}$ is a *sorted* set of n

independently sampled observations from a continuous univariate distribution F. The sample *medcouple* MC is defined as:

$$MC = \underset{x_{(i)} \le med_n \le x_{(j)}}{\text{med}} h\left(x_{(i)}, x_{(j)}\right),$$

where med_n is the median of the sample X_n. When $x_{(i)} \neq x_{(j)}$, that is when $x_{(i)}$ and $x_{(j)}$ are not tied (to the median), the function h is defined by:

$$h\left(x_{(i)}, x_{(j)}\right) = \frac{\left(x_{(j)} - med_n\right) - \left(med_n - x_{(i)}\right)}{x_{(j)} - x_{(i)}}.$$

For the case when $x_{(i)} = x_{(j)} = med_n$, that is there are observations tied to the median, the function h is defined as follows. If $m_1 < m_2 < \cdots < m_k$ are the indices of observations tied to the median med_n (i.e., $x_{m_l} = med_n$), for all $l = 1, 2, .., k$:

$$h\left(x_{(i)}, x_{(j)}\right) = \begin{cases} -1 & if\ i + j - 1 < k \\ 0 & if\ i + j - 1 = k \\ +1 & if\ i + j - 1 > k \end{cases}.$$

Brys, Hubert and Struyft (2003) also showed that the population form of the medcouple, defined for any continuous cdf F with median m_F is given by $MC_F = H_F^{-1}(0.5)$, where, for $u \in [-1,1]$,

$$H_F(u) = 4 \int_{m_F}^{+\infty} F\left(\frac{x_2(u-1) + 2m_F}{u+1}\right) dF(x_2).$$

They showed that the breakdown point of MC to be 25%, meaning that it can resist up to 25% outliers in the data. They also showed that the MC possesses the natural requirements of a skewness measure as defined by van Zwet (1964) and Oja (1981). For example, the MC is location and scale invariant; when F is symmetric, $MC_F = 0$.

In Brys, Hubert, and Struyf (2003), the medcouple was compared to other measures of skewness, including the octile skewness (OS) and the quartile skewness (QS); the latter is also

known as the Bowley coefficient (Bowley, 1920; Moors et al., 1996). The OS and QS are part of a class of measures of skewness suggested by Hinkley (1975):

$$\frac{(Q_{1-p} - Q_{0.5}) - (Q_{0.5} - Q_p)}{Q_{1-p} - Q_p}$$

where Q_p $(0 < p < 1)$ is the p^{th} percentile of univariate data sampled from a continuous distribution.

The OS and QS correspond to $p = 0.25$ and 0.125, respectively. Among the measures of skewness, the MC was found to be the overall winner. They found that the MC can detect skewness in the data and is not highly affected by the presence of outliers. Brys, Hubert, and Struyf (2004) stated specifically that the MC combines the strength of OS (breakdown point 12.5%) and the QS (breakdown point 25%) in that it has the sensitivity of the OS to detect skewness and the robustness of QS towards outliers.

2.1.3.2 Adjusted Boxplot Procedures for Skewed Distributions

Hubert and Vandervieren (2008) incorporated the medcouple (MC) into the traditional boxplot so that it can better accommodate skewed distributions. Their aim was to construct a boxplot that does not depend on sample size and without making any assumptions about the underlying distribution, except that the shape of the distribution is skewed. Their motivation was that the traditional boxplot is not appropriate in this case and Kimber (1990)'s SIQR boxplot with fences $[Q_1 - 3SIQR_L, Q_3 + 3SIQR_U]$ does not work sufficiently well for skewed distributions. Hubert and Vandervieren (2008) considered boxplots with fences of the form $[Q_1 - h_l(MC)IQR,\ Q_3 + h_u(MC)IQR]$ where $h_l(MC)\ and\ h_u(MC)$ are fence "constants" adjusted for skewness. They studied three simple models in order to determine the functions $h_l\ and\ h_u$.

(1) Linear model:

$$\begin{cases} h_l(MC) = 1.5 + aMC \\ h_u(MC) = 1.5 + bMC \end{cases}$$

(2) Quadratic model:

$$\begin{cases} h_l(MC) = 1.5 + a_1 MC + a_2 MC^2 \\ h_u(MC) = 1.5 + b_1 MC + b_2 MC^2 \end{cases}$$

(3) Exponential model:

$$\begin{cases} h_l(MC) = 1.5e^{a_3 MC} \\ h_u(MC) = 1.5e^{b_3 MC} \end{cases},$$

with $a, a_1, a_2, a_3, b, b_1, b_2, b_3 \in \mathbb{R}$.

The overall winner turned out to be the exponential model and the resulting boxplot fences are $[Q_1 - 1.5e^{-4MC}IQR,\ Q_3 + 1.5e^{3MC}IQR]$ when $MC > 0$ (positively skewed distributions) and $[Q_1 - 1.5e^{-3MC}IQR, Q_3 + 1.5e^{4MC}IQR]$ when $MC \leq 0$ (negatively skewed distributions).

A natural competitor to the previous boxplot is a boxplot with fences given by $[Q_2 - h_l(MC)SIQR_L, Q_2 + h_u(MC)SIQR_L]$. In this dissertation, such boxplot fences is considered and their properties and performances is explored. The performance comparisons include the proposed boxplot, the boxplot by Hubert and Vandervieren (2008), and the traditional boxplot.

2.2 Multivariate Outlier Detection

Analysis of multivariate data is important in many practical situations. Detection of outliers in higher dimensions is challenging for several reasons. These include the difficulty of visualization, and the lack of natural ordering of observations. Moreover it doesn't seem appropriate to look at each dimension and apply univariate outlier detection methods since the variables are often correlated and what might seem like outliers on the univariate scale (marginal

distribution) might not necessarily be outliers on the multivariate scale and vice versa. Higher dimensional outlier identification/detection can be classified into two major approaches: Distance-based methods and projection pursuit methods. A more general framework is the "new" nonparametric depth-based outlier detection approach that has been on the rise over the last two decades.

2.2.1 Distance Based Outlier Detection

The main idea behind distance based outlier detection methods is to calculate the distance of each multivariate observation from the "center" of the data and order these scalar quantities. The farthest points can potentially be outliers. One of the earliest distance-based methods computes the classical Mahalanobis distance (Mahalanobis, 1936) for each multivariate observation $x_i \in \mathbb{R}^p$ $(1 = 1,2, \dots, n)$. The classical Mahalanobis distance of observation x_i with respect to a "center" of some multivariate data $\boldsymbol{X}$ is:

$$CMD_i = \sqrt{(x_i - \boldsymbol{T})^T \boldsymbol{C}^{-1}(x_i - \boldsymbol{T})}$$

where $\boldsymbol{T}$ and $\boldsymbol{C}$ measure the center and the spread of the data $\boldsymbol{X}$ and are taken as the usual sample mean and the sample covariance matrix.

Assume that the multivariate data $\boldsymbol{X}$ represents n observations (sample measurements) on p variables:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}.$$

Let's adopt the following notation:

$\boldsymbol{x_k} = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{nk} \end{bmatrix}$; $\bar{x}_k = \frac{1}{n}\sum_{j=1}^{n} x_{jk}$; $s_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$. Assuming $\boldsymbol{T}$ to be the sample mean and $\boldsymbol{C}$ to be the sample covariance matrix, $\boldsymbol{T}$ and $\boldsymbol{C}$ are given respectively by:

$$\boldsymbol{T} = \bar{\boldsymbol{X}} = \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_p \end{bmatrix},$$

and

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & \cdots & s_{1p} \\ \vdots & \ddots & \vdots \\ s_{p1} & \cdots & s_{pp} \end{bmatrix}.$$

The sample covariance matrix $\boldsymbol{S}$ can also be computed directly as follows:

$$\mathbf{S} = \frac{1}{n-1}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})'.$$

Observations with large ${CMD_i}^2$ values are often classified as potential outliers. In order to effectively classify an observation as outlier or not, it is helpful to know the (approximate) distribution of the squared distance ${CMD_i}^2$ so that its statistical significance can be judged. It is well known that when $\boldsymbol{X}$ follows a multivariate normal distribution (outlier free), the approximate distribution of the squared Mahalanobis distances (based on the sample mean and covariance) is χ_p^2. Thus, for a given significance level α, any point whose CMD^2 value exceeds $\chi_{p;\alpha}^2$ (the $100(1-\alpha)^{th}$ percentile of the $Chi\text{-}square(p)$ distribution) is declared an outlier. However, this method of detecting outliers is known to be affected by the "masking" and "swamping" when the data contain multiple outliers (see Becker and Gather, 1999). Hadi (1992), which we paraphrase here, explains it in simple terms: Outliers do not always have large Mahalanobis distances. A small cluster of outliers will "attract" $\boldsymbol{T}$ in its direction and "inflate" $\boldsymbol{C}$, yielding a small CMD_i (resulting in masking). Also, $\boldsymbol{T}$ and $\boldsymbol{C}$ are "attracted" into the direction of

the cluster, "away" from some regular observations, which can result in some "good" observations having high classical Mahalanobis distances and hence possibly be declared as outliers (swamping). To alleviate these problems, robust versions of the Mahalanobis distance have been proposed in the literature (Rousseeuw and Zomeren, 1990). For each observation $x_i \in \mathbb{R}^p$ $(1 = 1,2, \dots, n)$ a robust Mahalanobis distance (to a "center" of the data) is given as: $RD_i = \sqrt{(x_i - \boldsymbol{T})^T \boldsymbol{C}^{-1}(x_i - \boldsymbol{T})}$, where $\boldsymbol{T} \in \mathbb{R}^p$ is some robust estimator of location (not the mean), and $\boldsymbol{C}$ is some robust estimator of the covariance matrix (not the sample covariance matrix). Again, to classify an observation as an outlier or not, the distribution (or at least the approximate distribution) of the robust squared distances needs to be known so that a critical value can be found (or at least approximately). Maronna and Zamar (2002) proposed transforming the robust Mahalanobis squared distances ${RD_i}^2$ so that they follow approximately a chi-Square distribution. Their new, scaled distances are given by: $D_i^2 = \chi^2_{p,0.95}(\frac{{RD_i}^2}{median\{{RD_i}^2\}})$ where $median\{{RD_i}^2\}$ is the median of the squared robust distances. Hardin and Rocke (2005) found that the distributions of the robust distances ${RD_i}^2$ could be better described by a scaled F-distribution. Unfortunately, finding the parameters of that F-distribution requires many time-consuming simulations and is not a trivial task. Also, it is desirable that the estimators $\boldsymbol{T}$ and $\boldsymbol{C}$ be chosen to have high breakdown point. Some of the earliest location and scale estimators with high breakdown points are the minimum covariance determinant, MCD, and the minimum volume ellipsoid, MVE, estimators (see Rousseeuw, 1984; Rousseeuw and Leroy, 1987, Rousseeuw and van Zomoren,1990).

Note that the above-mentioned outlier detection methods assume a multivariate normal distribution, which is not easy to justify in practice. The distributional results cited earlier have not been proven to hold when the underlying distribution is not multivariate normal.

2.2.2 Projection Pursuit Method for Outlier Detection

A second group of outlier detection methods is based on the so-called projection pursuit method. This concept was first developed by Friedman and Tukey (1974) (see also Huber, 1985). The idea behind the projection pursuit method is to find suitable projections of the data for which the outliers are more apparent (they "stick out") and hence more detectable. Subsequently, they can be downweighted to produce robust estimators $\boldsymbol{T}$ and $\boldsymbol{C}$. For each projection, each observation is assigned a one-dimensional index (its outlyingness value) reflecting how outlying the observation is in that direction. The oulyingness of an observation $\boldsymbol{x} \in \mathbb{R}^p$ in the direction of a vector $\boldsymbol{u}$ is defined as follows:

$$O^1(\boldsymbol{u}'\boldsymbol{x}, F_{\boldsymbol{u}'\boldsymbol{X}}) = \frac{|\boldsymbol{u}'\boldsymbol{x} - \mu(F_{\boldsymbol{u}'\boldsymbol{X}})|}{\sigma(F_{\boldsymbol{u}'\boldsymbol{X}})}$$

where $F_{\boldsymbol{u}'\boldsymbol{X}}$ is the distribution of the projected data, $\boldsymbol{u}'\boldsymbol{X}$ and $\mu(.)$ and $\sigma(.)$ are some chosen univariate location and scale estimator, respectively, of that distribution. It is assumed that $\mu(.)$ is translation and scale equivariant and that $\sigma(.)$ is translation invariant and scale equivariant (Zuo, 2003). This means that $\mu(F_{\boldsymbol{sY}+\boldsymbol{c}}) = s\mu(F_{\boldsymbol{Y}}) + c$ and $\sigma(F_{\boldsymbol{sY}+\boldsymbol{c}}) = |s|\sigma(F_{\boldsymbol{Y}})$ for any scalars s and c and any univariate random variable Y. This means that location and scale estimators of a linearly transformed random variable also transform accordingly.

The outlyingness of an observation $\boldsymbol{x} \in \mathbb{R}^p$ is its worst (maximal) outlyingness over all directions.

$$O^p(\boldsymbol{x}, F) = \sup_{\|\boldsymbol{u}\|=1} O^1(\boldsymbol{u}'\boldsymbol{x}, F_{\boldsymbol{u}'\boldsymbol{X}}) = \sup_{\|\boldsymbol{u}\|=1} \frac{|\boldsymbol{u}'\boldsymbol{x} - \mu(F_{\boldsymbol{u}'\boldsymbol{X}})|}{\sigma(F_{\boldsymbol{u}'\boldsymbol{X}})}$$

The Stahel-Donoho outlyingness (SDO) is a particular projection pursuit outlyingness obtained by replacing $\mu(.)$ and $\sigma(.)$ by the median (med) and the median absolute deviation

(MAD) respectively. The median absolute deviation (MAD) of a univariate data $X_n = \{x_1, x_2, \dots, x_n\}$ is the median of the absolute deviations of the observations x_i from the median of the data. $MAD(X_n) = med_i(|x_i - med(X_n)|)$.

We have:

$$SDO^p(\boldsymbol{x}, F) = \sup_{\|\boldsymbol{u}\|=1} SDO^1(\boldsymbol{u}'\boldsymbol{x}, F_{\boldsymbol{u}'\boldsymbol{X}}),$$

where

$$SDO^1(\boldsymbol{u}'\boldsymbol{x}, F_{\boldsymbol{u}'\boldsymbol{X}}) = \frac{|\boldsymbol{u}'\boldsymbol{x} - med(F_{\boldsymbol{u}'\boldsymbol{X}})|}{MAD(F_{\boldsymbol{u}'\boldsymbol{X}})},$$

so that

$$SDO^p(\boldsymbol{x}, F) = \sup_{\|\boldsymbol{u}\|=1} \frac{|\boldsymbol{u}'\boldsymbol{x} - med(F_{\boldsymbol{u}'\boldsymbol{X}})|}{MAD(F_{\boldsymbol{u}'\boldsymbol{X}})}.$$

Note that $\frac{|\boldsymbol{u}'\boldsymbol{x} - med(F_{\boldsymbol{u}'\boldsymbol{X}})|}{MAD(F_{\boldsymbol{u}'\boldsymbol{X}})}$ measures the outlyingness of observation $\boldsymbol{x}$ in the direction of vector $\boldsymbol{u}$ and $SDO^p(\boldsymbol{x}, F)$ is the maximal outlyingness value over all directions.

A disadvantage of the projection pursuit method is that theoretically, projections in all directions should be examined in order to calculate the maximum outlyingness, but projecting the data into all directions is practically impossible. In practice, a random number of directions are used instead, to decrease the computational burden. Other schemes exist for choosing the directions in which to project the data. In one version of the projection pursuit, for example, one projects the data onto the n lines formed by the center of the data cloud $\boldsymbol{T}$ (good choices for $\boldsymbol{T}$ are the MVE and the MCD location estimators) and each observation x_i. Along this line of thought, Pena and Prieto (2001) proposed an algorithm called kurtosis 1 that projects the data into 2p directions (The ones that maximize and minimize the kurtosis coefficient of the projected data). Even after such a reduction, the method is still "plagued with very large computational

requirements" (Hadi et. al 2009). For this reason, the SDO is restricted to small datasets and not used much in practice.

2.2.3 Using Data Depth for Outlier Detection

A more general framework for multivariate outlier detection is to associate with each multivariate observation a single (univariate) number (an index) that describes its "position" with respect to the data cloud. For example, one can compute the "robust" Mahalanobis distance of each point from a "center" of the data and set a cut off based on the (approximate) distribution of these distances to indicate a significant outlier. See, for example, Rousseeuw and van Zomeren (1990), Rocke and Woodruff (1996), Filzmozer, Maronna and Werner (2008), Hardin and Rocke (2005) and Cerioli (2010) for various refinements. The robust distances are computed using robust estimators of location and dispersion like the MCD and MVE (Rousseew 1984; Rousseew and Leroy 1985) which are high breakdown estimators that are also affine equivariant. Their high breakdown property allow these estimators to resist up to 50% outliers and their affine equivariant property allow the estimators to change accordingly when the data are subject to affine transformations such as rotations, translations and change of scale. This implies that the robust Mahalanobis distance between two points is not coordinate system dependent.

Instead of using the Mahalanobis robust distance, some researchers have studied the multivariate outlier detection problem using the concept of data depth. The depth of a multivariate observation is given by a univariate (single) number defined by a depth function. The values of depth can then be used to look for extremes among the data. A depth function is a real valued function that provides a "center-outward" ordering of the multivariate data. Several depth functions have been proposed in the literature. Examples are Tukey (1975) halfspace

depth, the simplicial depth (Liu, 1990), the majority depth (Singh,1991; Liu and Singh, 1993), the projection depth (Liu, 1992; Zuo, 2003), The Mahalanobis depth (Liu and Singh, 1993; based on Mahalanobis, 1936), the spatial depth (Serfling, 2002; based on Chaudhuri, 1996), the elliptical depth (Elmore, 2005), the spherical depth (Elmore, Hettmansperger and Xuan. 2006), and recently the triangle depth (Liu and Modarres, 2010). More details are provided later, in chapter 6, about some selected depth functions and their computations. The deeper an observation is with respect to the data cloud, that is, the higher its depth, the closer it is to the center of the data cloud and hence the less outlying it is. Conversely, the lower the depth of an observation, the more outlying it is from the center. Thus, depth and outlyingness are "inversely" related. The higher the outlyingness of an observation, the more outlying the observation is and the lower is its depth. From what precedes, it can be seen that a depth of an observation can be transformed into its outlyingness (and vice versa). Once the depths of the points in a data set are transformed into outlyingnesses, one can find outliers by identifying observations with extreme outlyingness values based on a cut-off.

The next step in the process of outlier detection is the determination of the cutoff value beyond which the outlyingness of an observation has to fall for the observation to be declared as a potential outlier. Dang and Serfling (2010) studied some outlyingness functions and proposed cut off values for the classical Mahalanobis distance outlyingness, halfspace or Tukey outlyingness and the Stahel-Donoho outlyingness, in case of multivariate normal data.

Dang and Serfling (2010) gave a brief example where they illustrated the outlier detection ability of the several outlyingness functions: the classical Mahalanobis distance outlyingness, the robust Mahalanobis distance outlyingness (using MCD estimators), the halfspace outlyingness, the classical Mahalanobis spatial outlyingness, the robust Mahalanobis

spatial outlyingness, and the projection depth. They assumed that the underlying distribution of the data is multivariate normal as do most classical outlier detection techniques. Hubert and Van der Veeken (2008), however, assumed that the underlying distribution is skewed. To detect outliers, they applied their skewness-adjusted boxplot, which was introduced by Hubert and vandervieren (2008) to both the Stahel-Donoho outlyingness values as well as the adjusted outlyingness values (see more details in next section 2.2.4 about the adjusted outlyingness). An alternative approach could be to apply the "modified adjusted boxplot" of Dovoedo and Chakraborti (2010) which is a modification of the adjusted boxplot of Hubert and vandervieren (2008), to the outlyingness values.

The cases being explored are those of bivarite (respectively trivariate) data of size $n = 100$ from the multivariate skew-normal distribution with shape parameter $\alpha = (10,4)^T$ ($\alpha = (10,4,4)^T$) and dependence parameter $\Omega = I_2$ ($\Omega = I_3$) where I_p is the identity matrix of $\mathbb{R}^p$. Our empirical findings suggest that, in the case being studied, the distributions of the outlyingness values resulting from most outlyingness functions mentioned in Dang and Serfling (2010), tend, in general, to be left skewed. This includes the robust Mahalanobis depth outlyingness and the robust Mahalanobis spatial outlyingness.

As mentioned previously, since the outlyingness values are univariate numbers, one could apply a boxplot to them to identify the extreme ones, which will correspond to the (potential) outliers. Our attempt to apply boxplots (the traditional boxplot, the "adjusted boxplot" and the "modified adjusted boxplot") to most of the above-mentioned outlyingness values has not been successful: The upper fences tend to be larger than one, we observe, and no outliers are detected consequently (since outlyingness values are defined to be between zero and 1). As an alternative approach, we can find a percentile (say, 99^{th}, 95^{th} , or even 90^{th} for example) of the

distribution of outlyingness values, for a given outlyingness function, from simulated uncontaminated data, and subsequently use it to detect outliers in contaminated data under various contamination schemes or scenarios. This approach is investigated in this dissertation in a performance simulation study. Notice that while this approach is used on skew-normal data, it can also be used on multivariate normal data. Dang and Serfling (2010) used a similar approach in one of their examples.

2.2.4 Multivariate Skewed Data and Outlier Detection

Most classical outlier detection methods assume that the underlying data follow a multivariate normal distribution. Consequently, their efficacy may be questionable when the data are skewed. A common approach to this problem is to transform some or all of the variables and apply normal theory methods to the transformed data. However, it is not easy to know what transformation(s) to use and which variables to pick that will render the data multivariate normal. Besides, as pointed out by Hubert and van der Veeken (2008), this procedure needs more preprocessing, and leads to variables that are often not interpretable. They proposed an outlier detection method for multivariate skewed data that is inspired by the Stahel-Donoho estimators. The first step of Hubert and van der Veeken (2008) procedure is to adjust the Stahel-Donoho "outlyingness" (SDO) to allow for asymmetry, which leads to the concept of "adjusted outlyingness" (AO).

Recall that for univariate data, the Stahel-Donoho outlyingness for point x_i with respect to the data X_n is defined by:

$$SDO^{(1)}(x_i, X_n) = \frac{|x_i - med(\boldsymbol{X_n})|}{MAD(\boldsymbol{X_n})}$$

where $med(\boldsymbol{X_n})$ is the median of the data $\boldsymbol{X_n} = \{x_1, x_2, \dots, x_n\}$ and $MAD(\boldsymbol{X_n}) = med_i(|x_i - med(\boldsymbol{X_n})|)$ is median absolute deviation of the data. Sometimes $MAD(\boldsymbol{X_n})$ is multiplied by a correction factor to make it unbiased (estimator of the standard deviation σ) at normal samples. The multivariate Stahel-Donoho outlyingness for point x_i with respect to the data X_n was also given.

$$SDO_i = SDO^{(p)}(x_i, \boldsymbol{X_n}) = \sup_{\|u\|=1} SDO^{(1)}(u'x_i, u'\boldsymbol{X_n})$$

For univariate data, Hubert and Van der Veeken (2008) defined the adjusted outlyingness AO of x_i with respect data $\boldsymbol{X_n}$ as:

$$AO_i = AO^{(1)}(x_i, \boldsymbol{X_n}) = \begin{cases} \dfrac{x_i - med(\boldsymbol{X_n})}{w_2 - med(\boldsymbol{X_n})} & \text{if } x_i > med(\boldsymbol{X_n}) \\ \dfrac{med(\boldsymbol{X_n}) - x_i}{med(\boldsymbol{X_n}) - w_1} & \text{if } x_i < med(\boldsymbol{X_n}) \end{cases},$$

where w_1 and w_2 are, respectively, the lower and the upper fences of the adjusted boxplot, given by $[Q_1 - 1.5e^{-4MC}IQR, Q_3 + 1.5e^{3MC}IQR]$ when $MC \geq 0$ and $[Q_1 - 1.5e^{-3MC}IQR, Q_3 + 1.5e^{4MC}IQR]$ when $MC < 0$.

Note that when the distribution of the data is skewed right ($MC > 0$), the scaling factor $(w_2 - med(\boldsymbol{X_n}))$ of $x_i - med(\boldsymbol{X_n})$ for points in the upper tail is larger than the scaling factor $(med(\boldsymbol{X_n}) - w_1)$ for points in the lower tail. That prevents regular observations in the upper tail from being falsely classified as outliers.

The definition of the adjusted outlyingness (AO) of point x_i with respect data $\boldsymbol{X_n}$ in multivariate case is similar to that of the SDO.

$$AO_i = AO^{(p)}(x_i, \boldsymbol{X_n}) = \sup_{\|u\|=1} AO^{(1)}(u'x_i, u'\boldsymbol{X_n})$$

Hubert and Van der Veeken (2008) applied the upper fence of the adjusted boxplot introduced by Hubert and Vandervieren (2008) to the AO and SDO values because the

distributions of the AO's and SDO's are unknown. Hubert and Van der veeken (2008) showed via simulation that their method outperforms the Stahel-Donoho outlyingness (SDO) method at multivariate skew-normal data, because the latter (SDO) does not take into account the skewness of the data. Specifically, the AO method detects higher percentages of outliers and flags less regular observations as outliers.

2.3 Summary

In this dissertation, a set of fences that uses the median and semi-interquartile ranges to define a boxplot procedure is proposed and studied. These can be applied in the case of any continuous and specified (univariate) location-scale distribution, but the focus is on skewed distributions where the traditional boxplot fences do not work well. The probability that at least one observation from a non-contaminated sample is falsely identified as an outlier is controlled to determine the fence constants. Exact expression for this probability, needed to obtain the fence constants, is derived for location-scale distributions. The proposed procedure is compared to a procedure by Sim, Gan and Chang (2005), for a skewed distribution, namely, the exponential distribution and a symmetric distribution, namely, the logistic dissertation. This is presented in chapter 3.

Second, the proposed boxplot procedure lends itself naturally to phase I control charting. The resulting control charts are studied and compared to the charts by Jones and Champ (2002). This is presented in chapter 4.

Third, for univariate skewed distributions, a modification of the adjusted boxplot by Hubert and Vandervieren (2008) is proposed and studied by considering fences constructed from the median and combining the use of the medcouple and the semi-interquartile ranges (SIQRs).

The performance of the resulting boxplot fences is compared with available methods, namely the traditional boxplot and the boxplot procedure by Hubert and Vandervieren (2008) fences. This is presented in chapter 5.

Fourth , for multivariate skewed data, an extensive simulation based comparative study of the outlier detection abilities of some of the available outlyingness functions in the literature is performed. Such a comparison has not yet been reported in the literature. Specifically, we consider robust affine-invariant outlyingness functions derived from the following depth functions: The robust Mahalanobis depth function (Liu and Singh, 1993), the Mahalanobis spatial depth function (Serfling, 2002; based on Chaudhuri, 1996), the robust ellipcal depth (Elmore, 2005), and the robust triangle depth (Liu and Modarres, 2010). Data from the skew-normal distribution of Azzalini and Dalla Valle (1996) are used. This is presented in chapter 6.

Finally, chapter 7 concludes with summary and some topics for future research.

CHAPTER 3

ON A MORE GENERAL BOXPLOT METHOD FOR IDENTLFYING OUTLIERS IN THE LOCATION-SCALE FAMILY

Abstract

Boxplots are among the most widely used exploratory data analysis (EDA) tools in statistical practice. Typical applications of boxplots include eliciting information about the underlying distribution (shape, location, etc.) as well as identifying possible outliers. This paper focuses on a type of lower and upper fences similar in concept to those used in a traditional boxplot. However, instead of constructing the upper and lower fences using the upper and lower quartiles and a multiple of the interquartile range (IQR), multiples of the upper and the lower semi-interquartile ranges (SIQR), respectively, measured from the sample median, are used. Any observation beyond the proposed fences is labelled an outlier. The fence constants (multiples) are obtained by controlling the probability that at least one of the observations in a random sample is wrongly classified as an outlier, the so-called "some-outside rate per sample" (Hoaglin et al. (1986)). An exact expression for this probability is derived for the family of location-scale distributions and is used in the determination of the fence constants. Tables for the constants are provided for practical implementation along with illustrations based on some data; the performance of the proposed rule is explored in a simulation study.

Keywords: Boxplot; Fences; Outlier identification; Interquartile range (IQR); Semi-interquartile range (SIQR); Order Statistics.

1. Introduction

Boxplots are among the most useful tools in Exploratory Data Analysis (EDA). They are typically used to study the shape of the distribution and its characteristics such as the location and the scale. Another important use of these plots is to identify possible outliers, observations that appear to come from a distribution different from that of the remaining data. Hadi et al. (2009) stated, "There are numerous definitions of outliers in the statistical and machine learning literatures." One commonly used definition is that outliers are a minority of observations in a dataset that have different patterns from that of the majority of observations in the dataset. The assumption here is that there is a core of at least 50% of observations in a dataset that are homogeneous (that is, represented by a common pattern) and that the remaining observations (hopefully few) have patterns that are inconsistent with this common pattern. The detection and study of outliers present a significant challenge to the data analyst in many areas of application. Sometimes, the outliers themselves may be of interest as they might lead to new knowledge and discovery. In other cases, the presence of outliers can be a problem as they can significantly distort classical analysis of data and the inferences drawn from that analysis. Thus outlier detection has received and continues to receive considerable attention both inside and outside of the statistics literature (see for example, Barnett (1978), Barnett and Lewis (1994), Cao et al. (2010), Cerioli (2010), Dang and Serfling (2010), Hawkins (2006), Louni (2008), Schwertman et al. (2004), Schwertman and de Silva (2007), Tukey (1977)).

In this paper, we focus on the case of univariate data and consider boxplot-type tools for outlier detection. Although much of the literature on outlier detection has focused on the normal distribution, we consider the broader class of location-scale distributions, given by a cdf $F_{\theta,\sigma}$ where F is a continuous cdf and θ and σ denote, respectively, the location and the scale

parameters. In the context of univariate outlier detection with boxplots, any observation falling either below the lower fence (LF) or above the upper fence (UF) is classified as a potential outlier. The range of these values, $\{x: x \in (-\infty, LF) \cup (UF, \infty)\}$, is called an outlier region. In a traditional boxplot, the LF and the UF are calculated at a distance of k times the interquartile range from the first and the third quartile, respectively. The constant k is called the fence constant, and the value of k is commonly taken to be 1.5 (for the inner lower and upper fences) and 3 (for the outer lower and upper fences). However, these choices of k have been questioned in the literature since they do not take account of the shape of the underlying distribution and/or the sample size. This can be a problem, particularly when the underlying distribution is skewed.

More generally, for a location-scale distribution, Sim et al. (2005; SGC hereafter) defined an outlier region as: $out(\alpha_n, \theta, \sigma^2) = \{x: x \in (-\infty, LF) \cup (UF, \infty)\}$, where α_n denotes the "error rate per observation," which is the probability that an observation in a sample of size n is incorrectly identified as an outlier; LF and UF are some suitably defined fences. Thus $\alpha_n =$ $\Pr[X \epsilon out(\alpha_n, \theta, \sigma^2) | X \text{ is not an outlier}]$. Note that this probability and hence the outlier region depend on the underlying cdf $F_{\theta,\sigma}$. Several authors have considered finding the fence constant k for a small nominal value of α_n.

Another error rate of interest in this context is the probability α, that one or more observations in the entire sample are wrongly (declared) classified as outliers. Thus

$$\alpha = \Pr[\text{one or more of the } X_1, X_2, \dots, X_n \in out(\alpha_n, \theta, \sigma^2) | \text{they are not outliers}]$$

Hoaglin et al. (1986) termed α_n "the outside rate per observation" and α "the some-outside rate per sample." The α criterion, also used in SGC, seems to be more meaningful since we are often interested in knowing if a dataset is contaminated (with outliers) or not, meaning that we want to

find if there is at least one outlier, and an entire sample of observations is generally examined together.

In order to account for possible asymmetry in the underlying distribution, SGC defined their fences as:

$$LF_1 = X_{(l)} - k_l^*\left(X_{(u)} - X_{(l)}\right);\ UF_1 = X_{(u)} + k_u^*\left(X_{(u)} - X_{(l)}\right),$$

where $X_{(l)}$ and $X_{(u)}$ are the lower and the upper fourths (quartiles), respectively, of a sample of size n, k_l^* and k_u^* are two fence constants determined so that the probability α is controlled at some given small value. They show that using a common value of $k_l^* = k_u^* = k = 3$ or 1.5 (as in the traditional boxplot) is not appropriate for normal data and may be even more problematic when the data follow an exponential distribution as that may lead to many false alarms. In fact, their values for the fence constants, for a given value of α, n and a distribution (the normal and exponential) are different from 3 and 1.5 in most cases.

To incorporate the skewness of the distribution and possibly improve outlier detection, Kimber (1990) proposed fences using a constant multiple of the lower and the upper semi-interquartile ranges (defined later) measured from the first and third quartile, respectively. Carling (2000) suggested that it may be advantageous to construct the fences from the median using a constant multiple of the interquartile range. Schwertman et al. (2004; hereafter SOA) also considered fences constructed from the median but pointed out that Carling's fences "can lead to overlooking outliers in the short tail and conversely identifying many non-outliers in the long tail". Consequently, they proposed a method of constructing fences for the normal and "near normal but somewhat asymmetric" distributions. For example, for near normal but somewhat asymmetric distributions, their fences are defined as:

$$LF_2 = X_{(m)} - \frac{2Z_{\frac{\alpha_n}{2}}}{k_n} SIQR_L \quad \text{and} \quad UF_2 = X_{(m)} + \frac{2Z_{\frac{\alpha_n}{2}}}{k_n} SIQR_U$$

where $SIQR_L = X_{(m)} - X_{(l)}$ and $SIQR_U = X_{(u)} - X_{(m)}$ are the lower and the upper "semi-interquartile ranges", respectively, $X_{(m)}$ is the sample median, k_n is a fence constant obtained from SOA Table 3.1 and $Z_{\alpha_n/2}$ is the $100(1-\alpha_n/2)^{th}$ standard normal percentile.

In this paper, we consider fences using the same order statistics, $X_{(l)}, X_{(m)}$ and $X_{(u)}$ as those in SOA but our fence constants are different as we control the SORS probability α (and not α_n) Moreover, we use an exact derivation for an expression for the probability α. Thus, our fences are defined by:

$$LF = X_{(m)} - k_l SIQR_L;\ UF = X_{(m)} + k_u SIQR_U, \tag{3.1}$$

where the fence constants k_l and k_u depend on α, n and F, the cdf of standardized variable $(X-\theta)/\sigma$. More details are provided later.

For illustration, the traditional boxplot with $k = 1.5$, its fences and the proposed fences in (1) are shown in Figure 3.1. Note that the boxes in both boxplots are the same, only the fences are different and that is what makes the outlier detection capabilities different for these procedures.

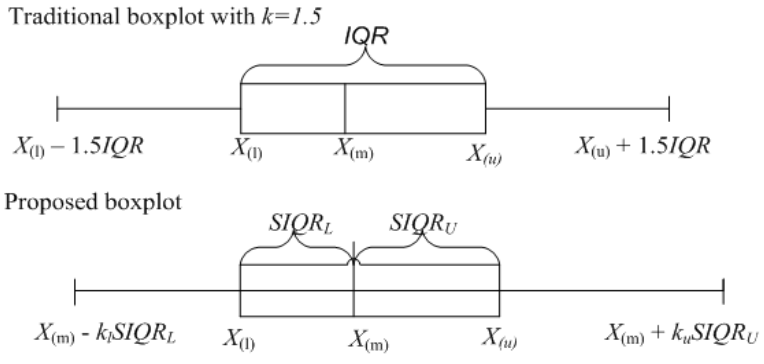


Figure 3.1: The traditional boxplot and the proposed boxplot

SOA's theoretical developments assume a normal or a near normal underlying distribution and use a large sample approximation for the expected values of normal order statistics. We assume, more generally, that the underlying distribution is in the location-scale family given by a cdf $F_{\theta,\sigma}(x)$. Thus, our results apply to a larger class of distributions that includes, for example, the normal and the exponential distribution. We find the fence constants k_l and k_u in Equation (3.1) so that the probability α is controlled at some small nominal value.

2. Construction of the fences

We derive an exact expression for the some-outside rate per sample (α) using the fences in (1) and use it to find the fence constants for a given small value of α, a sample size *n* and a cdf F. To this end, we show (see Appendix) that the "some-outside rate per sample" is given by:

$$\alpha = P[one\ or\ more\ of\ the\ X_1, X_2, \dots, X_n \epsilon\ out(\alpha_n, \theta, \sigma^2) | they\ are\ not\ outliers]$$

$$= \int_{-\infty}^{\infty}\int_{z_{(m)}}^{\infty} \left[1 - I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(m)},Z_{(u)}}(z_{(m)}, z_{(u)}) dz_{(u)} dz_{(m)} +$$

$$\int_{-\infty}^{\infty}\int_{z_{(l)}}^{\infty}\int_{z_{(m)}}^{\infty} \left[I_{G_l(y_l)}(\alpha = 1, \beta = l-1) I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(l)},Z_{(m)},Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})\, dz_{(u)} dz_{(m)} dz_{(l)} \quad (3.2)$$

Where $y_u = z_{(m)} + k_u(z_{(u)} - z_{(m)})$, $I_{G_u(y_u)}(\alpha = n-u, \beta = 1)$ is an incomplete Beta function evaluated at $G_u(y_u) = \frac{F(y_u) - F(z_{(u)})}{1 - F(z_{(u)})}$, $y_l = z_{(m)} - k_l(z_{(m)} - z_{(l)})$, $I_{G_l(y_l)}(\alpha = 1, \beta = l-1)$, is an incomplete Beta function evaluated at $G_l(y_l) = \frac{F(y_l)}{F(z_{(l)})}$ and $z_{(i)}$ is the value of $Z_{(i)}$, the i^{th} order statistic of a random sample of size *n* from the distribution of the standardized random variable $Z = (X - \theta)/\sigma$, where X has the cdf $F_{\theta,\sigma}(x)$. .

The fence constants k_l and k_u are found by solving Equation (3.2) for a given (nominal) small error rate α. This is done using a direct search method using both Mathcad and Mathematica. Note that the first term on the right hand side of Equation (3.2) depends on k_u but not on k_l, while the second term depends on both k_u and k_l. Hence we first find k_u to make the first term equal to $\alpha/2$ and then find k_l that makes the second term also equal to $\alpha/2$. Note also that for a symmetric distribution, we use $k_l = k_u = k$ and search for *k* that gives the desired value α. For ease of implementation, in Table 3.1 we provide the values of our fence constants for the normal, the logistic and the exponential distributions, for $\alpha = 0.05$ and 0.1. The logistic distribution is normal-like but with slightly heavier tails, whereas the exponential distribution is skewed to the right. Equation (3.2) can be used to determine the fence constants for any specified location scale distribution.

In some situations, particularly while dealing with skewed distributions (such as the exponential), a one-sided fence (say on the upper side) may be of more interest as higher values may be more rare or problematic. For the fence constant in the case of a one tail upper boxplot, we find the value of k_u that makes the first term in Equation (3.2) equal to α. Table 3.1 also gives values of the upper fence constant for the exponential distributions for $\alpha = 0.05$ and 0.1.

Table 3.1: Fence constants for selected sample sizes from the Normal, Logistic, and Exponential populations

Sample Size n	$mod(n,4)$	Normal		Logistic		Exponential				Exponential*	
		$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 0.05$	$\alpha = 0.1$
		k	k	k	k	k_l	k_u	k_l	k_u	k_u	k_u
12	0	6.744	5.245	7.980	6.135	4.140	9.350	3.236	7.290	7.290	5.550
16		6.515	5.292	7.935	6.360	3.600	9.970	2.966	8.000	8.000	6.275
20		6.345	5.295	7.890	6.485	3.265	10.316	2.787	8.442	8.445	6.756
24		6.225	5.291	7.863	6.575	3.050	10.540	2.660	8.750	8.750	7.110
28		6.135	5.286	7.854	6.648	2.900	10.710	2.566	8.985	8.985	7.387
40		5.973	5.279	7.870	6.819	2.623	11.035	2.388	9.467	9.467	7.968
52		5.890	5.282	7.917	6.953	2.472	11.259	2.284	9.787	9.787	8.357
72		5.822	5.299	8.015	7.131	2.323	11.535	2.179	10.163	10.163	8.811
88		5.799	5.317	8.095	7.249	2.249	11.710	2.126	10.394	10.394	9.083
100		5.790	5.331	8.153	7.327	2.190	11.830	2.096	10.540	10.540	9.254
120		5.785	5.354	8.243	7.443	2.154	12.001	2.057	10.749	10.749	9.494
152		5.789	5.391	8.373	7.600	2.094	12.235	2.012	11.025	11.025	9.804
13	1	10.550	8.050	12.565	9.510	5.186	19.000	4.040	14.150	14.150	10.285
17		8.885	7.137	10.865	8.627	4.200	16.390	3.438	12.800	12.800	9.763
21		8.040	6.658	10.025	8.190	3.650	15.090	3.108	12.130	12.130	9.530
25		7.533	6.367	9.536	7.943	3.350	14.320	2.903	11.740	11.740	9.415
29		7.195	6.174	9.230	7.789	3.130	13.830	2.756	11.504	11.505	9.364
45		6.534	5.797	8.685	7.557	2.679	12.960	2.449	11.132	11.132	9.381
61		6.266	5.653	8.525	7.528	2.476	12.689	2.302	11.076	11.076	9.502
89		6.065	5.558	8.475	7.587	2.294	12.587	2.168	11.158	11.158	9.740
101		6.020	5.541	8.485	7.625	2.230	12.599	2.132	11.212	11.212	9.836
149		5.939	5.526	8.582	7.786	2.123	12.724	2.039	11.455	11.455	10.174
125		5.967	5.526	8.529	7.706	2.170	12.649	2.075	11.333	11.333	10.014
10	2	11.810	8.300	13.670	9.540	8.244	13.354	5.735	9.940	9.940	7.217
14		9.095	7.026	10.910	8.345	5.286	12.825	4.137	10.010	10.010	7.627
18		8.010	6.483	9.840	7.869	4.247	12.503	3.499	10.043	10.043	7.884
22		7.430	6.185	9.295	7.633	3.716	12.303	3.154	10.080	10.080	8.075
26		7.067	5.999	8.974	7.502	3.385	12.176	2.936	10.120	10.120	8.230
30		6.820	5.873	8.772	7.425	3.160	12.096	2.785	10.160	10.160	8.360
42		6.414	5.666	8.475	7.345	2.774	11.995	2.520	10.297	10.297	8.671
46		6.331	5.626	8.428	7.342	2.696	11.988	2.464	10.342	10.342	8.757
50		6.266	5.595	8.395	7.345	2.631	11.988	2.417	10.387	10.385	8.837
62		6.129	5.534	8.347	7.375	2.487	12.017	2.312	10.516	10.516	9.047
86		5.991	5.484	8.350	7.468	2.320	12.129	2.188	10.749	10.749	9.379
102		5.950	5.474	8.383	7.535	2.251	12.215	2.137	10.885	10.885	9.560
106		5.937	5.473	8.392	7.552	2.237	12.237	2.126	10.920	10.920	9.602
11	3	7.726	5.930	9.092	6.908	4.009	13.354	3.159	9.937	9.937	7.217
15		7.175	5.784	8.707	6.925	3.515	12.825	2.913	10.010	10.010	7.628
19		6.840	5.675	8.470	6.930	3.216	12.503	2.747	10.045	10.045	7.886
23		6.614	5.599	8.330	6.940	3.015	12.304	2.629	10.080	10.080	8.076
27		6.456	5.544	8.240	6.958	2.870	12.177	2.541	10.119	10.119	8.230
31		6.338	5.504	8.185	6.981	2.759	12.095	2.473	10.162	10.162	8.360
55		6.010	5.405	8.115	7.144	2.423	11.994	2.248	10.431	10.431	8.911
75		5.913	5.390	8.167	7.275	2.294	12.068	2.157	10.637	10.637	9.225
103		5.857	5.398	8.266	7.435	2.190	12.215	2.082	10.886	10.886	9.560
151		5.835	5.432	8.437	7.658	2.090	12.472	2.009	11.233	11.233	9.985

* Upper Tail

As Banerjee and Iglewicz (2007) point out, there are situations in current practice, such as in data mining applications where outlier detection is important but where n is much larger than what used to be considered large, say 100, in the past. When n is greater than 150, the analytical solutions obtained using Mathcad or Mathematica become unstable and therefore not reliable. However, we can follow the approach of Banerjee and Iglewicz (2007) to calculate the approximate fence constants for our procedure for large sample sizes.

For example, for symmetric distributions, it can be shown that k is given by

$$k = \frac{F^{-1}\left(\left(1-\frac{\alpha}{2}\right)^{\frac{1}{n}}\right) - F^{-1}(0.5)}{F^{-1}(0.75) - F^{-1}(0.5)}.$$

When the distribution is asymmetric and the focus is on the upper tail and it can be shown that this leads to:

$$k_u = \frac{F^{-1}\left((1-\alpha)^{\frac{1}{n}}\right) - F^{-1}(0.5)}{F^{-1}(0.75) - F^{-1}(0.5)}$$

Table 3.2 shows the fence constants for the normal, logistic and exponential distributions, for a number of selected large sample sizes.

Table 3.2: Approximate fence constants for selected large sample sizes from the Normal, Logistic, and Exponential populations

	Normal		Logistic		Exponential*	
Sample Size n	$\alpha = 0.05$ k	$\alpha = 0.1$ k	$\alpha = 0.05$ k	$\alpha = 0.1$ k	$\alpha = 0.05$ k_u	$\alpha = 0.1$ k_u
1000	6.009	5.759	9.634	8.991	13.251	12.212
2000	6.245	6.004	10.265	9.622	14.251	13.212
5000	6.545	6.315	11.099	10.456	15.573	14.534
10000	6.764	6.541	11.73	11.087	16.573	15.534
50000	7.249	7.04	13.195	12.552	18.895	17.856
100000	7.448	7.245	13.826	13.183	19.895	18.856
500000	7.894	7.702	15.291	14.648	22.217	21.178
1000000	8.079	7.891	15.922	15.279	23.217	22.178

* Upper Tail

Our empirical checks suggest that, in general, a sample size of 2000 is sufficient to use the preceding large sample fence constant formulae. In case of the sample size between 150 and 2000, it is suggested, as in Banerjee and Iglewicz (2007), that the fence constant k be multiplied by an adjusting constant g, so that the final fence constant is kg. From extensive simulations, we found the following approximate formulae for obtaining the adjusting constant g, where $w = 1/n$.

In case of normal data:

For $\alpha = 0.05$, $g = 0.99639 + 25.01803w - 4739.49w^2 + 1119830w^3 - 1.00294\text{x}10^8 w^4$

For $\alpha = 0.1$, $g = 0.99789 + 16.81195w + 526.80509w^2 - 535302w^3 + 5.11715\text{x}10^7 w^4$

In case of logistic data:

For $\alpha = 0.05$, $g = 1.00013 + 12.01653w - 1645.59w^2 + 394182w^3 - 3.04731\text{x}10^7 w^4$

For $\alpha = 0.1$, $g = 0.99802 + 9.69903w - 102.47275w^2 - 29586.8w^3$

In case of exponential data:

For $\alpha = 0.05$, $g = 0.99826 + 16.95856w - 2171.49w^2 + 157335w^3$

For $\alpha = 0.1$, $g = 0.99983 + 13.35028w - 1006.87w^2$.

3. Illustration

We illustrate the methodology with some data and highlight the differences between the traditional boxplot, our boxplot, the SGC boxplot, and the SOA boxplot.

3.1. Example 1:

Consider the following data reported in Daniel (1959) in a 2^5 factorial experiment (with one missing observation). The data is given in Barnett (1978) also. For convenience, the data are presented here in an increasing order:

Table 3.3: Data from Daniel (1959)

-3.143	-2.666	-1.305	-0.898	-0.8138	-0.8138	-0.7577
-0.7437	-0.4771	-0.3087	-0.2526	-0.0982	-0.0842	-0.0561
0	0.0281	0.1263	0.1684	0.1964	0.2245	0.2947
0.3929	0.4069	0.4209	0.435	0.463	0.5472	0.6595
0.7437	1.08	2.147				

Note that the data with all $n = 31$ observations (contrasts) fail the Anderson Darling normality test. From Minitab, the Anderson Darling statistic is 1.205, with a P-value less than 0.05. However, the data without the first two and the last observations (-3.143, -2.666 and 2.147, respectively) do not. Specifically, for the reduced data, the Anderson Darling statistic is $AD = 0.464$ with a P-value = 0.236. These three observations seem to be inconsistent with the remaining data (based on the histogram not shown here).

We use the α_n criterion to find the fence constants, so that our boxplot and that of SGC can be meaningfully compared with the boxplots of SOA. Note that while there are many ways of defining the sample fourths (sample quartiles) available in the literature, we use $l = n/4$ when $mod(n, 4) = 0$ and $l = int\left(\frac{n}{4}\right) + 1$, otherwise. The quantity u is chosen to be $u = n - l + 1$. Also, we use $m = n/2$, when $mod(n, 2) = 0$ and $m = int\left(\frac{n}{2}\right) + 1$, otherwise, while defining the sample median. Using these definitions, for these data, we find: $q_1 = X_{(8)} = -0.7437$, $q_2 = X_{(16)} = 0.0281$, and $q_3 = X_{(24)} = 0.4209$. In Table 3.4, we show the fences obtained by the various methods.

Table 3.4: Fences for various boxplot procedures for the data in Daniel (1959)

Traditional Boxplot $k = 1.5$ $k = 3$	$LF = q_1 - 1.5(q_3 - q_1) = -2.4906$; $UF = q_3 + 1.5(q_3 - q_1) = 2.1678$ $LF = q_1 - 3(q_3 - q_1) = -4.2375$; $UF = q_3 + 3(q_3 - q_1) = 3.9147$
SOA's Boxplot* $\alpha_n = 0.05$ $\alpha_n = 0.1$	$LF = q_2 - 1.411(q_3 - q_1) = -1.6152$; $UF = q_2 + 1.411(q_3 - q_1) = 1.6714$ $LF = q_2 - 1.185(q_3 - q_1) = -1.352$; $UF = q_2 + 1.185(q_3 - q_1) = 1.4082$
SOA's Boxplot ** $\alpha_n = 0.05$ $\alpha_n = 0.1$	$LF = q_2 - 2.823(q_2 - q_1) = -2.1507$; $UF = q_2 + 2.823(q_3 - q_2) = 1.137$ $LF = q_2 - 2.369(q_2 - q_1) = -1.8003$; $UF = q_2 + 2.369(q_3 - q_2) = 0.9586$
SGC's Boxplot*** $\alpha_n = 0.05$ $\alpha_n = 0.1$	$LF = q_1 - 0.85(q_3 - q_1) = -1.7336$; $UF = q_3 + 0.85(q_3 - q_1) = 1.4108$ $LF = q_1 - 0.592(q_3 - q_1) = -1.4133$; $UF = q_3 + 0.592(q_3 - q_1) = 1.1103$
Our Boxplot*** $\alpha_n = 0.05$ $\alpha_n = 0.1$	$LF = q_2 - 2.83(q_2 - q_1) = -2.1561$; $UF = q_2 + 2.83(q_3 - q_2) = 1.1397$ $LF = q_2 - 2.248(q_2 - q_1) = -1.7069$; $UF = q_2 + 2.248(q_3 - q_2) = 0.9111$

*Assuming a normal distribution; **Assuming a near normal distribution
*** Note that $\alpha = 1 - (1 - \alpha_n)^n$ is used to find the fence constants for a given α_n.

Note that SOA's fence constants are computed using the formula $\frac{2Z_{\alpha_n/2}}{k_n}$ and $\frac{Z_{\alpha_n/2}}{k_n}$ for "near normal" and normal distribution, respectively, where k_n is found from their Table 1. For $n = 31$, SOA's Table 1 gives $k_n = 1.38876$. To find our fence constants as well as those of SGC's, we first find the value of α that corresponds to α_n using the relationship, $\alpha = 1 - (1 - \alpha_n)^n$. Using the resulting value of α, we set up Equation (3.2) in Mathcad and find our fence constant, as for Table 3.1. The same approach is used to find SGC's fence constants, except that we use their expression for the some-outside rate per sample (their Equation (7)) directly.

Next, using the fences given in Table 3.4, we find the number of outliers detected and the associated number of false alarms for each procedure. Recall that the false alarm corresponds to

the event that a particular observation is identified as an outlier when in fact it is not. The results are summarized in Table 3.5 below.

Table 3.5: Performance of various procedures in detecting outliers based on Daniel (1959)'s data

	*Actual number of outliers in the data = 3									
	Traditional Boxplot		SOA's Boxplot*		SOA's Boxplot**		SGC's Boxplot		Our Boxplot	
			α_n							
	$k = 1.5$	$k = 3$	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.1
Number of outliers detected	2	0	3	3	3	4	3	3	3	4
Number of false alarms	0	0	0	0	0	1	0	0	0	1

*Assuming data follow a normal distribution **Assuming data follow a near normal distribution

We see from Table 3.5 that the traditional boxplot with $k = 3$ does not detect any of the three outliers present in the data, although the one with $k = 1.5$ does detect two and there are no false alarms. Both SOA's and SGC's procedures detect all three outliers with no false alarm, except at $\alpha_n = 0.1$, where four outliers are indicated by SOA's boxplot assuming a near normal distribution, leading to one false alarm. The performance of our method is the same as that of SOA's boxplot for a near normal distribution. However, note that our procedure can be applied to any location-scale distribution, while SOA's method is recommended for normal and near normal distributions only.

3.1. Example 2

As a second illustration, consider the example in Montgomery (2009; pp. 325) in which a chemical engineer wishes to control the average time between failures of a valve. She observed twenty times between failures for this valve. The data are shown in Table 3.6.

Table 3.6: Times between failures data

286	948	536	124	816	729	4	143	431	8
2837	596	81	227	603	492	1199	1214	2831	96

First, note that the data with all twenty observations do not fail the Anderson Darling exponential test. From Minitab, the Anderson Darling statistic is 0.53 with a P-value = 0.44. The quartiles for these data are given by: $q_1 = X_{(5)} = 124$, $q_2 = X_{(10)} = 492$, and $q_3 = X_{(16)} = 948$. We use the same quartiles in the calculations for the traditional and SGC's boxplots. However, SOA's boxplots are not calculated for this example because they require the assumption of a normal or a near normal distribution. Also, in this example we use the α criterion that was originally used to derive the fences for our and SGC's boxplots.

Assume we want to detect process improvement. Because the times between failures (non-negative random variable) are monitored, higher values are of interest. One-tailed boxplots are constructed here with only an upper fence and the lower fence is set equal to zero. One could argue that we should also set the lower fence for the traditional boxplot to zero. Doing so does not change the final conclusions, however. For our one-tailed procedure, the value of the upper fence constant k_u is obtained from Table 3.1 for $\alpha = 0.05$ and $\alpha = 0.1$. The method for obtaining the fence constants for the SGC method is similar to ours; we use only the first term of SGC's Equation (7). The results are given in Table 3.7.

Table 3.7: Fences for various boxplot procedures for the times between failures data

Traditional Boxplot	
$k = 1.5$	$LF = q_1 - 1.5(q_3 - q_1) = -1112$; $UF = q_3 + 1.5(q_3 - q_1) = 2184$
$k = 3$	$LF = q_1 - 3(q_3 - q_1) = -2348$; $UF = q_3 + 3(q_3 - q_1) = 3420$
SGC Boxplot*	
$\alpha = 0.05$	$LF = 0$; $UF = q_3 + 4.365(q_3 - q_1) = 4544.76$
$\alpha = 0.1$	$LF = 0$; $UF = q_3 + 3.495(q_3 - q_1) = 3827.88$
Our Boxplot*	
$\alpha = 0.05$	$LF = 0$; $UF = q_2 + 8.445(q_3 - q_2) = 4342.92$
$\alpha = 0.1$	$LF = 0$; $UF = q_2 + 6.756(q_3 - q_2) = 3572.736$

*One tailed upper fence.

Table 3.8 summarizes the findings regarding the outlier detection performance of these procedures for the times to failure data:

Table 3.8: Performance of various procedures in detecting outliers based on Times to failure data

	*Actual number of outliers in the data = 0					
	Traditional Boxplot		SGC's Boxplot		Our boxplot	
			α			
	$k = 1.5$	$k = 3$	0.05	0.1	0.05	0.1
Number of outliers detected	2	0	0	0	0	0
Number of false alarms	2	0	0	0	0	0

*Assuming Exponential distribution

Note that for these data, for the traditional boxplots, the "inner" fence (with $k = 1.5$) falsely identifies the two observations, with respective values 2837 and 2831, as outliers. The "outer" fences (with $k = 3$) however do not flag any observation as outlier, which is in accordance with the previous exponential goodness-of-fit test. This illustrates that the use of the traditional boxplot (with $k = 1.5$) may be problematic, particularly with skewed-exponential data. Both our and SGC's procedures do not flag any observation as outlier at both $\alpha = 0.05$ and 0.10, which is in accordance with the previous exponential goodness-of-fit test.

4. Performance Study

Given there are several methods for detecting outliers, there is naturally the question about which of the methods perform better, when, and by how much. The efficacy of an outlier detection procedure can be studied by studying the probability of detecting r $(= 1,2,3, \dots, n_0, \dots)$

outliers in a given sample of size n that has n_0 outliers. Because the desired probabilities are not easy to obtain analytically, we use simulations to estimate them. Our method is compared with that of SGC's as they use the α criterion while determining the fence constants. Recall that SOA's method uses the α_n criterion. We adapt the algorithm by SGC. The steps are as follows.

Step 1: Generate n observations $X_1, X_2, \dots, X_n$ iid from the hypothesized (location-scale) distribution $F_{\theta,\sigma}$ with specified values for θ and σ. Identify the standardized cdf F. For example if $F_{\theta,\sigma}$ is $N(2,3)$, $\theta = 2$ and $\sigma = 3$, and F is the standard normal cdf.

Step 2: Find the three quartiles $X_{(l)}, X_{(m)}, X_{(u)}$ of $F_{\theta,\sigma}$ and calculate our fences $LF = X_{(m)} - k_l\left(X_{(m)} - X_{(l)}\right)$ and $UF = X_{(m)} + k_u\left(X_{(u)} - X_{(m)}\right)$, where k_l and k_u are obtained from Table 3.1 according to the standardized hypothesized cdf F, the sample of size n and the error rate α. The fences for SGC's method are calculated using the formula $LF_1 = X_{(l)} - k_l^*\left(X_{(u)} - X_{(l)}\right)$ and $UF_1 = X_{(u)} + k_u^*\left(X_{(u)} - X_{(l)}\right)$, where k_l^* and k_u^* are obtained from their Table 1.

Step 3: Generate an observation y from any arbitrarily chosen distribution with cdf *H* (independently of the $X's$).

For example, we can take H to be the cdf of $Expo(1/5)$, that is the exponential distribution with mean 5. Large deviation of the mean of H from that of $F_{\theta,\sigma}$ and/or higher standard deviation of H compared with that of $F_{\theta,\sigma}$ will make the contamination of the data and the detection of outliers easier, since if the generated outliers are "too close" to regular observations, they are not "real outliers" and cannot be detected. In our simulation study, we use, successively, $H = Expo(1/5)$, $H = Expo(1/6)$ and $H = N(1,6)$ as the distribution from which outliers are generated.

Step 4: If ($y < LF$ or $y > UF$) and ($y < LF_1$ or $y > UF_1$) then y is labeled as an outlier. Step 3 is repeated until n_0 outliers are generated. These n_0 observations replace n_0 of the randomly selected observations of the n observations generated in step 1.

Step 5: Once the sample obtained in step 1 is contaminated with n_0 outliers in step 4, we re-compute the fences and count how many outliers are detected, i.e. how many of the observations fall above the upper fence or below the lower fence.

The whole process (steps 1 through 5) is repeated 100,000 times and finally we calculate the proportion of times $0,1,\ldots,n_0,n_0+1,\ldots$ outliers are detected. The 95% margin of error for results reported in Table 3.9 and Table 3.10 is at most 0.0031. Two-tailed boxplots are used throughout, as the practitioner most likely would not know in what direction the outliers would be.

While our focus when developing the proposed procedure has been on skewed distributions, our fence constants have been derived for the entire location-scale family of distributions and as such we include, in the performance comparison, some popular symmetric members of the family, namely the normal and the logistic distributions. The results of the simulation study when the underlying distribution is exponential and logistic are shown in Tables 3.9 and 3.10, respectively. The results for the normal distribution are similar to those for the logistic and are not reported here. The proportions of outliers detected are shown according to the number of outliers used to contaminate the initial sample. For example, when the initial sample is contaminated with two outliers, the probabilities of detecting (i) at most one outlier (ii) exactly two outliers and (iii) three or more outliers are estimated by the empirical proportions,

and can be found in the columns under "$n_0 = 2$" in Table 3.9 and Table 3.10 corresponding to each of these categories.

Table 3.9: Empirical proportions of outliers detected with respect to the hypothesized $Expo(1)$ distribution based on 100,000 simulated samples of size n with n_0 outliers at $\alpha = 0.05$

Sample size *n*	Method	Actual number of Outliers present									
		$n_0 = 1$			$n_0 = 2$			$n_0 = 4$			
		Number of Outliers detected									
		0	1	>=2	<=1	2	>=3	<=2	3	4	>=5
Distribution of outliers Expo(1/5)											
50	New Procedure	0.097	0.863	0.040	0.248	0.723	0.028	0.460	0.264	0.261	0.015
	SGC Procedure	0.088	0.874	0.038	0.258	0.716	0.026	0.370	0.369	0.249	0.012
75	New Procedure	0.078	0.881	0.041	0.211	0.755	0.033	0.228	0.340	0.411	0.021
	SGC Procedure	0.065	0.895	0.040	0.162	0.806	0.032	0.195	0.330	0.457	0.017
150	New Procedure	0.035	0.920	0.045	0.095	0.865	0.040	0.124	0.253	0.594	0.029
	SGC Procedure	0.024	0.932	0.044	0.115	0.848	0.037	0.093	0.324	0.558	0.025
Distribution of outliers Expo(1/6)											
50	New Procedure	0.058	0.901	0.041	0.266	0.703	0.031	0.321	0.308	0.354	0.017
	SGC Procedure	0.084	0.877	0.039	0.287	0.685	0.028	0.332	0.305	0.349	0.014
75	New Procedure	0.078	0.880	0.042	0.178	0.787	0.034	0.197	0.340	0.441	0.022
	SGC Procedure	0.038	0.922	0.040	0.145	0.823	0.032	0.183	0.376	0.423	0.017
150	New Procedure	0.019	0.936	0.046	0.133	0.827	0.040	0.125	0.269	0.576	0.030
	SGC Procedure	0.028	0.928	0.044	0.079	0.883	0.038	0.080	0.235	0.658	0.027
Distribution of outliers Norm(1,6)											
50	New Procedure	0.076	0.883	0.041	0.242	0.727	0.032	0.197	0.355	0.430	0.019
	SGC Procedure	0.054	0.907	0.039	0.186	0.787	0.028	0.211	0.384	0.392	0.013
75	New Procedure	0.011	0.946	0.043	0.039	0.925	0.036	0.012	0.084	0.876	0.028
	SGC Procedure	0.016	0.942	0.042	0.051	0.914	0.035	0.017	0.098	0.860	0.025
150	New Procedure	0.015	0.940	0.045	0.026	0.934	0.040	0.008	0.071	0.889	0.032
	SGC Procedure	0.009	0.946	0.044	0.022	0.938	0.039	0.004	0.054	0.910	0.032

From an examination of the numbers in Table 3.9, it is clear that neither procedure is uniformly best when the underlying distribution is exponential. When the degree of contamination is low (one or two outliers), there is no clear winner between the two procedures; the performance of our procedure is similar to that of SGC's. However, when the degree of contamination is high (four outliers present), our procedure appears to be slightly more powerful.

For example, for a sample of size $n = 50$ from the $Expo(1)$ distribution in which four observations are replaced by outliers from the $Expo(1/5)$ distribution, the proposed procedure detects exactly four outliers about 26.1% of the time while SGC procedure detects the two outliers about 24.9% of the time. For a sample of size $n = 75$ from the same distribution for which four observations are replaced by outliers from the $Expo(1/6)$ distribution, our procedure detects exactly four outliers about 44.1% of the time while SGC procedure detects the same about 42.3% of the time.

Table 3.10: Empirical proportions of outliers detected with respect to the hypothesized $logis(0,1)$ distribution based on 100,000 simulated samples of size n with n_0 outliers at $\alpha = 0.05$

Sample size *n*	Method	Actual number of Outliers present									
		$n_0 = 1$			$n_0 = 2$			$n_0 = 4$			
		Number of Outliers detected									
		0	1	>=2	<=1	2	>=3	<=2	3	4	>=5
Distribution of outliers Logis(3,1)											
50	New Procedure	0.268	0.595	0.137	0.533	0.383	0.084	0.629	0.130	0.207	0.034
	SGC Procedure	0.148	0.656	0.196	0.465	0.406	0.129	0.432	0.264	0.244	0.060
75	New Procedure	0.151	0.628	0.222	0.401	0.455	0.144	0.642	0.096	0.204	0.058
	SGC Procedure	0.081	0.632	0.288	0.348	0.470	0.182	0.595	0.127	0.196	0.083
150	New Procedure	0.054	0.546	0.400	0.249	0.452	0.298	0.331	0.153	0.329	0.187
	SGC Procedure	0.052	0.499	0.448	0.179	0.457	0.363	0.333	0.179	0.279	0.209
Distribution of outliers Logis(2,2)											
50	New Procedure	0.167	0.677	0.157	0.172	0.692	0.136	0.431	0.275	0.253	0.040
	SGC Procedure	0.070	0.713	0.217	0.057	0.743	0.200	0.417	0.259	0.262	0.062
75	New Procedure	0.088	0.676	0.235	0.257	0.558	0.184	0.337	0.230	0.341	0.092
	SGC Procedure	0.085	0.621	0.294	0.137	0.600	0.263	0.145	0.234	0.471	0.150
150	New Procedure	0.053	0.549	0.398	0.203	0.494	0.303	0.107	0.199	0.444	0.250
	SGC Procedure	0.055	0.497	0.449	0.176	0.474	0.350	0.107	0.138	0.449	0.306
Distribution of outliers Logis(4,1)											
50	New Procedure	0.173	0.670	0.156	0.475	0.429	0.096	0.620	0.155	0.194	0.031
	SGC Procedure	0.081	0.704	0.215	0.414	0.440	0.146	0.468	0.294	0.191	0.047
75	New Procedure	0.142	0.633	0.225	0.172	0.630	0.197	0.540	0.176	0.222	0.062
	SGC Procedure	0.084	0.632	0.284	0.081	0.662	0.257	0.479	0.244	0.200	0.077
150	New Procedure	0.002	0.577	0.421	0.213	0.473	0.313	0.366	0.187	0.280	0.166
	SGC Procedure	0.000	0.524	0.476	0.140	0.477	0.383	0.232	0.211	0.328	0.229

From an examination of the numbers in Table 3.10, it can be seen, again that neither procedure is uniformly best when the underlying distribution is logistic. It appears that when the initial sample is contaminated with two outliers, there is a slight advantage to SGC's procedure. However, when the initial sample is contaminated with one or four outliers, the performance of the proposed procedure and that of SGC are comparable. For example, for a sample of size $n = 50$ from the $Logis(0,1)$ distribution in which four observations are replaced by outliers from the $Logis(3,1)$ distribution, the proposed procedure detects exactly four outliers about 20.7% of the time compared to the 24.4% of the time for the SGC procedure. However, for a sample of size $n = 150$ from the same underlying logistic distribution in which four observations are replaced by outliers from the $Logis(3,1)$ distribution, the proposed procedure detects exactly four outliers about 32.9% of the time while the SGC procedure detects the four outliers about 27.9% of the time.

In addition, the proposed procedure lends itself more naturally to statistical quality control, in phase I or II control charting. The fences are constructed from the median, which can be taken to be the centerline. The upper and lower fences will represent the upper and lower control limits of the chart, respectively. Such a control chart is expected to be more robust; results will be reported in a separate paper.

5. Summary and Conclusions

The traditional boxplot is a widely used tool in data analysis. Its popularity stems from the facts that it is easy to construct and it visually delivers a lot of useful information in one package about the underlying distribution. The lower fence of the traditional boxplot is

constructed as a constant k times the interquartile range, below the first quartile and the upper fence is constructed as the same constant k times the interquartile range, above the third quartile. The constant k is commonly chosen to be 1.5 or 3. The problem with this one size fits all approach, however, is that it does not take into account the shape of the underlying distribution and the sample size at hand. SGC show that such values of k may be inappropriate, particularly when the distribution is skewed, because they may lead to more false alarms (detecting an outlier when it is not). In order to find the fence constant k, they suggest instead controlling the *some-outside rate per sample* probability, and taking into account the sample size and the shape of the underlying distribution. We control the same probability as in SGC, but our fences are constructed as multiples of the semi-interquartile ranges measured from the median, as has been recommended by SOA (for asymmetric data). Between our procedure and SGC's, neither is uniformly better; but the simulation study shows that, in general, for exponential data, our procedure is slightly more powerful when the degree of contamination is higher. In addition, the proposed procedure lends itself more naturally to phase I control charting with skewed data; the results of this on-going research will be presented elsewhere.

APPENDIX

In what follows, f and F represent the pdf and cdf of the standardized random variable $Z = (X - \theta)/\sigma$ where X follows the location-scale cdf $F_{\theta,\sigma}$. Let $Z_{(1)} < \cdots < Z_{(l)} < \cdots < Z_{(m)} < \cdots Z_{(u)} < \cdots < Z_{(n)}$ be the order statistics of $Z_1, \ldots, Z_n$ for a random sample of size n from F.

The following result 1 is useful in establishing result 2 and result 4 below.

Result 1

$\Pr[Z_{(n)} < y_u | Z_{(m)} = z_{(m)}, Z_{(u)} = z_{(u)}] = I_{G_u(y_u)}(\alpha = n-u, \beta = 1)$

Where $I_{G_u(y_u)}(\alpha = n-u, \beta = 1)$, is an incomplete Beta function evaluated at $G_u(y_u) = \frac{F(y_u)-F(z_{(u)})}{1-F(z_{(u)})}$ and $y_u = z_{(m)} + k_u(z_{(u)} - z_{(m)})$

Proof

By the probability integral transformation, we have:

$\Pr[Z_{(n)} < y_u | Z_{(m)} = z_{(m)}, Z_{(u)} = z_{(u)}] = \Pr[W_{(n)} < F(y_u) | W_{(m)} = F(z_{(m)}), W_{(u)} = F(z_{(u)})]$

where $W_{(i)}$ is the i^{th} order statistic of a random sample of size n from uniform(0,1).

The conditional pdf of $W_{(n)}$ given $W_{(m)} and\ W_{(u)}$ is given by:

$$g_{W_{(n)}|w_{(m)},w_{(u)}}(w_{(n)}) = \frac{g(w_{(n)}w_{(m)}, w_{(u)})}{g_{1,2}(w_{(m)}, w_{(u)})} = (n-u)\frac{(w_{(n)} - w_{(u)})^{n-u-1}}{(1-w_{(u)})^{n-u}} I_{0<w_{(u)}<w_{(n)}<1}\ ;\ w_{(m)} < w_{(u)}$$

Then:

$$\Pr[W_{(n)} < F(y_u) | W_{(m)} = F(z_{(m)}), W_{(u)} = F(z_{(u)})] = \int_{F(z_{(u)})}^{F(y_u)} (n-u)\frac{(w_{(n)} - F(z_{(u)}))^{n-u-1}}{(1-F(z_{(u)}))^{n-u}} dw_{(n)}$$

The proof of the result is completed by a change of variable $t = \frac{w_{(n)}-F(z_{(u)})}{1-F(z_{(u)})}$ in the integral.

Next, we prove result 2, which is useful in the exact derivation of α (Equation 3.2).

Result 2

$$\Pr[\{Z_{(n)} > Z_{(m)} + k_u(Z_{(u)} - Z_{(m)})\}] = \int_{-\infty}^{\infty}\int_{z_{(m)}}^{\infty} [1 - I_{G_u(y_u)}(\alpha = n-u, \beta = 1)] f_{Z_{(m)},Z_{(u)}}(z_{(m)}, z_{(u)}) dz_{(u)} dz_{(m)}$$

Proof

$\Pr[\{Z_{(n)} > Z_{(m)} + k_u(Z_{(u)} - Z_{(m)})\}]$

$$= \int_{-\infty}^{\infty}\int_{z_{(m)}}^{\infty} [1 - \Pr(Z_{(n)} < y_u | Z_{(m)} = z_{(m)}, Z_{(u)} = z_{(u)})] f_{Z_{(m)},Z_{(u)}}(z_{(m)}, z_{(u)}) dz_{(u)} dz_{(m)}$$

where $y_u = z_{(m)} + k_u(z_{(u)} - z_{(m)})$ Result 2 now follows by using Result 1.

We now prove result 3, which is used in the derivation of result 4.

Result 3

$\Pr\left[Z_{(1)} < y_l | Z_{(l)} = z_{(l)}, Z_{(m)} = z_{(m)}\right] = I_{G_l(y_l)}(\alpha = 1, \beta = l-1)$, where $G_l(y_l) = \frac{F(y_l)}{F(z_{(l)})}$ and

$y_l = z_{(m)} - k_l(z_{(m)} - z_{(l)})$

Proof

It is easy to see that the conditional pdf of $W_{(1)}$ given $W_{(l)}$ and $W_{(m)}$ is given by:

$$g_{W_{(1)}|w_{(l)},w_{(m)}}(w_{(1)}) = \frac{g(w_{(1)}w_{(l)}, w_{(m)})}{g_{2,3}(w_{(l)}, w_{(m)})} = (l-1)\frac{(w_{(l)} - w_{(1)})^{n-u-1}}{{w_{(l)}}^{l-2}} I_{0<w_{(1)}<w_{(l)}} \; ; \; w_{(l)} < w_{(m)} < 1$$

Using the probability integral transformation,

$\Pr\left[Z_{(1)} < y_l | Z_{(l)} = z_{(l)}, Z_{(m)} = z_{(m)}\right] = \Pr\left[W_{(1)} < F(y_l) | W_{(l)} = F(z_{(l)}), W_{(m)} = F(z_{(m)})\right]$

$$= \int_0^{F(y_l)} (l-1)\frac{(w_{(l)} - w_{(1)})^{n-u-1}}{{w_{(l)}}^{l-2}} dw_{(1)}$$

Result 3 now follows by a change of variable: $t = \frac{w_1}{F(z_{(l)})}$ in the integral.

Next, we prove a lemma which will be used in the derivation of the exact expression of α (Equation 3.2).

Lemma

Let $1 \le r < l < m < u < k \le n$.

Then $f_{Z_{(r)},Z_{(k)}|z_{(l)},z_{(m)},z_{(u)}}(z_{(r)}, z_{(k)}) = f_{Z_{(r)}|z_{(l)},z_{(m)}}(z_{(r)}) f_{Z_{(k)}|z_{(m)},z_{(u)}}(z_{(r)})$

Proof

The conditional pdf of $Z_{(r)}$ and $Z_{(k)}$ given $Z_{(l)}, Z_{(m)}$ and $Z_{(u)}$ is given by:

$$f_{Z_{(r)},Z_{(k)}|z_{(l)},z_{(m)},z_{(u)}}(z_{(r)}, z_{(k)}) = \frac{f(z_{(r)}, z_{(l)}, z_{(m)}, z_{(u)}, z_{(k)})}{f(z_{(l)}, z_{(m)}, z_{(u)})}$$

$$= \frac{(n-u)!\,(l-1)!}{(r-1)!\,(l-r-1)!\,(k-u-1)!\,(n-k)!} \frac{\left[F(z_{(r)})\right]^{r-1}\left[F(z_{(l)}) - F(z_{(r)})\right]^{l-r-1}}{\left[F(z_{(l)})\right]^{l-1}}$$

$$X\frac{\left[F\left(z_{(k)}\right)-F\left(z_{(u)}\right)\right]^{k-u-1}\left[1-F\left(z_{(k)}\right)\right]^{n-k}}{\left[1-F\left(z_{(u)}\right)\right]^{n-u}}f(z_{(r)})f\left(z_{(k)}\right)I_{z_{(r)}<z_{(k)}}$$

where $z_{(r)} < z_{(l)} < z_{(m)} < z_{(u)} < z_{(k)}$.

Now, we find $f_{Z_{(r)}|z_{(l)},z_{(m)}}\left(z_{(r)}\right)f_{Z_{(k)}|z_{(m)},z_{(u)}}\left(z_{(k)}\right)$ and compare with the preceding expression to complete the proof.

It is easy to show that the conditional pdf of $Z_{(r)}$ given $Z_{(l)}, Z_{(m)}$ is given by:

$$f_{Z_{(r)}|z_{(l)},z_{(m)}}\left(z_{(r)}\right)=\frac{f_{Z_{(r)},Z_{(l)},Z_{(m)}}\left(z_{(r)},z_{(l)},z_{(m)}\right)}{f_{Z_{(l)},Z_{(m)}}\left(z_{(l)},z_{(m)}\right)}$$

$$=\frac{(l-1)!}{(r-1)!\,(l-r-1)!}\frac{\left[F\left(z_{(r)}\right)\right]^{r-1}\left[F\left(z_{(l)}\right)-F\left(z_{(r)}\right)\right]^{l-r-1}}{\left[F\left(z_{(l)}\right)\right]^{l-1}}f\left(z_{(r)}\right)\ where\ z_{(r)}<z_{(l)}<z_{(m)}$$

Similarly, the conditional pdf of $Z_{(k)}$ given $Z_{(m)}, Z_{(u)}$ is given by:

$$f_{Z_{(k)}|z_{(m)},z_{(u)}}\left(z_{(k)}\right)$$

$$=\frac{(n-u)!}{(k-u-1)!\,(n-k)!}\frac{\left[F\left(z_{(k)}\right)-F\left(z_{(u)}\right)\right]^{k-u-1}\left[1-F\left(z_{(k)}\right)\right]^{n-k}}{\left[1-F\left(z_{(u)}\right)\right]^{n-u}}f\left(z_{(k)}\right)\ where\ z_{(m)}<z_{(u)}<z_{(k)}$$

The proof of the lemma is completed by observing that

$$f_{Z_{(r)},Z_{(k)}|z_{(l)},z_{(m)},z_{(u)}}\left(z_{(r)},z_{(k)}\right)=f_{Z_{(r)}|z_{(l)},z_{(m)}}\left(z_{(r)}\right)f_{Z_{(k)}|z_{(m)},z_{(u)}}\left(z_{(k)}\right)$$

We now show result 4, which is useful in the derivation of Equation 3.2 (exact expression of α). The proof of result 4 requires result 1, result 3 and the lemma.

Result 4

$$\Pr\left[\left\{Z_{(1)}<Z_{(m)}-k_l\left(Z_{(m)}-Z_{(l)}\right),\ Z_{(n)}<Z_{(m)}+k_u\left(Z_{(u)}-Z_{(m)}\right)\right\}\right]=$$

$$\int_{-\infty}^{\infty}\int_{z_{(l)}}^{\infty}\int_{z_{(m)}}^{\infty}\left[I_{G_l(y_l)}(\alpha=1,\beta=l-1)I_{G_u(y_u)}(\alpha=n-u,\beta=1)\right]f_{Z_{(l)},Z_{(m)},Z_{(u)}}(z_{(l)},z_{(m)},z_{(u)})\,dz_{(u)}dz_{(m)}dz_{(l)},$$

where $y_l = z_{(m)} - k_l(z_{(m)} - z_{(l)})$, $G_l(y_l) = \frac{F(y_l)}{F\left(z_{(l)}\right)}$; y_u and $G_u(y_u)$ are previously defined.

Proof

We have:

$$\Pr\left[\left\{Z_{(1)} < Z_{(m)} - k_l\left(Z_{(m)} - Z_{(l)}\right),\ Z_{(n)} < Z_{(m)} + k_u\left(Z_{(u)} - Z_{(m)}\right)\right\}\right]$$

$$= \int_{-\infty}^{\infty}\int_{Z_{(l)}}^{\infty}\int_{Z_{(m)}}^{\infty} Pr\left[Z_{(1)} < y_l, Z_{(n)} < y_u | Z_{(l)} = z_{(l)}, Z_{(m)} = z_{(m)}, Z_{(u)} = z_{(u)}\right]$$

$$\text{x } f_{Z_{(l)}Z_{(m)}Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})dz_{(u)}dz_{(m)}dz_{(l)}$$

Using the lemma, this is equal to

$$\int_{-\infty}^{\infty}\int_{Z_{(l)}}^{\infty}\int_{Z_{(m)}}^{\infty} Pr\left[Z_{(1)} < y_l | Z_{(l)} = z_{(l)}, Z_{(m)} = z_{(m)}\right] Pr\left[Z_{(n)} < y_u | Z_{(m)} = z_{(m)}, Z_{(u)} = z_{(u)}\right]$$

$$\text{x } f_{Z_{(l)}Z_{(m)}Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})dz_{(u)}dz_{(m)}dz_{(l)}$$

Therefore, using results 1 and 3, we have that:

$$\Pr\left[\left\{Z_{(1)} < Z_{(m)} - k_l\left(Z_{(m)} - Z_{(l)}\right),\ Z_{(n)} < Z_{(m)} + k_u\left(Z_{(u)} - Z_{(m)}\right)\right\}\right] =$$

$$\int_{-\infty}^{\infty}\int_{Z_{(l)}}^{\infty}\int_{Z_{(m)}}^{\infty} \left[I_{G_l(y_l)}(1, l-1) I_{G_u(y_u)}(n-u, 1)\right] f_{Z_{(l)}, Z_{(m)}, Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})\, dz_{(u)}dz_{(m)}dz_{(l)}$$

This ends the proof of result 4.

We are now ready for the derivation of Equation (3.2).This derivation makes use of results 2 and 4.

Derivation of the exact expression of α (Equation 3.2)

Note that the event: "One or more of $X_1, X_2, \ldots, X_n \in out(\alpha_n, \theta, \sigma^2)$" is equivalent to the union of the two disjoint events: $\left\{X_{(n)} > UF_{sample}\right\} and\ \left\{X_{(1)} < LF_{sample}, X_{(n)} < UF_{sample}\right\}$, where $X_{(i)}$ is the i^{th} order statistics of a random sample of size n from the distribution X with cdf $F_{\theta,\sigma}(x)$.

Thus

$$\alpha = \Pr\left[\left\{X_{(n)} > X_{(m)} + k_u(X_{(u)} - X_{(m)})\right\}\right] + \Pr\left[\left\{X_{(1)} < X_{(m)} - k_l\left(X_{(m)} - X_{(l)}\right), X_{(n)} < X_{(m)} + k_u(X_{(u)} - X_{(m)})\right\}\right]$$

This yields

$$\alpha = \Pr\left[\left\{Z_{(n)} > Z_{(m)} + k_u(Z_{(u)} - Z_{(m)})\right\}\right] + \Pr\left[\left\{Z_{(1)} < Z_{(m)} - k_l\left(Z_{(m)} - Z_{(l)}\right), Z_{(n)} < Z_{(m)} + k_u\left(Z_{(u)} - Z_{(m)}\right)\right\}\right] \quad (3.3)$$

We now compute each term on the right hand side of Equation (3.3). Using Result 2, we get for the first term:

$$\Pr\left[\left\{Z_{(n)} > Z_{(m)} + k_u\left(Z_{(u)} - Z_{(m)}\right)\right\}\right]$$

$$= \int_{-\infty}^{\infty}\int_{Z_{(m)}}^{\infty}\left[1 - I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(m)},Z_{(u)}}\left(z_{(m)}, z_{(u)}\right) dz_{(u)} dz_{(m)}$$

where $y_u = z_{(m)} + k_u(z_{(u)} - z_{(m)})$ and $I_{G_u(y_u)}(\alpha = n-u, \beta = 1)$, is an incomplete Beta function evaluated at $G_u(y_u) = \frac{F(y_u) - F(z_{(u)})}{1 - F(z_{(u)})}$.

Similarly, using Result 4, we get for the second term:

$$\Pr\left[\left\{Z_{(1)} < Z_{(m)} - k_l\left(Z_{(m)} - Z_{(l)}\right), Z_{(n)} < Z_{(m)} + k_u\left(Z_{(u)} - Z_{(m)}\right)\right\}\right]$$

$$= \int_{-\infty}^{\infty}\int_{Z_{(l)}}^{\infty}\int_{Z_{(m)}}^{\infty}\left[I_{G_l(y_l)}(\alpha = 1, \beta = l-1) I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(l)},Z_{(m)},Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})\, dz_{(u)} dz_{(m)} dz_{(l)}$$

where $y_l = z_{(m)} - k_l(z_{(m)} - z_{(l)})$, $G_l(y_l) = \frac{F(y_l)}{F(z_{(l)})}$, y_u and $G_u(y_u)$ are previously defined.

Hence, we get for Equation (3.3):

$$\alpha = \int_{-\infty}^{\infty}\int_{Z_{(m)}}^{\infty}\left[1 - I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(m)},Z_{(u)}}\left(z_{(m)}, z_{(u)}\right) dz_{(u)} dz_{(m)}$$

$$+ \int_{-\infty}^{\infty}\int_{Z_{(l)}}^{\infty}\int_{Z_{(m)}}^{\infty}\left[I_{G_l(y_l)}(\alpha = 1, \beta = l-1) I_{G_u(y_u)}(\alpha = n-u, \beta = 1)\right] f_{Z_{(l)},Z_{(m)},Z_{(u)}}(z_{(l)}, z_{(m)}, z_{(u)})\, dz_{(u)} dz_{(m)} dz_{(l)},$$

which completes the derivation of Equation (3.2).

REFERENCES

Barnerjee, S., and Iglewicz, B. (2007), "A Simple Univariate Outlier Identification Procedure Designed for Large Samples," *Communication in Statistics - Simulation and Computation*, 36, 2, 249–263.

Barnett, V. (1978), "The Study of Outliers: Purpose and Model," *Applied Statistics*, 27, 3, 242-250.

Barnett, V., and Lewis, T. (1994), *Outliers in Statistical Data* (3rd ed.), Chichester: John Wiley.

Cerioli, A (2010), "Outlier Detection with High-breakdown Estimators," *Journal of the American Statistical Association*, 105, 147–156.

Carling, K. (2000), "Resistant Outlier Rules and the non-Gaussian Case," *Computational Statistics and Data Analysis*, 33, 249–258.

Cao D. S., Liang Y. Z., Xu Q. S., Li H. D., and Chen X. (2010), "A New Strategy of Outlier Detection for QSAR/QSPR," *Journal of Computational Chemistry*, 31, 3, 559-602.

Dang X., and Serfling R. (2010), "Nonparametric Depth-based Multivariate Outlier Identifiers, and Masking Robustness Properties," *Journal of Statistical Planning and Inference*, 140, 198-213.

Daniel, C. (1959), "Use of Half-normal Plots in Interpreting Factorial Two-level Experiments," *Technometrics*, 1, 311–341.

Hadi, A. S., Imon, A. H. M. R., and Werner, M. (2009), "Detection of Outliers," *WIREs Computational Statistics*, 1, 57–70.

Hawkins, D. M., (2006), "Outliers," *Encyclopedia of Statistical Sciences*, New York: John Wiley.

Hoaglin, D. C., Iglewicz, B., and Tukey, J. W. (1986), "Performance of Some Resistant Rules for Outlier Labeling," *Journal of the American Statistical Association*, 81, 991–999.

Kimber, A.C. (1990), "Exploratory Data Analysis for Possibly Censored Data from Skewed Distribution," *Applied Statistics*, 39, 21–30.

Louni, H. (2008), "Outlier Detection in ARMA Models," *Journal of Time Series Analysis*, 29, 6, 1057-1065.

Montgomery, D. C. (2009), *Introduction to Statistical Quality Control* (6th ed.), New York: John Wiley.

Schwertman, N. C., and de Silva, R. (2007), "Identifying Outliers With Sequential Fences," *Computational Statistics and Data Analysis*, 51, 3800–3810.

Schwertman, N. C., Owens, M. A., and Adnan, R. (2004), "A Simple More General Boxplot Method for Identifying Outliers," *Computational Statistics and Data Analysis*, 47, 165–174.

Sim, C. H., Gan, F. F., and Chang, T. C. (2005), "Outlier Labeling With Boxplot Procedures," *Journal of the American Statistical Association*, 100, 642–652.

Tukey, J. W. (1977), *Exploratory Data Analysis*, New York: Addison-Wesley.

CHAPTER 4

BOXPLOT-BASED PHASE I CONTROL CHARTS FOR TIME BETWEEN EVENTS

Abstract

To monitor the quality/reliability of a (production) process, it is sometimes advisable to monitor the time between certain events (say occurrence of defects) instead of the number of events, particularly when the events occur rarely. In this case, it is common to assume that the times between the events follow an exponential distribution. In this paper, we propose a one-sided and a two-sided control chart for phase I data following an exponential distribution. The control charts are derived from a modified boxplot procedure proposed in Dovoedo and Chakraborti (2009). The chart constants are obtained by controlling the overall Type I error rate. The in-control robustness and the out-of-control performance of the proposed charts are examined and compared with those of the charts by Jones and Champs (2002) in a simulation study. It is seen that the proposed charts are considerably more in-control robust and have out-control properties comparable to the Jones and Champs (2002) charts. Charting constants are provided and a numerical example using some data is given for illustration.

Keywords: control chart; exponential distribution; phase I; Poisson process; reliability

1. Introduction

Reliability of a product or a process is often examined in terms of the failures that occur. Assuming that failures occur according to a homogeneous Poisson process, the times between

the failures (events) are known to follow an exponential distribution. This is a standard assumption in many quality/reliability studies. From a practical point of view, it is desirable that failures occur rarely. However, when failure is a rare event, it becomes difficult to understand the reliability/quality of the process under consideration and improve it, by monitoring the failures. This is because a substantial amount of time may go by before a failure is observed and it may be impractical to wait that long to gather data. In this case, Montgomery (2009) recommends that the time between failures be monitored with regard to the quality of the process. Several control charts have been proposed for this purpose in the literature (Lucas, 1985; Vardeman and Ray, 1985; Gan, 1988). Liu et al. (2006) compare some of these charts including the CUSUM and EWMA charts for time between events. Some authors have suggested transforming the exponential data to symmetry then apply the regular Shewhart chart. These authors include Nelson (1994), Kittlitz (1999) and Kao (2010). We do not pursue this route here.

Jones and Champ (2006) (hereafter JC) point out that all the control charts mentioned above either assume that the in-control process parameter (the mean) is known or that it is unknown and is estimated from an in-control phase I reference sample, but the problem of phase I control charting, i.e., obtaining a reference sample wasn't considered. The phase I or the retrospective phase is an important component of an overall control charting regime where the objectives are somewhat different from the phase II or the monitoring phase. The reader is referred to Chakraborti et al. (2009) for an overview of phase I control charts. More recently, Ozsan et al. (2010) study the effects of parameter estimation (from phase 1) on some performance measures of the exponential EWMA control chart and found that the effect of estimation can be serious, especially if small samples are used. As in JC, this paper focuses on the development of Phase I control charts when *m* independent samples each of size $n = 1$ are

taken on the quality measurement periodically and each sample is assumed to have come from an exponential distribution with an unknown mean. JC point out that for a phase I chart one should control the overall false alarm rate (the probability of at least one false alarm) and not the individual false alarm rate since the overall false alarm rate is seriously inflated otherwise. For example, when $m = 200$ and the control limits are set such that the individual false alarm rate is $\alpha_1 = 0.0027$ the overall false alarm rate is $\alpha = 1-(1-0.0027)^{200} = 0.4177$. Clearly, this is too high an overall false alarm rate, especially when the cost of investigating false alarms is prohibitively high from an operational point of view. Thus, it is of interest to control the overall false alarm rate while designing a phase I control chart, as has been recommended in the literature.

In this paper, new phase I control charts are proposed based on the modified boxplot proposed by Dovoedo and Chakraborti (2009). A boxplot is one of the most popular graphical statistical tools used in practice and the process personnel may already be familiar with it. The idea of constructing control charts from boxplots is not new (see for example, Iglewicz and Hoaglin, 1987, Alemi, 2004). The proposed control charts are constructed for individual observations from an exponential distribution with an unknown mean so that the overall false alarm rate is controlled at a given nominal value.

Robustness is a desirable property of a control chart so that its in-control performance is stable. Robustness of control charts has been investigated for various charts. For example, Pehlivan and Testik (2010) find that the time between events EWMA chart can be designed to be robust to the departure of the assumed exponential distribution. Also, Borror et al. (2003) came to a similar conclusion with respect to the time between events CUSUM chart. Because the proposed control charts are constructed from the median of the data (see more details later), the

control limits are expected to be more robust than those proposed by JC, which are based on the mean, in the presence of one or more unusual out-of-control observations. In addition, the two-sided control chart by JC is approximate, while the proposed chart is exact.

This paper is structured as follows. In section 2, the design of the one-sided control chart is considered and the proposed chart is developed. The design of the two-sided control chart is considered and the proposed chart is developed in section 3. Section 4 compares the in-control robustness, to the assumption of exponential data, of the proposed control charts and the charts by JC via a simulation study. In addition, the out-of-control performances of the proposed control charts are compared with those by JC in section 5 via simulation. In section 6, a numerical example is given for illustration. Summary of findings and conclusions are offered at the end.

2. Control Limit for the One-sided Control Chart

In case of a one-sided control chart, the interest is often in the lower control limit. This is because smaller (shorter) times between failures are a sign of process deterioration. Suppose there are m random times between failures, X_i where X_i follows an exponential distribution with an unknown mean $\mu_i = 0.0027,\ i = 1,...,m$. The process is in-control at time i when $\mu_i = \mu$ (unknown). Denote $Y_i = \frac{X_i}{\bar{X}},\ i = 1,...,m$, where $\bar{X} = (1/m)\sum_{i=1}^{m} X_i$ is the sample average and let α denote the overall false alarm rate. JC use the joint probability distribution of the Y_i's to establish the expression of the lower control limit LCL in order to maintain the overall false alarm rate at a nominal level α_0. Their lower control limit is given by

$$LCL = [1-(1-\alpha_0)^{1/(m-1)}]\bar{X}, \tag{4.1}$$

and their center line is given by $CL = \bar{X}$.

For the proposed one-sided control chart, let $X_{(1)}, X_{(2)}, ..., X_{(m)}$ denote the corresponding ordered observations and let $X_{(l)}, X_{(m_1)}, X_{(u)}$ denote the first, second and third quartiles, respectively. Dovoedo and Chakraborti (2009) use the following definition for the indices l, m_1, and u: $m_1 = ceil(m/2)$; $l = floor(m/4)+1$ if $mod(m,4) \neq 0$ and $l = floor(m/4)$ otherwise, where $ceil(a)$ denotes the smallest integer greater than or equal to a, $floor(a)$ denotes the largest integer less than or equal to a and $mod(a,b)$ denotes the remainder in the division of an integer a by an integer b; $u = m - l + 1$. This is because if the indices are not integers, the joint distributions of order statistics $X_{(l)}, X_{(m_1)}, X_{(u)}$ are intractable and the exact derivation of the probability α is difficult. Following Dovoedo and Chakraborti (2009), we propose the following lower control limit and center line for the one-sided chart, in which the constant k_l is appropriately determined to control α.

$$LCL = X_{(m_1)} - k_l(X_{(m_1)} - X_{(l)}) \tag{4.2}$$

$$CL = X_{(m_1)}$$

We now derive an exact expression for α, the overall false alarm rate, which is the probability that at least one of the observations fall below the lower control limit when the process is in-control (*IC*). This expression will be used to find the lower fence constant k_l for a given nominal value α_0. Note that the event that at least one observation falls below the lower control limit is equivalent to $X_{(1)} < LCL$. We can then write

$$\alpha_0 = \Pr[X_{(1)} < X_{(m_1)} - k_l(X_{(m_1)} - X_{(l)}) \mid IC]$$

By standardizing (that is by dividing the respective ordered *X*'s in the above equation by the common mean μ) and by conditioning on the resulting $Z_{(l)}$ and $Z_{(m)}$, we obtain, successively:

$$\alpha_0 = \Pr[z_{(1)} < Z_{(m_1)} - k_l(Z_{(m_1)} - Z_{(l)}) \mid IC]$$

$$= \int_0^{+\infty} \int_{z_{(l)}}^{+\infty} \Pr[Z_{(1)} < y_{(l)} \mid Z_{(l)} = z_{(l)}, Z_{(m_1)} = z_{(m_1)}, IC] f_{Z_{(l)}, Z_{(m1)}}(z_{(l)}, z_{(m_1)}) dz_{(m_1)} dz_{(l)},$$

where $y_{(l)} = z_{(m_1)} - k_l(z_{(m_1)} - z_{(l)})$, $Z_{(i)} = X_{(i)} / \mu$ and $f_{Z_{(l)}, Z_{(m1)}}$ denotes the joint distribution of two order statistics $Z_{(l)}$ and $Z_{(m_1)}$ in a random sample of size *m* from the exponential distribution with mean 1.

It is shown in Dovoedo and Chakraborti (2009) that:

$$\Pr[Z_{(1)} < y_{(l)} \mid Z_{(l)} = z_{(l)}, Z_{(m_1)} = z_{(m_1)}, IC] = I_{G_l(y_{(l)})}(\alpha = 1, \beta = l - 1),$$

where $I_{G_l(y_{(l)})}(\alpha = 1, \beta = l - 1)$ is an incomplete Beta function evaluated at $G_l(y_{(l)}) = \dfrac{F(y_{(l)})}{F(z_{(l)})}$ and $z_{(i)}$ is the value of $Z_{(i)}$, the i^{th} order statistic of a random sample of size *m* from the distribution of the standardized random variable $Z = X / \mu$ which follows an exponential distribution with mean 1; F is the cdf of Z.

Thus, we have

$$\alpha_0 = \int_0^{+\infty} \int_{z_{(l)}}^{+\infty} I_{G_l(y_{(l)})}(\alpha = 1, \beta = l - 1) f_{Z_{(l)}, Z_{(m1)}}(z_{(l)}, z_{(m_1)}) dz_{(m_1)} dz_{(l)} \qquad (4.3)$$

Note that Equation (4.3) actually holds for any location-scale distribution, including the exponential distribution, which is our focus here.

The right hand side of Equation (4.3), which depends on k_l, is set up in Mathematica. A direct search is used to find the constant k_l that makes that right hand side equal to some desired

nominal α_0. Again, once the fence constant k_l is found, this is used in the construction of the lower control limit of the proposed one-sided control chart and the median $X_{(m_1)}$ is taken as the center line for the proposed control chart. In Table 4.1, we provide the charting constant k_l for some selected values of m and some typical values of the overall false alarm rate. The size of the phase I sample, *m*, is often recommended to be between 20 and 100 but we also provide the constants for *m* = 150. For larger values of *m*, one can use simulations to determine the charting constants as both Mathcad or Mathematica implementations become unstable.

Table 4.1: Charting constants for the proposed one-sided control chart

	Number of phase I observations, *m*					
α_0	77	30	50	75	100	150
0.01	3.995	3.695	2.922	2.465	2.346	2.234
0.05	2.818	2.804	2.433	2.164	2.102	2.05
0.1	2.406	2.472	2.228	2.033	1.993	1.963
0.2	2.035	2.153	2.021	1.891	1.875	1.868

3. Control Limits for the Two-sided Control Chart

In some cases, one prefers to use a two-sided control chart. JC uses the fact that when the process is in-control, in case of exponential observations, $Y_i = \frac{X_i}{\bar{X}}$ (using earlier notation) is related to an *F*-distribution and Boole's inequality to derive approximate expressions for the lower and upper control limits to maintain the overall false alarm rate at the nominal level α_0. The approximate lower and upper control limits are:

$$LCL = \frac{m\bar{X}}{1+(m-1)F_{2(m-1),2,1-\alpha_0/m+\tau}}$$

$$UCL = \frac{m\bar{X}}{1+(m-1)F_{2(m-1),2,\tau}} \qquad (4.4)$$

where the F-distribution has numerator and denominator degrees of freedom $2(m-1)$ and 2; and τ satisfies $0 < \tau < \alpha_0 / m$. The center line is given by $CL = \bar{\bar{X}}$.

The proposed two-sided control chart is derived in a simple manner from the adjusted boxplot proposed by Dovoedo and Chakraborti (2009); their fences are now used as the control limits. Thus the lower and the upper control limits are $X_{(m_1)} - k_l(X_{(m_1)} - X_{(l)})$ and $X_{(m_1)} + k_u(X_{(u)} - X_{(m_1)})$, respectively. They establish an exact expression for the probability that at least one observation from an uncontaminated sample from any location-scale distribution is falsely classified as an outlier (falls outside the fences). In the present case, this is the probability that the chart signals at least one false alarm. The derived expression is given by

$$\alpha = \int_0^{+\infty} \int_{z_{(m_1)}}^{+\infty} [1 - I_{G_u(y_{(u)})}(\alpha = m-u, \beta = 1)] f_{Z_{(m_1)}, Z_{(u)}}(z_{(m_1)}, z_{(u)}) dz_{(u)} dz_{(m_1)}$$

$$+ \int_{-\infty}^{+\infty} \int_{z_{(l)}}^{+\infty} \int_{z_{(m_1)}}^{+\infty} [I_{G_l(y_{(l)})}(\alpha = 1, \beta = l-1) I_{G_u(y_{(u)})}(\alpha = m-u, \beta = 1)] f_{Z_{(l)}, Z_{(m_1)}, Z_{(u)}}(z_{(l)}, z_{(m_1)}, z_{(u)}) dz_{(u)} dz_{(m_1)} dz_{(l)} \tag{4.5}$$

where $y_{(u)} = z_{(m_1)} + k_u(z_{(u)} - z_{(m_1)})$, $I_{G_u(y_{(u)})}(\alpha = m-u, \beta = 1)$ is an incomplete Beta function evaluated at $G_u(y_{(u)}) = \dfrac{F(y_{(u)}) - F(z_{(u)})}{1 - F(z_{(u)})}$, $y_{(l)} = z_{(m_1)} - k_l(z_{(m_1)} - z_{(l)})$, $I_{G_l(y_{(l)})}(\alpha = 1, \beta = l-1)$ is an incomplete Beta function evaluated at $G_l(y_{(l)}) = \dfrac{F(y_{(l)})}{F(z_{(l)})}$.

As explained in Dovoedo and Chakraborti (2009), the constants k_l and k_u can be found by solving Equation (4.5) for a given (nominal) overall false alarm rate α say α_0. This is done using a direct search method in Mathcad or Mathematica. Note that the first term on the right hand side of Equation (4.5) depends on k_u but not on k_l, while the second term depends on both

k_u and k_l. Thus, one can first find k_u to make the first term equal to $\alpha_0/2$ and then find k_l that makes the second term also equal to $\alpha_0/2$.

The control limits of the proposed two-sided control chart are given by

$$LCL = X_{(m_1)} - k_l(X_{(m_1)} - X_{(l)}),$$

$$UCL = X_{(m_1)} + k_u(X_{(u)} - X_{(m_1)}), \qquad (4.6)$$

and the center line is $CL = X_{(m_1)}$

The charting constants k_u and k_l are provided in Table 4.2 for some selected values of *m* and the overall nominal false alarm rate α_0. The nominal false alarm rate values investigated here are $\alpha_0 = 0.01, 0.05$ and 0.1 as in JC but additionally we also take $\alpha_0 = 0.2$.

Table 4.2: Charting constants for the proposed two-sided control chart

	No. of phase I observations, *m*											
	20		30		50		75		100		150	
α_0	k_l	k_u	k_l	k_u	k_l	k_u	k_l	k_u	k_l	k_u	k_l	k_u
0.01	4.617	15.56	4.136	17.224	3.144	15.983	2.602	15.535	2.451	14.902	2.352	14.815
0.05	3.265	10.32	3.16	12.096	2.631	11.988	2.294	12.068	2.207	11.829	2.127	12.104
0.1	2.787	8.442	2.785	10.16	2.417	10.387	2.157	10.637	2.096	10.539	2.05	11.128
0.2	2.348	6.756	2.425	8.36	2.197	8.837	2.012	9.224	1.973	9.253	1.949	9.967

4. Comparison of In-Control Robustness

The JC control charts are based on the mean while the proposed charts are based on the median. Here, we first investigate the in-control robustness of the two one-sided control charts to the assumption of the underlying exponential distribution via simulation. Similar comparisons are thendone for the two-sided charts. The in-control robustness is an important attribute of a control chart and should be investigated since in practice the underlying distribution may not be exactly exponential. The more robust the control chart, the more confidence the user will have on the advertised false alarm rate associated with that control chart. Without the assurance of a

robust false alarm rate, the performance of a control chart in detecting changes by a out-of-control signal becomes somewhat meaningless.

For this study, we consider four alternative distributions: two slightly more and two slightly less skewed than the exponential distribution. The two more skewed distributions are the *Gamma*(1.1,1) and the *Gamma*(1.2,1) distributions, respectively, and the two less skewed distributions are the *Gamma*(0.8,1) and the *Gamma*(0.9,1) distributions. Note that the skewness of a distribution can be well-characterized by a robust measure called the medcouple (*MC*) introduced by Brys et al. (2003). The *MC* value of the exponential distribution (with mean 1) is 0.3333 whereas the two more skewed distributions have *MC* values 0.3809 and 0.3547 and the two less skewed distributions have *MC* values, 0.3151 and 0.2994, respectively.

Results in this section are based on 100,000 simulations with $m = 30$ observations. The results for the one-sided control charts are reported in the Table 4.3. The 95% margin of error for the empirical overall false alarm rates is at most 0.003. For the simulation results reported in Table 4.3 (one-sided charts case), the (lower) control limits for the proposed chart are computed using Equation (4.2), where the fence constants k_l are found from Table 4.1, and the control limits for JC are computed using Equation (4.1).

Table 4.3: Empirical overall false alarm rates for the one-sided control charts

		Distribution			
α_0	Method	*Gamma*(1.2,1)	*Gamma*(1.1,1)	*Gamma*(0.9,1)	*Gamma*(0.8,1)
0.01	Proposed	0.01657	0.01321	0.00761	0.00512
	JC	0.0024	0.00487	0.02055	0.04377
0.05	Proposed	0.07698	0.06313	0.03867	0.02775
	JC	0.01551	0.02833	0.08742	0.15255
0.1	Proposed	0.14572	0.1235	0.07875	0.0578
	JC	0.03844	0.06089	0.16066	0.25457
0.2	Proposed	0.27296	0.23688	0.16388	0.12548
	JC	0.09023	0.13446	0.2927	0.41461

It can be seen that in almost all cases, the empirical false alarm rates for JC's chart deviate more greatly from the nominal overall false alarm rate α_0. The absolute deviations between the empirical false alarm rate and the nominal false alarm rate α_0 are larger for JC's procedure in all of the 16 cases. In other words, for all 16 cases, the proposed procedure is more robust in terms of the overall false alarm rate and the difference with JC's procedure can be substantial. For example, for the *Gamma*(0.8,1) distribution with $\alpha_0 = 0.2$, the absolute deviation is $|0.2 - 0.41461| = 0.21461$ for the JC procedure, while it is only $|0.2 - 0.12548| = 0.07452$ for the proposed one. Moreover, the variation among the empirical overall false alarm rates is seen to be substantially higher for the JC control chart for any given nominal probability α_0.

The results for the two-sided control charts are shown in Table 4.4. The 95% margin of error for the empirical overall false alarm rates is at most 0.00302 at most.

For the simulation results reported in Table 4.4 (two-sided charts case), the control limits for the proposed chart are computed using Equation (4.6), where the fence constants k_l and k_u are found from Table 4.2, and the control limits for JC are computed using Equation (4.4) where the parameter τ was taken to be $\alpha_0 / 2m$, the mid-value between 0 and α_0 / m.

Table 4.4: Empirical overall false alarm rates for the two-side control charts

		Distribution			
α_0	Method	*Gamma*(1.2,1)	*Gamma*(1.1,1)	*Gamma*(0.9,1)	*Gamma*(0.8,1)
0.01	Proposed	0.01187	0.01072	0.00974	0.00994
	JC	0.0026	0.00483	0.02016	0.04175
0.05	Proposed	0.0582	0.05588	0.04922	0.04826
	JC	0.01641	0.02833	0.08446	0.14418
0.1	Proposed	0.11833	0.10739	0.09635	0.09364
	JC	0.03804	0.05992	0.15237	0.2406
0.2	Proposed	0.23123	0.21306	0.19005	0.18259
	JC	0.08393	0.1231	0.26838	0.38343

Again, from Table 4.4, we reach the same conclusion as for Table 4.3. The empirical false alarm rates for the JC chart deviate more greatly from the desired overall nominal false alarm rate α_0. Specifically, the absolute deviations of the empirical false alarm rate from the nominal false alarm rate are smaller (better) for the two-sided JC procedure in none of 16 cases studied here. In all these cases the proposed two-sided procedure is more robust, and the differences can be substantial. For example, for the *Gamma*(1.2,1) distribution, using $\alpha_0 = 0.2$, the deviation is $|0.2 - 0.08393| = 0.11607$ for the JC chart, while it is only $|0.2 - 0.23123| = 0.03123$ for the proposed chart. Moreover, like in the case of the one-sided control chart, the variation among the empirical overall false alarm rates is seen to be substantially higher for the JC control chart for any given nominal probability α_0.

In summary, the phase I control charts by JC are seen to suffer from a lack of in-control robustness which can be problematic for its practical applications. In fact the more (or less) skewed the underlying distribution is, compared to the exponential distribution (with mean 1), the worse the in-control robustness performance is of the JC charts. The proposed charts are seen to be more in-control robust. This is most likely due to the fact that these are based on the median of the data set and not the mean.

5. Comparison of Out-of-Control Performance

In order to examine the performance of our charts and compare with JC's charts, we estimate the probability of at least one out-of-control signal in a simulation study. This is done by calculating the proportion of cases where the charts signal out-of-control in the simulations, for several out-of-control situations. The simulation scheme is similar to the scheme adopted by

JC: m is the number of observations and t denotes the number of out-of-control observations among the *m* observations. The quantity δ measures the shift in the process. Negative values of δ are of more interest as they indicate process deterioration. The steps are as follows for each chart for each chart:

(i) Generate $m-t$ observations from an exponential distribution with mean μ_0

(ii) Generate t observations from an exponential distribution with mean $\mu_0 + \delta\mu_0$

(iii) Compute the control limit(s) for a given overall false alarm rate α_0 using Equations (4.2) and (4.6) for our charts and using Equations (4.1) and (4.4) for JC's charts

(iv) In case of a one-side control chart, increase a counter b by one if at least one observation (from the combined sample of *m* observations) falls below the lower control limit; and in case of the two-sided control chart, increase the counter by 1 if at least one observation falls either below the lower control limit or above the upper control limit

(v) Repeat Steps (i)–(iv) a large number (*M*) of times

(vi) Calculate b/M, the proportion of times the chart produces at least one out-of-control signal. This proportion provides the estimated error rate.

The shift sizes considered here are not too extreme. This is because an extreme shift like $\delta = -0.99$ are rare in practice. Specifically, we use the following values for δ: -0.05; -0.1; -0.25 and -0.5. Note that negative shift imply a decrease in mean, and deterioration of process quality because we are monitoring times between failures. Note also that $\delta = -0.5$ means that the process mean has decreased by half.

Tables 4.5 and 4.6 show the performance of the one-sided and two-sided control charts, respectively, in terms of the observed proportions of out-of-control signals. All results are based

on 100,000 simulations with $m = 30$. The 95% margin of error for reported results in Table 4.5 and Table 4.6 is approximately 0.00295 at most.

From Table 4.5 we see that the performance of our one-sided chart is comparable to that of JC's, and in fact, there is no clear winner in all cases. The same is true for the two-sided control charts shown by the results in Table 4.6. For example, in case of the two-sided charts, for a nominal in-control false alarm rate $\alpha_0 = 0.05$, with 3 out of the 30 observations replaced by out-of-control observations from $Expo(0.95)$ ($\delta = -0.05$), the proportion of charts with at least one signal is 0.04975 = 4.975% for the JC procedure, while it is 0.05052 = 5.052% for the proposed chart. When $\alpha_0 = 0.2$, with 3 out of the 30 observations replaced by out-of-control observations from the $Expo(0.5)$ ($\delta = -0.5$), the proportion of charts with at least one signal is 0.20506 = 20.506% for the JC procedure while it is 0.20458 = 20.458% for the proposed chart. Thus, it seems fair to conclude that the two charts have similar out-of-control performance.

Table 4.5: Out-of-control performance of the one-sided phase I control charts

			Shift Size δ			
α_0	t	Method	-0.05	-0.1	-0.25	-0.5
0.01	1	Proposed	0.01008	0.00972	0.01065	0.01046
		JC	0.00996	0.00962	0.00948	0.00992
	3	Proposed	0.01094	0.01035	0.00978	0.00947
		JC	0.00985	0.01004	0.01012	0.01078
	5	Proposed	0.01009	0.00987	0.00974	0.00908
		JC	0.00985	0.01039	0.0097	0.01161
	10	Proposed	0.01031	0.00973	0.0097	0.00866
		JC	0.00977	0.01001	0.01034	0.01056
0.05	1	Proposed	0.05078	0.05003	0.051	0.05095
		JC	0.05084	0.04978	0.04843	0.04976
	3	Proposed	0.05258	0.05057	0.05026	0.04806
		JC	0.0505	0.04912	0.05011	0.0534
	5	Proposed	0.04916	0.05165	0.0505	0.04513
		JC	0.05044	0.05059	0.05118	0.05404
	10	Proposed	0.05052	0.05029	0.04908	0.0444
		JC	0.05021	0.05062	0.05153	0.0551
0.1	1	Proposed	0.10098	0.10015	0.10006	0.099
		JC	0.10218	0.10117	0.09934	0.09986
	3	Proposed	0.10182	0.10083	0.09957	0.09487
		JC	0.1004	0.09916	0.09982	0.10498
	5	Proposed	0.09817	0.10096	0.10035	0.09044
		JC	0.10019	0.10026	0.10302	0.10649
	10	Proposed	0.10132	0.09944	0.09815	0.08973
		JC	0.09965	0.10079	0.10214	0.11163
0.2	1	Proposed	0.20154	0.1996	0.19978	0.19705
		JC	0.20228	0.20144	0.20011	0.19953
	3	Proposed	0.20218	0.19859	0.19948	0.19107
		JC	0.19882	0.20035	0.20082	0.2093
	5	Proposed	0.19896	0.20149	0.1978	0.185
		JC	0.20056	0.20046	0.20414	0.21286
	10	Proposed	0.20154	0.19878	0.1962	0.18142
		JC	0.19917	0.20158	0.20309	0.22043

Table 4.6: Out-of-control performance of the two-sided phase I control charts

			Shift Size δ			
α_0	t	Method	-0.05	-0.1	-0.25	-0.5
0.01	1	Proposed	0.01069	0.01013	0.0094	0.01056
		JC	0.01012	0.01021	0.01005	0.01079
	3	Proposed	0.01053	0.00986	0.01005	0.01048
		JC	0.00982	0.0095	0.01067	0.01217
	5	Proposed	0.01004	0.01011	0.01036	0.01132
		JC	0.00968	0.01003	0.01093	0.01317
	10	Proposed	0.00984	0.0101	0.01107	0.01366
		JC	0.0102	0.01057	0.01208	0.01891
0.05	1	Proposed	0.05063	0.05046	0.04992	0.05199
		JC	0.04926	0.04867	0.04924	0.05186
	3	Proposed	0.05052	0.04865	0.05063	0.05183
		JC	0.04975	0.04814	0.05187	0.05632
	5	Proposed	0.05065	0.0507	0.05201	0.05403
		JC	0.04854	0.04923	0.05194	0.0617
	10	Proposed	0.05009	0.05135	0.05197	0.06134
		JC	0.04937	0.05006	0.05486	0.07889
0.1	1	Proposed	0.10125	0.09886	0.10059	0.09921
		JC	0.09561	0.09666	0.09809	0.09988
	3	Proposed	0.0997	0.10017	0.10212	0.10214
		JC	0.09474	0.09628	0.09968	0.1083
	5	Proposed	0.09949	0.10111	0.10418	0.10526
		JC	0.0952	0.0979	0.10152	0.11889
	10	Proposed	0.09895	0.10047	0.10517	0.11925
		JC	0.09695	0.09598	0.10613	0.14336
0.2	1	Proposed	0.20095	0.19779	0.2024	0.20086
		JC	0.18282	0.18269	0.18595	0.1893
	3	Proposed	0.19912	0.19848	0.20099	0.20458
		JC	0.18456	0.18293	0.18835	0.20506
	5	Proposed	0.20032	0.19689	0.2028	0.20816
		JC	0.18489	0.1818	0.19237	0.2167
	10	Proposed	0.20034	0.20086	0.2085	0.22539
		JC	0.18326	0.18545	0.1967	0.25543

6. Illustration: Application to Data

As an illustration, consider the example in Montgomery (2009) in which a chemical engineer wishes to control the average time between failures of a valve. She observed twenty

times between failures for this valve. The data are shown in Table 4.7 and were used by JC as an illustration of their two-sided control chart.

Table 4.7: Times between failures data

286	948	536	124	816	729	4	143	431	8
2837	596	81	227	603	492	1199	1214	2831	96

Note that the data with all twenty observations do not fail the Anderson Darling test for exponential distribution. From Minitab, the Anderson Darling statistic is found to be 0.53 with a P-value = 0.44.

The quartiles for these data are given by: $q_1 = X_{(5)} = 124$, $q_2 = X_{(10)} = 492$, and $q_3 = X_{(16)} = 948$. For $m = 20$ and $\alpha = 0.1$, Table 4.2 gives the lower and upper chart constants: $k_l = 2.787$ and $k_u = 8.442$, respectively. Hence, the centerline for the proposed two-sided control chart is $CL = X_{(10)} = 492$ and the control limits are given by:

$$UCL = q_2 + k_u(q_3 - q_2) = X_{(10)} + k_u(X_{(16)} - X_{(10)}) = 4341.552$$

$$LCL = q_2 - k_l(q_2 - q_1) = X_{(10)} - k_l(X_{(10)} - X_{(5)}) = -533.616$$

Because $LCL < 0$ we use $LCL = 0$. The control chart in Figure 4.1 shows no out-of-control signals. This coupled with the fact that an exponential distribution fit the data well (as previously mentioned), we conclude that these data can be used as reference data and these control limits can then be used for phase II monitoring. Note that JC two-sided control chart leads to the same conclusion.

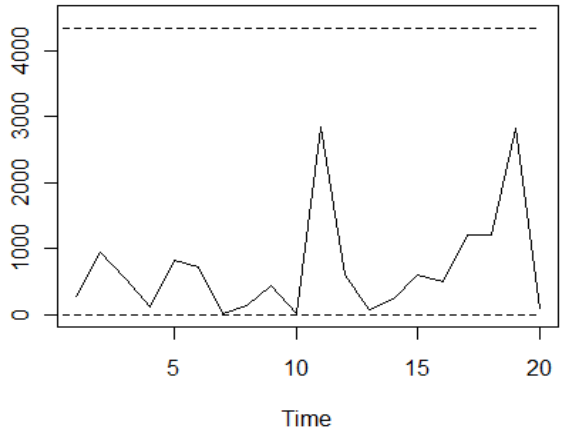


Figure 4.1. Phase I control chart for times between failures

7. Summary and Conclusions

Phase I control charts are considered for observations from an exponential distribution with an unknown mean. The charts are based on the sample median and are motivated by a modified boxplot procedure considered by Dovoedo and Chakraborti (2009). Since the boxplot is a part of routine exploratory data analysis, the proposed charts should be attractive to the practitioners. The necessary chart constants are tabulated for a number of sample sizes and are obtained by controlling the overall false alarm rate. Simulation results show that the proposed charts have out-of-control performance similar to the control charts by JC but they are much more in-control robust. Hence, the proposed charts are recommended to be used in practice.

REFERENCES

Alemi, F. (2004), "ARL Performance of Tukey's Control Chart," *Quality Management in Health Care*, 13, 216–221.

Borror C.M., Keats J.B., and Montgomery D.C. (2003), "Robustness of the time between events CUSUM," *International Journal of Production Research*, 41, 3435-3444.

Brys, G., Hubert, M., and Struyf, A. (2003), "A Comparison of Some New Measures of Skewness," in: *Developments in Robust Statistics (ICORS 2001),* eds. Dutter, R., Filzmoser, P., Gather, U., and Rousseeuw P. J., 98-113, Heidelberg: Springer-Verlag.

Chakraborti, S., Human, S. W., and Graham, M. A. (2009), "Phase I Statistical Process Control Charts: An Overview and Some Results," *Quality Engineering*, 21**,** 52-62.

Dovoedo, Y. H., and Chakraborti, S. (2009), "On Some Properties of a Simple and More General Boxplot-type Method for Identifying Outliers," *American Statistical Association Proceedings of the Statistical graphics section*, Alexandria, VA, pp. 3900-3908.

Gan F. F. (1998), "Designs of One- and Two-sided Exponential EWMA Charts," *Journal of Quality Technology*, 30, 55–69.

Iglewicz, B., and Hoaglin, D. C. (1987), "Use of Boxplots for Process Evaluation," *Journal of Quality Technology*, 19, 180-190.

Jones L. A., and Champs, C. W. (2002), "Phase I Control Charts for Times Between Events," *Quality and Reliability Engineering International*, 18, 479-488.

Kao S. (2010), "Normalization of the Origin-shifted Exponential Distribution for Control Chart Construction," *Journal of Applied Statistics*, 27, 1067-1087.

Kittlitz R.G., (1999), "Transforming the Exponential for SPC Applications," *Journal of Quality Technology*, 31, 301–308.

Liu J. Y., Xie M., Goh T.N., and Sharma P. R. (2006), "A Comparative Study of Exponential Time between Events Charts,", *Quality Technology and Quantitative Management*, 3, 347-359.

Lucas J. M. (1985), "Counted Data CUSUM's," *Technometrics*, 27, 129–144.

Montgomery, D.C. (2009), *Introduction to Statistical Quality Control* (6th ed.), New York: John Wiley.

Nelson L. S. (1994), "A Control Chart for Parts-per-million Nonconforming Items," *Journal of Quality Technology*, 26, 239–240.

Ozsan G, Testik M.C., and Weiß C.H (2010), "Properties of the exponential EWMA Chart with

parameter estimation," *Quality and Reliability Engineering International,* 26, 555-569.

Pehlivan C., and Testik M.C. (2010), "Impact of model misspecification on the exponential EWMA Charts: A robustness Study when the time-between-events are not exponential," *Quality and Reliability Engineering International*, 26, 177-190.

Vardeman S., and Ray D. (1985), "Average Run Lengths for CUSUM Schemes When Observations Are Exponentially Distributed," *Technometrics*, 27, 145–150.

CHAPTER 5

A MODIFIED ADJUSTED BOXPLOT FOR SKEWED DISTRIBUTIONS

Abstract

The boxplot (Tukey 1977) is a popular exploratory data analysis tool for displaying and summarizing distributions. It is also used for outlier detection and works well when the underlying distribution is symmetric. However, when the distribution is skewed, the boxplot typically yields increased false alarms, which diminishes its practical utility. Hubert and Vandervieren (2008, hereafter HV) considered an adaptation of Tukey's boxplot, called the adjusted boxplot, for skewed distributions. In this article, a modification of their boxplot is considered, called the modified adjusted boxplot. The proposed boxplot has fences measured from the median and are constructed using some multiples of the upper and the lower semi-interquartile ranges ($SIQRs$), respectively. These multiples depend on a robust measure of skewness, called the medcouple (MC), which allows the whiskers to account for skewness. Like the HV procedure, the proposed procedure is distribution-free and hence can be used without any knowledge about the form of the underlying distribution. Examples are provided using real data illustrating the proposed boxplot and contrasting with Tukey's and HV's boxplots. The performance of the proposed boxplot is investigated in a simulation study. It is seen that in most of the situations investigated, the proposed boxplot has advantages in terms of the percentage of outliers detected. Conclusions and suggestions for future work are offered.

Keywords: Boxplot; Fences; Outlier detection; Semi-interquartile range ($SIQR$); Median; Medcouple (MC)

1. Introduction

Not all observations in a dataset can be guaranteed to have come from the process under consideration. When an observation (or a group of observations) stands apart from the rest of the data, it naturally deserves more scrutiny. As early as in 1969, Grubbs stated, "An outlying observation is one that appears to deviate markedly from the other members of the sample in which it occurs". The vast statistical literature on the subject of outliers (Barnett 1978; Barnett and Lewis 1994; Brant 1990; Hawkins 1980; Hadi 2009; Rosner 1983; Schwertman et al. 2004; Schwertman and de Silva 2007; Sim et al. 2005; Tukey 1977) attests to its relevance in data analysis. Incorrect treatment of outliers can profoundly distort the statistical analysis, which can lead to erroneous conclusions and potential losses. Tukey's boxplot (also called the box-and-whisker plot) is one of the most popular tools, which is often used in exploratory data analysis, among other purposes, for a tentative detection of outliers. Because this boxplot does not use the extreme observations in the computation of the fences, it does not suffer from the problem known as "masking" in which the presence of some outliers makes other outliers difficult to be detected. The lower and upper fences of Tukey's boxplot are given by $LF = Q_1 - 1.5IQR$ and $UF = Q_3 + 1.5IQR$ where Q_1 and Q_3 are the sample lower and upper quartiles and $IQR = Q_3 - Q_1$. For the normal distribution, the probability that an observation falls outside of these fences (called the "probability of exceedance") is 0.7%, 0.35% on each side (lower and upper). However, while not always explicitly stated, the boxplot is more appropriate when the underlying distribution is more or less symmetric. When the distribution is skewed, say to the right, many observations can exceed the upper fence naturally and would therefore be incorrectly

identified as outliers by the boxplot. To illustrate this point, Table 5.1 lists the upper exceedance probabilities for some distributions. It can be observed that the upper exceedance probability increases as the skewness of a distribution increases (MC becomes larger). More details will be given later (in section 2.1) about the medcouple MC, which was proposed by Brys et al. (2003). The medcouple values reported in Table 5.1 are computed in Mathcad, using either the functional form given in Brys et al. (2004) or from a large sample from the distribution and using the R package robustbase. It is seen from Table 5.1 that with Tukey's boxplot, in general, the upper probability of exceedance increases rapidly as the medcouple value increases, that is as the distribution becomes more skewed, the chance of incorrect labelling of observations increases rapidly.

Table 5.1: The 20 different distributions used in the simulation study. MC is the medcouple, and Prob. Stands for probability of upper exceedance

No.	Distribution	MC	Prob.	No.	Distribution	MC	Prob.
1	$N(0,1)$	0	0.0035	11	$\Gamma(3,0.1)$	0.178	0.0253
2	$G^1{}_{0.05}$	0.022	0.0053	12	$\Gamma(5.5,0.1)$	0.129	0.0184
3	$G_{0.2}$	0.091	0.0148	13	$Pareto(1,2)$	0.468	0.0936
4	$G_{0.75}$	0.311	0.0693	14	$Pareto(1,5)$	0.395	0.0684
5	G_1	0.401	0.0947	15	$Pareto(1,8)$	0.368	0.0611
6	χ^2_2	0.333	0.0481	16	$F(80,10)$	0.259	0.0516
7	χ^2_{10}	0.136	0.0193	17	$F(20,20)$	0.196	0.0355
8	χ^2_{25}	0.085	0.0125	18	$F(20,80)$	0.125	0.0198
9	$\Gamma(1,0.1)$	0.336	0.0481	19	$F(50,50)$	0.124	0.0209
10	$\Gamma(1.5,0.1)$	0.264	0.0375	20	$F(75,75)$	0.102	0.0168

In order to increase the sensitivity of the boxplot to the skewness of the underlying distribution, researchers have suggested various adjustments to Tukey's boxplot. Kimber (1990) proposed using a boxplot with fences given by $LF = Q_1 - 3SIQR_L$, $UF = Q_3 + 3SIQR_U$, where $SIQR_L = Q_2 - Q_1$ and $SIQR_U = Q_3 - Q_2$ are the lower and the upper semi-interquartile

[1] More information about the *Gg* distribution later

ranges ($SIQRs$) respectively. The *SIQR* based fences are expected to work better with skewed data. Carling (2000) compared boxplots with upper fences of the form $c_1^U = Q_3 + k_1(Q_3 - Q_1)$ and $c_2^U = Q_2 + k_2(Q_3 - Q_1)$, respectively, and concluded that c_2^U leads to better performance in the non-Gaussian case with respect to "resistance" and "efficiency" (see Carling 2000, for more details). He pointed out that the right skewness increases the variation of the third quartile more than that for the median and this gives motivation to construct fences from the median since the median is known to be more robust. In fact, Schwertman et al. (2004) proposed a boxplot with fences from the median: $LF = Q_2 - k_1 SIQR_l,\ UF = Q_2 + k_1 SIQR_u,$ for the normal and near normal distributions, in which they controlled the outside rate per observation α_n (Hoaglin et al. 1986) to find the fence constant k_1. The quantity α_n is the probability that a single observation from a sample of size n is falsely classified as an outlier. However, it seems more meaningful to control the so-called some-outside rate per sample, α (Hoaglin et al. 1986), that is, the probability that at least one observation from an uncontaminated sample is falsely classified as outlier. This is because we are often interested in knowing if a dataset is contaminated or not, meaning that we want to find if there is at least one outlier in the data. Note that Sim et al. (2005), and Dovoedo and Chakraborti (2009) also controlled this probability α.

When constructing boxplot fences that are more sensitive to skewness, it seems reasonable and desirable that the lower and upper fences (fence constants) be different and that they be functions of a robust measure of skewness. This would allow the fences to account and adjust for skewness. Along that line, HV studied boxplots with fences of form:

$$[Q_1 - h_l(MC)IQR, Q_3 + h_u(MC)IQR],$$

where $h_l(MC)$ and $h_u(MC)$ are some suitably determined functions of the medcouple. However, for reasons given by Carling (2000) as previously stated, and following Kimber

(1990), particularly for skewed distributions, we consider a boxplot with fences from the median and using distances between the median and the first and the third quartile, respectively, on either end, given by

$$[Q_2 - k_l(MC)SIQR_l,\ Q_2 + k_u(MC)SIQR_u],$$

where $k_l(MC)$ and $k_u(MC)$ are two suitably determined functions of the medcouple. Note that our boxplot fences generalize Schwertman et al. (2004)'s fences by (i) introducing two separate fence constants, one on the lower end and one on the upper end, respectively and (ii) making both fence constants depend on the medcouple. This boxplot, which is a natural competitor to HV's boxplot, is studied here. For illustration, the proposed boxplots with the fences are shown in Figure 5.1 along with Tukey's fences with $k = 1.5$.

[Figure 5.1 here; see Appendix]

The paper is structured as follows. In section 2, we present the proposed modified adjusted boxplot. In section 3, we apply the proposed boxplot, Tukey's boxplot and the HV boxplot, for illustration to some datasets. In order to study the performance of the boxplot fences as outlier detection rules, a simulation study is performed in section 4 to compare the three boxplots, and the results are discussed. We conclude in Section 5 with some directions for further research.

2. Construction of the Modified Adjusted Boxplot

2.1 The medcouple (MC): A Robust Measure of Skewness

Brys et al. (2003) introduced a measure of skewness that is robust to outliers. Their measure, called the medcouple (MC), is inspired by the concept of quartile skewness introduced by Bowley (1920) and Moors et al. (1996). Brys et al. (2003) compared the medcouple to the

quartile skewness and the octile skewness, which are measures of skewness in a family of skewness measures (Hinkley (1975)) of the form:

$$\frac{(Q_{1-p}-Q_{0.5})-(Q_{0.5}-Q_p)}{Q_{1-p}-Q_p}, \text{ with } 0 < p < 1.$$

The quartile skewness corresponds to $p = 0.25$ and the octile skewness corresponds to $p = 0.125$. Brys et al. (2003) showed that the medcouple is the overall winner among the skewness measures, in that it combines the strength of the octile skewness (sensitivity to detect skewness) and the strength of quartile skewness (robustness towards outliers).

Let $\mathrm{X_n} = \{\mathrm{x_1}, \ldots, \mathrm{x_n}\}$ be a random sample from a univariate continuous distribution. Let $\{\mathrm{x_{(1)}}, \ldots, \mathrm{x_{(n)}}\}$ be the sorted sample. The medcouple is defined by:

$$MC_n = \underset{x_{(i)} \le med_n \le x_{(j)}}{med} h\left(x_{(i)}, x_{(j)}\right),$$

where med_n is the sample median of the sample X_n. When $x_i \neq x_j$ the function h is defined by:

$$h\left(x_{(i)}, x_{(j)}\right) = \frac{\left(x_{(j)} - med_n\right) - \left(med_n - x_{(i)}\right)}{x_{(j)} - x_{(i)}}.$$

However when $x_{(i)} = x_{(j)} = med_n$, that is when there may be ties in the data, the function h is defined as follows. If $m_1 < m_2 < \cdots < m_k$ are the indices of observations tied to the median med_n $(i.e., x_{m_l} = med_n)$, for all $l = 1, 2, .., k$:

$$h\left(x_{(i)}, x_{(j)}\right) = \begin{cases} -1 & if\ i + j - 1 < k \\ 0 & if\ i + j - 1 = k \\ +1 & if\ i + j - 1 > k \end{cases}.$$

Brys et al. (2004) showed that (i) the medcouple takes values between -1 and 1 and can resist up to 25% outliers before breaking down and (ii) the medcouple takes negative, positive and zero values for left skewed, right skewed and symmetric distributions, respectively. They provided a

fast algorithm for computing the medcouple which is implemented in the package *robustbase* in R.

2.2 Modifying the Adjusted Boxplot

Recall that our boxplot has fences of form:

$[Q_2 - k_l(MC)SIQR_L, Q_2 + k_u(MC)SIQR_U]$. Following HV, we impose the constraint $k_l(0) = k_u(0) = 4$ so that the boxplot reduces to Tukey's boxplot for symmetric distributions. In order to determine the fence constants $h_l(MC)$ and $h_u(MC)$, following HV, we consider three models (linear, quadratic and exponential).

(1) Linear model:

$$\begin{cases} k_l(MC) = 4 + aMC \\ k_u(MC) = 4 + bMC \end{cases} \quad (1)$$

(2) Quadratic model:

$$\begin{cases} k_l(MC) = 4 + a_1MC + a_2MC^2 \\ k_u(MC) = 4 + b_1MC + b_2MC^2 \end{cases} \quad (2)$$

(3) Exponential model:

$$\begin{cases} k_l(MC) = 4e^{a_3MC} \\ k_u(MC) = 4e^{b_3MC} \end{cases} \quad (3)$$

with $a, a_1, a_{2,} a_{3,}\ b, b_1, b_{2,} b_{3,} \epsilon \mathbb{R}$.

2.3 Determination of Fence Constants

The fence constants depend on the *constant*s (parameters) $a, a_1, a_{2,} a_{3,}\ b, b_1, b_{2,} b_{3,} \epsilon \mathbb{R}$. These are found by fitting a range of distributions and finding those fences such that the probability of exceedance is close to 0.7%. Recall that 0.7% is the probability of exceedance with Tukey's boxplot for the normal distribution.

Thus, for the linear model, the parameters a and b satisfy:

$$\begin{cases} Q_2 - (4 + aMC)SIQR_L \approx \mathrm{Q}_\alpha \\ Q_2 + (4 + bMC)SIQR_U \approx \mathrm{Q}_\beta \end{cases},$$

where Q_α and Q_β denote, respectively, the $100\alpha^{th}$ and the $100\beta^{th}$ percentiles of the distribution under consideration. Note that the lower and the upper fences are equated to one lower and one upper percentile so as to cover the range of the distributions. We take $\alpha = 0.0035$ and $\beta = 0.9965$ so that the probability of exceedance is 0.7% as in HV but other choices are possible. The preceding system of equations is equivalent to

$$\begin{cases} \dfrac{(Q_2 - Q_\alpha)}{(Q_2 - Q_1)} - 4 \approx aMC \\ \dfrac{(-Q_2 + Q_\beta)}{(Q_3 - Q_2)} - 4 \approx bMC \end{cases}.$$

Now linear regression without an intercept can be used to find estimates of parameters a and b for any given distribution and given α and β.

Similarly, for the quadratic model, the parameters a_1, a_2, b_1 and b_2 can be found that satisfy:

$$\begin{cases} Q_2 - (4 + a_1MC + a_2MC^2)SIQR_L \approx Q_\alpha \\ Q_2 + (4 + b_1MC + b_2MC^2)SIQR_U \approx Q_\beta \end{cases},$$

which can be re-written as

$$\begin{cases} \dfrac{(Q_2 - Q_\alpha)}{(Q_2 - Q_1)} - 4 \approx a_1MC + a_2MC^2 \\ \dfrac{(-Q_2 + Q_\beta)}{(Q_3 - Q_2)} - 4 \approx b_1MC + b_2MC^2 \end{cases}.$$

Hence, in this case, quadratic regression without the intercept can be used to estimate the parameters.

In the exponential model the parameters a_3 and b_3 satisfy:

$$\begin{cases} Q_2 - 4e^{a_3 MC} SIQR_L \approx Q_\alpha \\ Q_2 + 4e^{b_3 MC} SIQR_U \approx Q_\beta \end{cases},$$

which can be re-written as

$$\begin{cases} ln\left(\frac{1}{4}\frac{Q_2 - Q_\alpha}{Q_2 - Q_1}\right) \approx \mathrm{a}_3 MC \\ ln\left(\frac{1}{4}\frac{Q_\beta - Q_2}{Q_3 - Q_2}\right) \approx \mathrm{b}_3 MC \end{cases}.$$

Hence linear regression on the log scale, without the intercept, can be used to find the estimates of the parameters.

The parameters for each model were estimated by using a total of 10,302 distributions from the family of gamma (Γ), chi-square (χ^2), F, Pareto and G_g-distribution (Hoaglin et al. 1985) covering a wide range of distributions of different shapes, locations and scales. Note that as in HV, we do not include distributions with medcouple values greater than 0.6 as it is difficult to construct a good model incorporating both symmetric distributions and very highly skewed distributions. The family of G_g-distributions is defined as follow. For a non-zero real number g, If Z follows the standard normal distribution, then $(\exp(Zg) - 1)/g$ follows the G_g-distribution. The specific distributions used in the study are shown in Table 5.2 and include G_g-distributions. Note that the ranges for the parameters of these distributions were chosen to be the same as those used by HV. In order to estimate the parameters of each of the three models, the $100\alpha^{th}$ and the $100\beta^{th}$ percentiles as well as the medcouple value of each the distribution was needed. For convenience, these were calculated from a large sample (10,000 observations) from the respective distributions and are taken to serve as the actual values.

Table 5.2: Distributions used to fit the models

Chi-Square	G_g	*Gamma*		*Pareto*		*F*	
df	g	*Shape*	*Scale*	*Location*	*Shape*	df_1	df_2
[1,30]	[0,1]	[0.1,10]	0.1	1	[0.1,20]	[1,100]	[1,100]

2.4 The Modified Adjusted Boxplot

Regression analyses (outputs not shown here due to lack of space but are available from the authors) show that in general, much more accurate upper fences are found when using the exponential model. For the lower fences, the exponential model gives more accurate estimates than the linear model followed by the quadratic model. Thus we adopt the exponential model with $a_3 = -1.95$ and $b_3 = 2.18$ in the our modified adjusted boxplot. To make the fitted model simpler, the estimated parameters are rounded down to the nearest integers, $a_3 = -2$ and $b_3 = 2$. Because we round-down, this results in smaller fences and therefore, in a more robust boxplot.

To summarize, according to the proposed fences, all observations that fall outside the interval

$$[Q_2 - 4e^{-2MC}SIQR_L, Q_2 + 4e^{2MC}SIQR_U],$$

are labeled as potential outliers.

3. Illustrations

We now illustrate the proposed procedure, Tukey's procedure and HV's procedure by applying them to some actual datasets. The computed fences are reported in Table 5.3.

Table 5.3: Various boxplot procedure fences for various data sets

The coal mine data	
Tukey's Fences $k = 1.5$	$LF = Q_1 - 1.5(Q_3 - Q_1) = -320; UF = Q_3 + 1.5(Q_3 - Q_1) = 632$
HV's Fences	$LF = Q_1 - 1.5e^{-4MC}IQR = -35.567\,; UF = Q_3 + 1.5e^{3MC}IQR = 1454.27$
Proposed Fences	$LF = Q_2 - 4e^{-2MC}SIQR_L = -24.46\,; UF = Q_2 + 4e^{2MC}SIQR_U = 1546.33$
The Minnesota land rent data	
Tukey's Fences $k = 1.5$	$LF = Q_1 - 1.5(Q_3 - Q_1) = -0.19;\ UF = Q_3 + 1.5(Q_3 - Q_1) = 0.49$
HV's Fences	$LF = Q_3 - 1.5e^{-4MC}IQR = 0.0017;\ UF = Q_3 + 1.5e^{3MC}IQR = 0.96$
Proposed Fences	$LF = Q_2 - 4e^{-2MC}SIQR_L = 0.0104;\ UF = Q_2 + 4e^{2MC}SIQR_U = 1.044$
The Crohn's disease adverse events data	
Tukey's Fences $k = 1.5$	$LF = Q_1 - 1.5(Q_3 - Q_1) = 25.75;\ UF = Q_3 + 1.5(Q_3 - Q_1) = 83.75$
HV's Fences	$LF = Q_3 - 1.5e^{-3MC}IQR = 20.106;\ UF = Q_3 + 1.5e^{4MC}IQR = 77.99$
Proposed Fences	$LF = Q_2 - 4e^{-2MC}SIQR_L = 16.347;\ UF = Q_2 + 4e^{2MC}SIQR_U = 76.579$

3.1 The Coal Mine Data

The data are the time intervals between the coal mine disasters (Jarret, 1979) measured in days between March 15th, 1851 and March 22nd, 1962, inclusive. The histogram of the data (not shown here due to space) suggests that the distribution is right-skewed. Note that the dataset

contains one zero which was removed so that the gamma goodness of fit test could be performed. The Anderson-Darling (AD) statistic reported by Minitab is $AD = 1.091$ with a $p\text{-}value = 0.01$. Hence, the data fail the gamma distribution goodness of fit test at significance level $\alpha = 0.05$. However, the data with the three largest observations removed do not fail the gamma distribution goodness of fit test. The AD test statistic reported by Minitab is 0.408 with p-value > 0.25.

The sample size is $N = 190$ and the median is: $Q_2 = \frac{X_{(95)}+X_{(96)}}{2} = \frac{113+114}{2} = 113.5$. The medcouple value is computed to be $MC_{190} = 0.398$ (in R, using the package "robustbase"). For the quartiles we use the standard fourths or hinges: $Q_1 = X_{(48)} = 37$ and $Q_3 = X_{(143)} = 275$. The fences for the three boxplots are calculated and shown in Table 5.3. It found that with Tukey's boxplot, the twelve larger observations are flagged as outliers. It appears that too many observations are declared as outliers and one reason could be that Tukey's boxplot does not take account of skewness. By contrast, both the HV and the proposed boxplot only flag the three largest observations as outliers, which seems more reasonable in the light of the histogram of the data.

3.2 The Minnesota Land rent Data

As a second illustration, we consider the Minnesota agricultural data from Weisberg (1985). The dataset contains $n = 67$ observations corresponding to the 67 counties in Minnesota. The objective of the study was to investigate the rent structure of Minnesota agricultural land with emphasis on alfalfa hay and several variables were measured. We examine the variable which is the ratio of total pasture acres to the total cropland acres for which the normal distribution does not seem appropriate as the AD statistic is 3.128 with $p\text{-}value < 0.005$. Subsequently, using the

STATFIT feature of the software package ProModel, we find that up to 10 different distributions fit these data, among which are lognormal, gamma, beta, and Weibull distributions. The beta distribution appears to be the most appropriate; as the corresponding AD statistic is 0.527 with $p\text{-}value = 0.719$. The parameters of the fitted beta distribution are $\alpha = 1.694$ and $\beta = 724.616$. The medcouple computed in *R* is 0.34848 and the three quartiles are: $Q_1 = \frac{X_{(17)}+X_{(18)}}{2} = 0.065$, $Q_2 = X_{(34)} = 0.12$, and $Q_3 = \frac{X_{(50)}+X_{(51)}}{2} = 0.235$. The fences for the three boxplots are calculated and shown in Table 5.3. It is found that Tukey's boxplot declares the three largest observations (0.56, 0.66, and 0.72) as outliers. Note that even after removing these observations, the remaining data fail the normality test; the corresponding AD statistic is 1.696 with $p\text{-}value < 0.005$. On the other hand, both the proposed and HV's boxplot procedures declare no outliers which seems more consistent and reasonable in light of the histogram of the data (not displayed here).

3.3 Crohn's Disease Adverse Events Data

The underlying distributions in the preceding illustrations appear to be skewed. It is of interest to consider a situation where the distribution is more or less symmetric, that is, the *MC* is close to zero. This dataset is one of the R software datasets and is named the CrohnD data. The variable of interest here is Age. The medcouple computed in *R* equals -0.0769 and the three quartiles are: $Q_1 = \frac{X_{(29)}+X_{(30)}}{2} = 47.5$, $Q_2 = X_{(59)} = 56$, and $Q_3 = \frac{X_{(88)}+X_{(89)}}{2} = 62$. The fences are computed and shown in Table 5.3. It is seen that both Tukey's and HV's boxplots declare the smallest observation (19) as an outlier while the proposed procedure declares none. Note that for these data with all $n = 117$ observations, the AD statistic was 0.736 with a *P*-value$= 0.054$, which does not reject the normality assumption at a significance level of 0.05. However, if the

smallest observation is excluded, as per our boxplot, the AD statistic decreases to 0.575, which leads to a higher P-value, 0.132, which supports the normality assumption more strongly. In summary it appears that deleting the smallest data point leads to a data set that conforms better to the normality assumption and hence the proposed boxplot may be more effective in this case.

Next, we compare the performance of the three boxplot procedures in a simulation study.

4. Performance Comparisons

The distributions involved in the simulation included the normal distribution and many distributions like the G_g, χ^2, Γ, Pareto and F-distributions (see Table 5.1). The distributions are chosen to have various degrees of skewness. Throughout the study, 10,000 simulations were used so that the 95% margin of error is at most 0.07%.

4.1 Performance with Uncontaminated Data

For each distribution, 10,000 random samples of size 1000 were generated and the percentages of lower and upper outliers (the percentage of observations that fall outside the fences) were calculated for each procedure. Notice that large sample sizes are used because we want to have an idea of the *expected* false alarm rate, as in HV. Also, the medcouple value obtained from a large sample represents the population medcouple more accurately. The average (over the 10,000 simulations) percentages of falsely detected outliers for Tukey's boxplot (circles), HV's boxplot (crosses) and the proposed boxplot (triangles) are reported in Figure 5.2.

[Figure 5.2 here; see appendix]

First, we compare the proposed boxplot procedure with Tukey's boxplot. The difference between the proposed boxplot and the standard boxplot is small for symmetric or almost

symmetric distributions. For example, for the normal distribution (distribution 1) the percentage of observations falsely declared as outliers is 0.725% for Tukey's boxplot while it is 0.8946% for the proposed boxplot. However, more pronounced differences show up when the underlying distributions are skewed. For example, for the $Pareto(location = 1, shape = 5)$ distribution (distribution 14), the percentage of observations declared as outliers is 6.844% for Tukey's boxplot while it is only 1.256% for the proposed boxplot. Overall, the percentage of observations falsely declared as outliers is lower for the proposed boxplot in 18 out of the 20 distributions studied.

Next, we compare the average percentages of observations falsely declared as outliers for the HV and the proposed boxplots. Figure 5.2 shows that in most cases (14 out of 20), the percentage of regular observations declared as outliers is slightly less for the proposed procedure. This includes the following distributions: $Chi\text{-}square(10)$, $Chi\text{-}square(25)$, $Gamma(3,0.1)$ and all F-distributions studied. However, there are a couple of cases for which HV's boxplot falsely declared markedly less observations as outliers than the proposed boxplot. These are the G_1 (distribution 5) and $Pareto(location = 1, shape = 2)$ (distribution 13). We observe that when the medcouple of the distribution is relatively high, say higher than 0.38, the proposed boxplot tends to declare a higher percentage of regular observations as outliers than HV's boxplot.

To summarize, for a majority of the distributions under study, the percentages of observations falsely declared as outliers corresponding to the proposed boxplot are smaller than those for both Tukey's and HV's boxplots.

Next, we compare the performance of the three boxplot procedures with contaminated data.

4.2 Performance with Contaminated Data

To compare the boxplots in more detail, we study their performances under contaminated data, that is, data with outliers. We generate 10,000 samples of size 1000 from each distribution in Table 5.1. We then randomly replace 5% of these 1000 observations (a total of 50 observations) with 50 "extreme" observations from a normal distribution with appropriately chosen mean and standard deviation. Here "extreme" refers to an observation that falls beyond the fences of all three procedures simultaneously. The final data set is referred to as the 5% right contaminated data. Similarly, we simulated the 5% left contaminated data set. Next, we calculate, for each procedure, the proportion of observations (out of 1,000) it declares as outliers and average these proportions over the 10,000 simulations. Finally, we calculate the deviations of the average proportions from the true contamination level, 5%.

4.2.1 5% Left (Lower) Contamination

The absolute deviations between the average empirical percentages of outliers detected by each procedure and the true contamination level (5%) (that is the true percentage of outliers) are shown in Figures 5.3 and 5.4. Note that smaller absolute deviations are better. The objective is to detect the true 5% outliers in the lower tail without too many false alarms.

Performance in the lower tail:

[Figure 5.3 here; see appendix]

From Figure 5.3 we see that all three boxplots perform comparably on most distributions and no method is always superior. There are cases where Tukey's boxplot does not perform as well as the other two boxplots. For example, for the $Chi\text{-}square(10)$ (distribution 7), the absolute deviation from the true 5% lower outliers is 1.104% for Tukey's boxplot, while it is

only 0.03% and 0.022% respectively for the proposed boxplot and HV's. However, there are a couple of cases (for symmetric or almost symmetric distributions) where Tukey's boxplot performs better than both the proposed boxplot and HV's. For example, at the $G_{0.05}$ (distribution 2) that has medcouple $MC_n = 0.01$, the absolute deviation from the true 5% is 0.166% for Tukey's boxplot, but 0.569% and 0.692%, respectively, for the proposed boxplot and HV's boxplot.

Performance in both tails:

[Figure 5.4 here; see appendix]

Figure 5.4 shows the absolute deviations between the average empirical percentages of outliers detected and the nominal level of contamination (5%) in both tails. First, we compare the proposed procedure with Tukey's. It is clearly seen that while for the proposed procedure the percentage of outliers detected in both tails is markedly closer to the true 5% that is not the case for Tukey's procedure. For example, at the $Gamma(shape = 1, scale = 0.1)$ (distribution 9), the absolute deviation for Tukey's boxplot is 4.833% while it is only 0.407% for the proposed boxplot. We observe that for all distributions studied (20 out of 20) the absolute deviations are much smaller for the proposed procedure.

Next, we compare HV's procedure with ours. It is again seen from Figure 5.4 that the absolute deviations are slightly smaller for the proposed boxplot for 19 out of the 20 distributions studied. For example, for the $F(80,10)$ distribution (distribution 16), the absolute deviation is 1.581% for the proposed procedure while it is 1.822% for HV's procedure. However, the $G_{0.05}$ distribution (distribution 2) is the only distribution for which the absolute deviation is smaller for HV's procedure (0.067%) than for the proposed procedure (0.236%).

4.2.2 5% Right (Upper) Contamination

Performance in upper tail:

The absolute deviations from the contamination level (5%) of the average percentages of observations declared as upper outliers are presented in Figure 5.5. Again, the objective is to detect the true 5% outliers in the upper tail without too many false alarms in the lower tail.

[Figure 5.5 here; see appendix]

It is seen from Figure 5.5 that the absolute deviations are higher for Tukey's boxplot at most distributions while they are comparable for the proposed boxplot and HV's. For example, At the $Gamma(shape = 3, scale = 0.1)$ (distribution 10), the absolute deviation from the true 5% upper outliers is 2.336% for Tukey's boxplot, while it is only 0.109% and 0.049% respectively for the proposed boxplot and HV's. However, there are few cases where Tukey's boxplot yields smaller absolute deviations than both the proposed boxplot and HV's. For example, for the $N(0,1)$ distribution (distribution 1), the absolute deviation is 0.209% for Tukey's boxplot but 0.846% and 0.811% for the proposed boxplot and HV's boxplot, respectively. Note that in all cases studied, the absolute deviation of the average percent outliers detected in the upper tail (from the true 5% outliers) is lower for the proposed procedure than for Tukey's and those for HV's procedure are slightly lower than the one proposed. However, as it is seen later, when both tails are considered, the absolute deviations of the average percentages of outliers detected are lower for the proposed procedure than for HV procedure for about half the distributions studied.

Performance in both tails:

[Figure 5.6 here; see appendix]

The absolute deviations between the average empirical percentages of outliers detected by each procedure in both tails and the true contamination level (5%) are shown in Figure 5.6. First,

we compare the proposed procedure to Tukey's procedure. It is seen that, in 17 out of the 20 cases studied, the average percentage of outliers detected in both tails by the proposed procedure is markedly closer to the true 5% than that of Tukey's procedure. For example, at the $F(20,20)$ (distribution 17), the absolute deviation for Tukey's boxplot is 2.333% while it is only 0.597% for the proposed boxplot.

Next, we compare HV's procedure to ours. It is seen that there is no clear winner. The absolute deviations are slightly smaller for the proposed boxplot than for HV's for half (10 out of 20) of the distributions studied. For example, at the $Chi\text{-}square(10)$ (distribution 7) the absolute deviation is 1.749% for the proposed procedure while it is 1.216% for HV's procedure. At the $F(75,75)$ distribution (distribution 20) the absolute deviation, which is smaller for the proposed procedure, is 0.892% while it is 1.228% for HV procedure.

We observe that HV procedure performs better for the G_1 distribution (distribution 5) and the $Pareto(location = 1, shape = 2)$ (distribution 13). Those two distributions have medcouple values 0.389 and 0.482, respectively. As pointed out previously, the percentage of regular observations declared as outliers (in the lower tail) tend to be high for highly skewed distributions (medcouple values larger than 0.38).

We now summarize our findings from the simulation study. The upper panel in Table 5.4 shows the comparison between the proposed procedure and Tukey's procedure and the lower panel in Table 5.4 shows the comparison between the proposed procedure and HV's procedure.

Table 5.4: Comparisons among various boxplot procedures

	Uncontaminated data	(5%) Left contaminated data	(5%) Right contaminated data
Comparison between the proposed and Tukey's boxplot procedures			
	Number of distributions for which the		
Procedure	false alarm rate is smaller (better)	absolute deviation* is smaller (better)	
Proposed Procedure	18	20	17
Tukey's Procedure	2	0	3
Comparison between the proposed and HV's boxplot procedures			
	Number of distributions for which the		
	false alarm rate is smaller (better)	absolute deviation* is smaller (better)	
Proposed Procedure	14	19	10
HV Procedure	6	1	10

*of the average percentages of observations declared as outliers from the true contamination level (5%)

It is seen that the proposed procedure clearly performs better than Tukey's procedure for both uncontaminated and contaminated data. It is also seen that the proposed procedure performs better than HV's for uncontaminated and left contaminated data, but for right contaminated data, the performances of the two procedures are similar.

5. Conclusions and Further Work

A modification of HV's adjusted boxplot is proposed by considering fences measured from the median and combining the use of the medcouple and the semi-interquartile ranges to take account of skewness in the underlying distribution. The procedure is distribution-free and hence can be applied without a knowledge of the underlying distribution. Three illustrations with real data are provided in three different applications. Results from the simulation study indicate a consistent advantage of the proposed procedure over Tukey's procedure in terms of (i) the

average percentage of outliers detected in case of contaminated data (data with outliers) and (ii) the false alarm rate in case of uncontaminated data. These results also show, in general, the same advantages of the proposed procedure over HV's, particularly for moderately skewed distributions. An extension of this work will be to examine the effect of the degree of contamination on the simulation results. Another extension will be to use the proposed procedure to detect outlying observations in multivariate skewed data along the line of Hubert and Van der Veeken (2008). These problems will be considered elsewhere.

APPENDIX

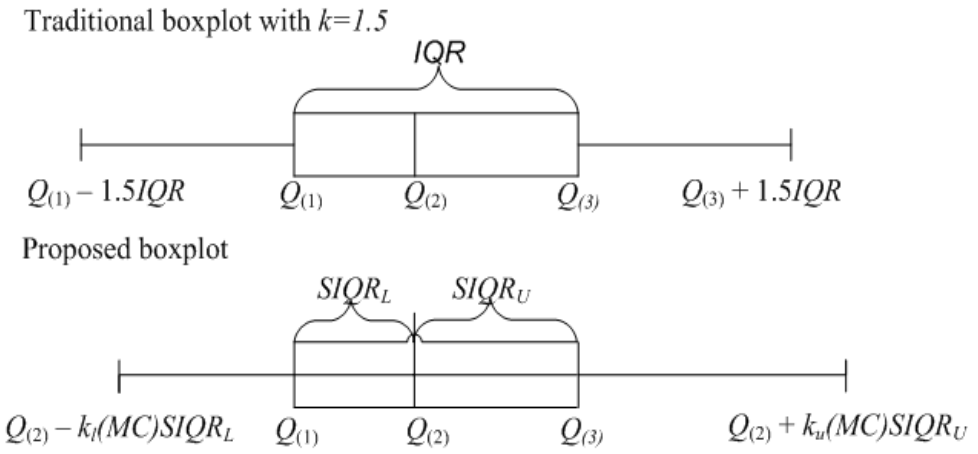


Figure 5.1: A graphical display of Tukey's traditional boxplot and the proposed boxplot

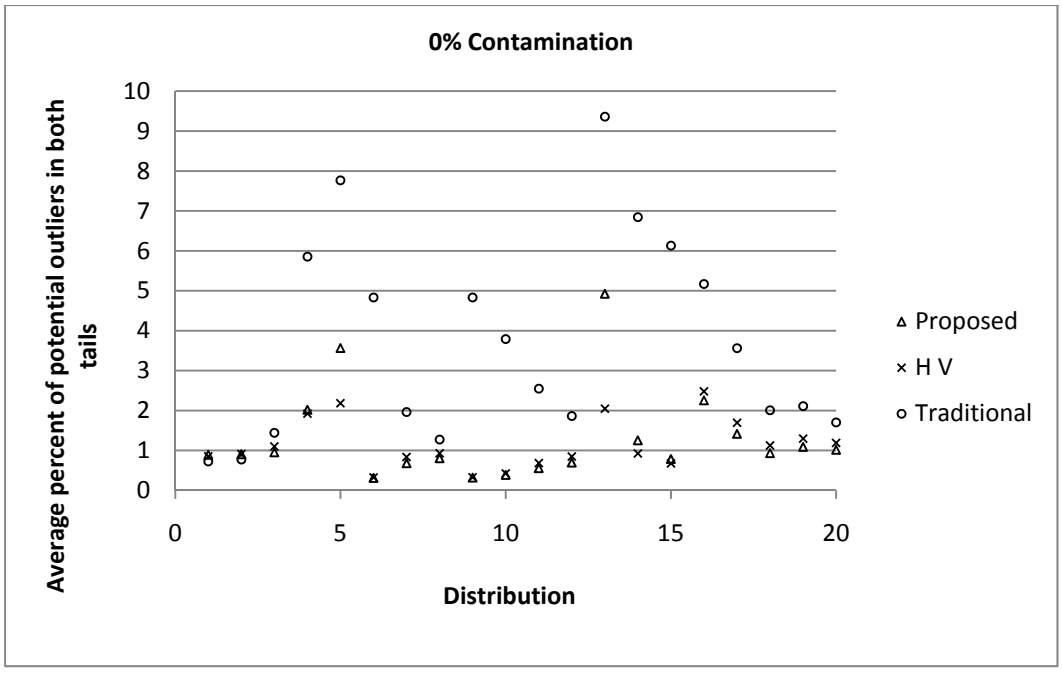


Figure 5.2: Comparison of average percentages of outliers declared, in both tails combined. Results for Tukey's boxplot (circles), the adjusted boxplot (crosses) and the proposed boxplot (triangles)

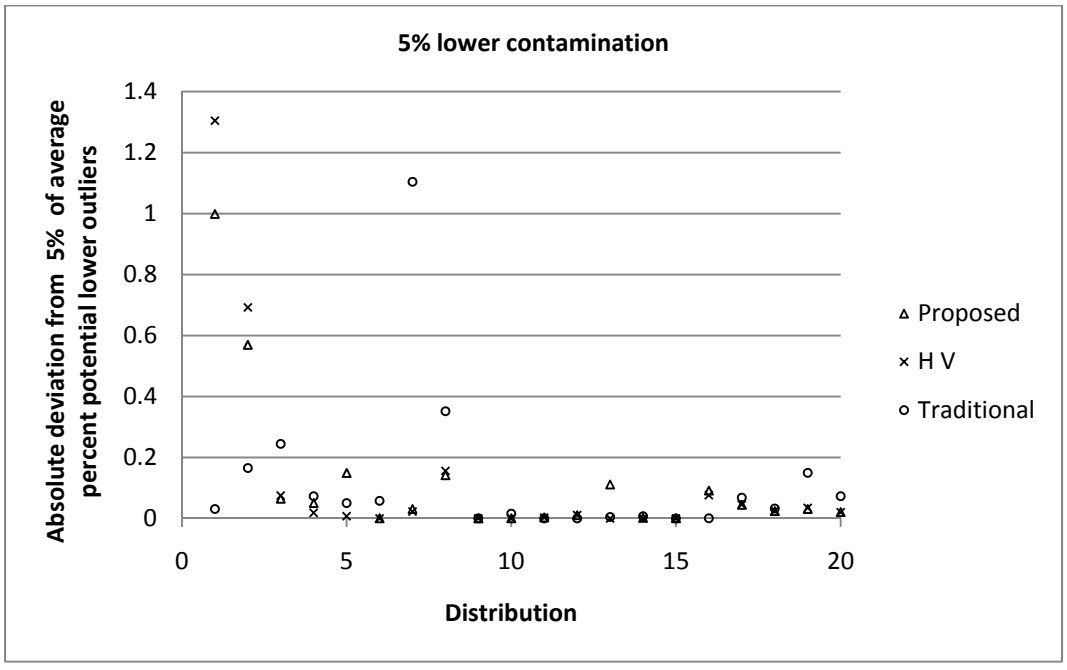


Figure 5.3: Absolute deviation of the average percentages of outliers declared in the lower tail, from the true percentage of outliers (5%). Results for Tukey's boxplot (circles), the adjusted boxplot (crosses) and the proposed boxplot (triangles)

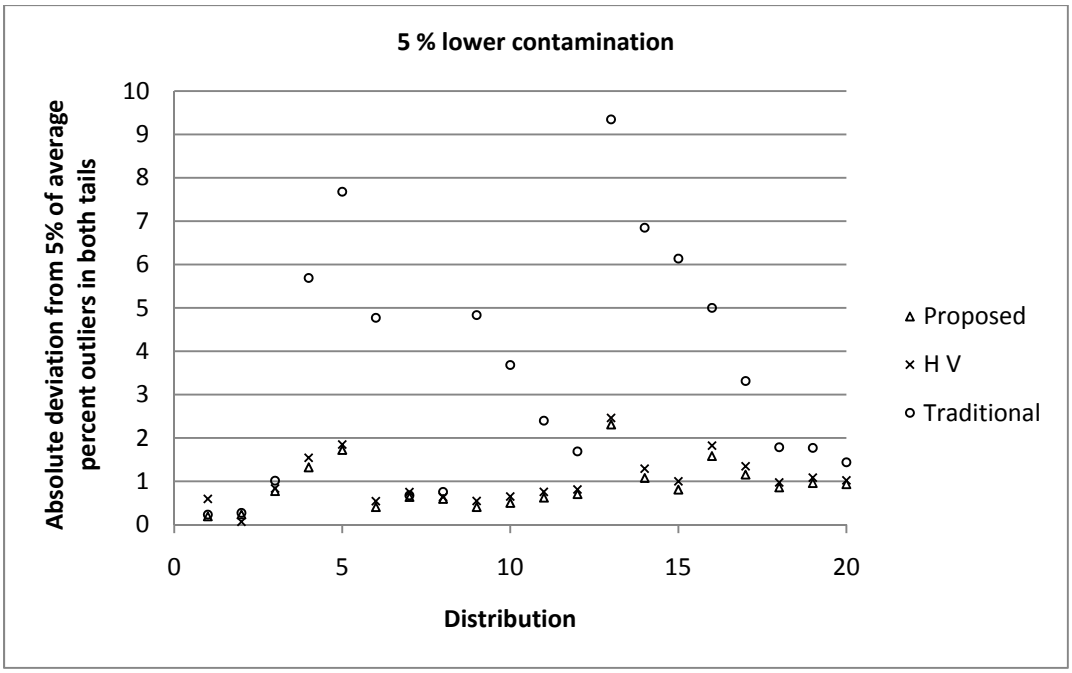


Figure 5.4: Absolute deviation of the average percentages of outliers declared, in the two tails combined, from the true percentage of outliers (5%). Results for Tukey's boxplot (circles), the adjusted boxplot (crosses) and the proposed boxplot (triangles)

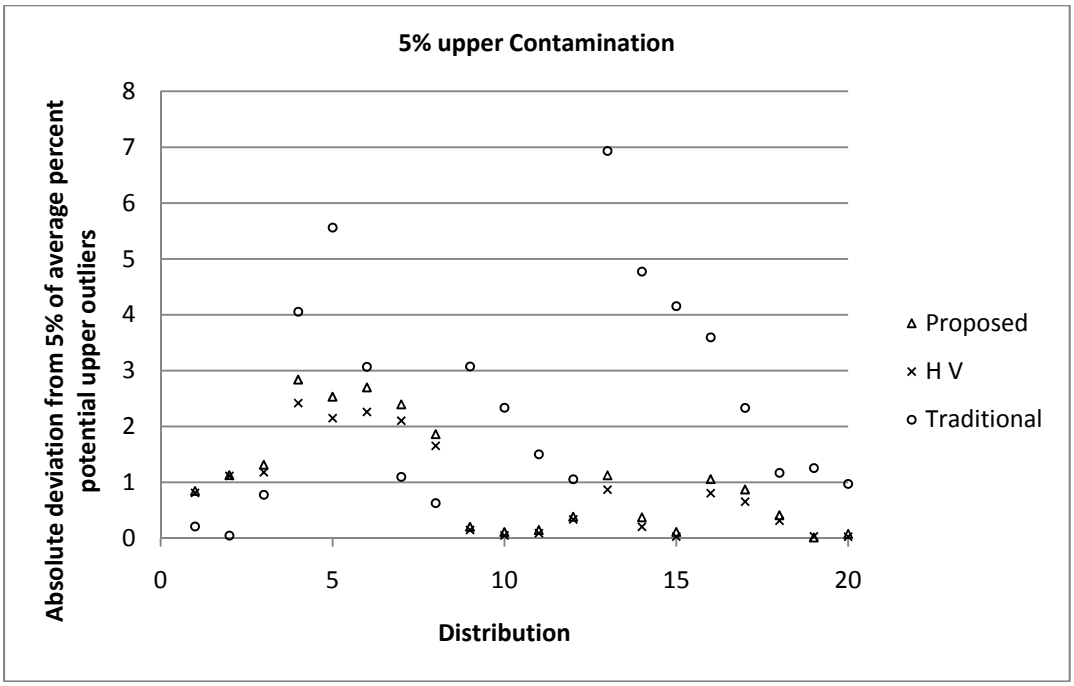


Figure 5.5: Absolute deviations of the average percentages of outliers declared, (in the two tails combined, from the true percentage of outliers (5%). Results for Tukey's boxplot (circles), the adjusted boxplot (crosses) and the proposed boxplot (triangles)

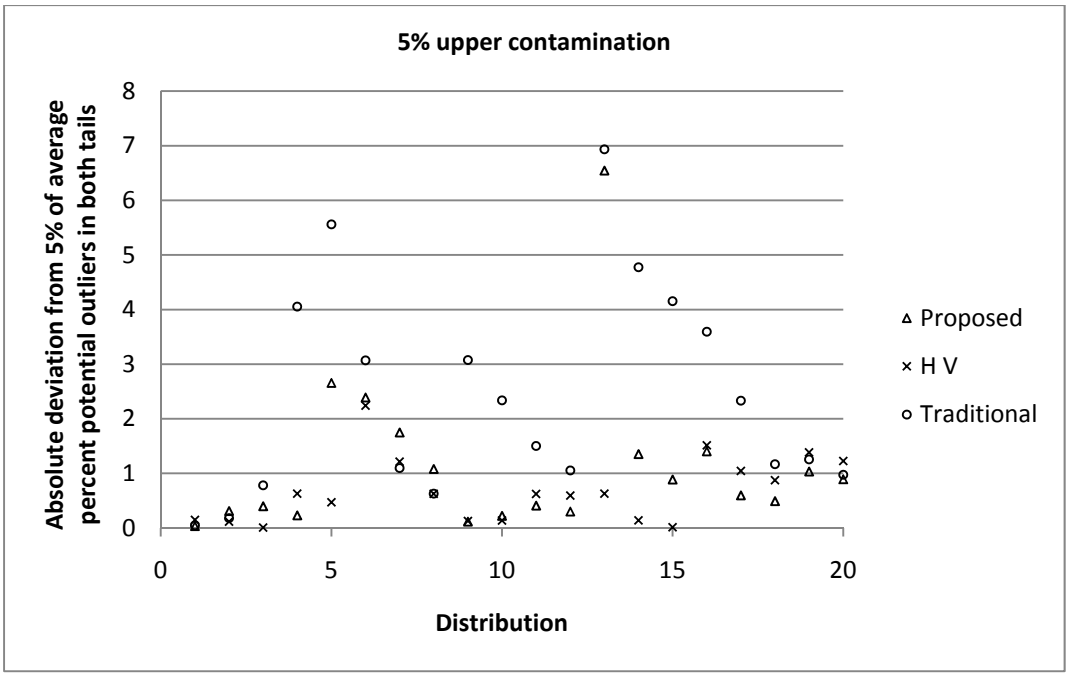


Figure 5.6: Absolute deviations of the average percentages of outliers declared, in the two tails combined, from the true percentage of outliers (5%). Results for Tukey's boxplot (circles), the adjusted boxplot (crosses) and the proposed boxplot (triangles)

REFERENCES

Barnett, V. (1978), "The Study of Outliers: Purpose and Model," *Journal of Applied Statistics*, 27, 242-250.

Barnett, V., and Lewis, T. (1994), *Outliers in Statistical Data* (3rd ed.), Chichester: John Wiley.

Bowley, A.L. (1920). *Elements of Statistics*, New York: Charles Scribner's Sons

Brant, R. (1990), "Comparing Classical and Resistant Outlier Rules," *Journal of the American Statistical Association*, 85, 1083-1090. .

Brys, G., Hubert, M., and Struyf, A. (2003), "A Comparison of Some New Measures of Skewness," in: *Developments in Robust Statistics (ICORS 2001),* eds. Dutter, R., Filzmoser, P., Gather, U., and Rousseeuw P. J., 98-113, Heidelberg: Springer-Verlag.

Brys, G., Hubert M., and Struyf A. (2004), "A Robust Measure of Skewness," *Journal of Computational and Graphical Statistics*, 13, 996-1017.

Carling K. (2000), "Resistant Outlier Rules and the Non-Gaussian Case," *Computational Statistics and Data Analysis*, 33, 249-258.

Dovoedo, Y. H. and Chakraborti, S. (2009), "On some properties of a simple and more general boxplot-type method for identifying outliers," *American Statistical Association Proceedings of the statistical graphics section* pp. 3900-3908.

Grubbs F.E. (1969), "Procedures for Detecting Outlying Observations in Samples," *Technometrics*, 11, 1-21.

Hadi, A. S., Rahmatullah Imon, A. H. M., and Werner, M. (2009), "Detection of Outliers," Wires *Computational Statistics*, 1, 57-70.

Hawkins, D. M. (1980), *Identification of Outliers*, London: Chapman & Hall.

Hinkley, D. V. (1975), "On Power Transformations to Symmetry," *Biometrika*, 62, 101-111.

Hoaglin, D. C., Iglewicz, B., and Tukey, J. W. (1986), "Performance of Some Resistant Rules for Outlier Labeling," *Journal American Statistician Association*, 81, 991-999.

Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (1985), *Exploring Data Tables, Trends and Shapes*, New York: Wiley.

Hubert, M., and Van der veeken, S. (2008), "Outlier Detection for Skewed Data," *Chemometrics*, 22, 235-246.

Hubert, M., Vandervieren, E. (2008), "An Adjusted Boxplot for Skewed Distributions," *Computational Statistical Data Analysis*, 52, 5186-5201.

Jaret, R.G. (1979), "A Note on the Intervals Between Coal Mining Disasters," *Biometrika*, 66, 191-193.

Kimber, A.C. (1990), "Exploratory Data Analysis for Possibly Censored Data from Skewed Distributions," *Journal of Applied Statistics*, 39, 21-30.

Moors, J. J. A., Wagemakers, R. Th. A., Coenen, V. M. J., Heuts, R. M. J., and Janssens, M. J. B. T. (1996), "Characterizing Systems of Distributions by Quantiles Measures," *Statistica Neerlandica*, 50, 417-430.

Oja, H. (1981), "On Location, Scale, Skewness and Kurtosis of Univariate Distributions," *Scandinavian Journal of Statistics*, 8, 154-168.

Rosner, B. (1983), "Percentage Points for the Generalized ESD Many-outlier Procedure," *Technometrics*, 25, 165-172.

Schwertman, N. C., Owens, and M. A., Adnan, R. (2004), "A Simple More General Boxplot Method for Identifying Outliers," *Computational Statistics and Data Analysis*, 47, 165-174

Schwertman, N. C., and de Silva, R. (2007), "Identifying Outliers With Sequential Fences," *Computational Statistics and Data Analysis*, 51, 165-174.

Sim, C. H., Gan, F. F., and Chang, T. C. (2005), "Outlier Labeling With Boxplot Procedures," *Journal of the American Statistical Association*, 100, 642-652.

Tukey, J.W. (1977), *Exploratory Data Analysis*, New York: Addison-Wesley.

Van Zwet, W.R. (1964), "Convex Transformations of Random Variables," *Mathematisch Centrum*, Amsterdam.

Weisberg, S. (1985), *Applied Linear Regression*, New York: John Wiley.

CHAPTER 6

OUTLIER DETECTION FOR MULTIVARIATE SKEW-NORMAL DATA: A COMPARATIVE STUDY

Abstract

Outlier detection in the multivariate setting is a challenging problem. A general way of detecting multivariate outliers consists of using robust depth functions, or, equivalently, the corresponding "outlyingness" functions; the more outlying is an observation, the more extreme (less deep) it is in the data cloud and is potentially an outlier. Most outlier detection studies in the literature assume that the underlying distribution is multivariate normal. This paper deals with the case of multivariate skewed data, specifically when the data follow the multivariate skew-normal (Azzalini and Dalla Valle 1996) distribution. We compare the outlier detection capabilities of four robust outlier detection methods or identifiers, through their outlyingness functions, in a simulation study. Two scenarios are considered for the occurrence of outliers: "the cluster" and "the radial". Conclusions and recommendations are offered for each scenario.

Keywords: outlier detection; data depth; outlyingness; skew-normal distribution.

1. Introduction

In order to detect outliers in multivariate data, it is common to associate with each data point a univariate (scalar) number describing its "position" with respect to the data cloud. The position is often measured in terms of a certain distance function and point(s) that are found extreme are considered outliers. For example, one approach is to compute a Mahalanobis distance (or a

robust version of it) for each data point and set a cut-off value based on the (approximate) distribution of those distances. The points with distances greater than the cut-off are deemed to be potential outliers. See for example, Rousseeuw and van Zomeren (1990), Rocke and Woodruff (1996), Filzmozer, Maronna and Werner (2008), Hardin and Rocke (2005) and Cerioli (2010). The robust versions of Mahalanobis distances are computed using certain robust estimators of location and dispersion, like the MCD and MVE (Rousseew, 1984; Rousseew and Leroy, 1985) which are high breakdown estimators that are also affine equivariant. The high breakdown property allows those estimators to resist up to 50% outliers and the affine equivariant property allows the estimators to change accordingly when the data are subject to affine transformations such as rotations, translations and a change of scale. Another approach to outlier detection is to use the concept of depth. Suitably defined, the depth of the data points, may be computed, which yield univariate measures which can be ordered naturally and extreme points can be located. The depth of a data point is calculated via a depth function, which is a real valued function that provides a "center-outward" ordering of the multivariate data. Several depth functions have been proposed (see more details further) in the literature. The lower the depth of a data point (the less deep it is within the data cloud), the more outlying it is and is potentially an outlier. Inversely equivalent to the ideas of the depth and the depth function are the concepts of outlyingness and outlyingness functions. Not surprisingly, the higher the outlyingness of a point, the more outlying the point is.

Once the outlyingness of the data points are computed the next step is to be able to judge which, if any, of the points have "extreme" outlyingness. To this end a cutoff value can be helpful. Any observation whose outlyingness is larger than the cutoff value may be deemed a potential outlier. Dang and Serfling (2010) studied some outlyingness functions and proposed

cut off values for the case of classical Mahalanobis distance outlyingness, halfspace (or Tukey) outlyingness and the Stahel-Donoho outlyingness (Stahel 1981; Donoho 1982). They assumed that the underlying distribution is multivariate normal as do most classical outlier detection techniques. In many applications however the underlying multivariate distribution is skewed. Very little work seems to have been done for such distributions. Among these, Hubert and Van der Veeken (2008) apply a skewness-adjusted boxplot to outlyingness values in order to detect outliers. Their proposed outlyingness function is a modification of the Stahel-Donoho outlyingness. Their motivation is that the exact distribution of the outlyingness values is unknown, but possibly skewed. An alternative approach would be to use the modified adjusted boxplot of Dovoedo and Chakraborti (2009), which is a modification of the adjusted boxplot of Hubert and vandervieren (2008), instead of the skewness-adjusted boxplot of Hubert and Van der Veeken (2008). We don't pursue this work here.

In this paper, we perform a comparative study of the outlier detection abilities of a number of robust and affine invariant outlyingness functions using data from the skew-normal distribution of Azzalini and Dalla Valle (1996). Specifically, the study focuses on (i) the robust Mahalanobis distance outlyingness (using MCD estimators) (Dang and Serfling, 2010), (ii) the robust Mahalanobis spatial outlyingness (using MCD estimators) (Dang and Serfling, 2010), (iii) the robust triangle depth outlyingness (based on the robust triangle depth function of Liu and Modarres, 2010),(iv) the robust elliptical outlyingness (based on the robust elliptical depth function of Elmore, 2005). Such a comparison is not available in the current literature.

This paper is structured as follows. In section 2, we review the various depth functions (and the corresponding outlyingness functions) that are used in the simulation study. In section 3, we present the simulation plan and the simulation results. As an illustration, the outlyingness

functions used in the simulation study are applied to a data set in section 4. Section 5 concludes our findings.

2. Some Data Depth Functions and Corresponding Outlyingness Functions

The idea of data depth was first introduced by Tukey (1975). A definition of a depth function, as provided in Dang and Serfling (2006), is that for a given a probability distribution F on $\mathbb{R}^p$, any function $D(x, F)$ which provides an F-based "center-outward" ordering of observations $x \in \mathbb{R}^p$ may be regarded as a depth function.

Note then that the depth function $D(x, F)$ measures how "deep" or "central" a point x is with respect to the distribution F. The deeper a point, the less likely it is an outlier. Following Tukey's (1975) halfspace depth, several depth functions have been proposed in the literature including the simplicial depth (Liu, 1990), the projection depth (Liu, 1992; Zuo 2003), The Mahalanobis depth (Liu and Singh, 1993), the spatial depth (Serfling, 2002; based on Chaudhuri, 1996), the elliptical depth (Elmore, 2005), the spherical depth (Elmore et al.. 2006), and most recently, the triangle depth (Liu and Modarres, 2010). Zuo and Serfling (2000) listed some desirable properties that a statistical depth function should have. These include affine invariance, maximality at the center, monotonicity relative to the deepest point and vanishing at infinity. The most important of these properties is *affine invariance* and it means roughly that the depth of a point $x \in \mathbb{R}^p$ with respect to a distribution F should not depend on the underlying coordinate system. Specifically, for any non-singular p-by-p matrix A and any px1 constant vector b

$$D(x; F) = D(Ax + b; F_{AX+b}),$$

where F and F_{AX+b} represent the distributions functions of data X and data $AX + b$.

Maximality at the center means roughly that the "center" of the distribution F (if it exists) should have the largest depth with respect to F.

Monotonicity relative to the deepest point means that as $x \in \mathbb{R}^p$ "moves away" from the "center" (deepest point), its depth should decrease. Specifically, if θ is the deepest point with respect to F, the following holds

$$D(x; F) \leq D(\theta + \alpha(x - \theta); F),$$

for any $\alpha \in [0,1]$

Vanishing at infinity means that as $||x||$ approaches infinity, the depth of x should approach zero (it vanishes).

$$D(x; F) \to 0 \text{ as } \|x\| \to +\infty$$

We now briefly review the outlyingness functions that will be used in the simulation study.

2.1 Mahalanobis Depth and Outlyingness

The Mahalanobis depth function (Liu and Singh, 1993) is based on the Mahalanobis distance (Mahalanobis, 1936). The Mahalanobis depth, $MDE(x, F)$, of $x \in \mathbb{R}^d$, with respect to a p-variate distribution F is defined in terms of the Mahalanobis squared distance $SMD(x, F)$, given by

$$MDE(x, F) = [1 + SMD(x, F)]^{-1},$$

and

$$SMD(x, F) = (x - \mu_F)^T {\Sigma_F}^{-1} (x - \mu_F),$$

where μ_F and Σ_F represent, respectively, the mean vector and the covariance matrix of F. The sample version of the Mahalanobis depth function is obtained by replacing μ_F and Σ_F with some estimatiors of location and dispersion, respectively. If the sample mean and the sample

covariance matrix of the data are used to compute the SMD, we get the classical squared Mahalanobis depth function, $CSMD(.,X)$. Note however that the classical estimators are non-robust and highly affected by outliers. It is thus common to replace them with robust estimators like the minimum covariance determinant (MCD) based estimators of location and scatter as will be done here. The resulting distance is the robust squared Mahalanobis distance, $RSMD(.,X)$.

Similarly, we can define the (sample) classical and the robust Mahalanobis depth function, $CMDE(.,X)$ and $RMDE(.,X)$, corresponding to the distances $CSMD(x,X)$ and $RSMD(x,X)$, respectively. These are given by

$$CMDE(.,X) = [1 + CSMD(x,F)]^{-1},$$

and

$$RMDE(.,X) = [1 + RSMD(x,F)]^{-1}.$$

From the robust Mahalanobis depth function $RMDE(.,X)$, for example, one can define a corresponding outlyingness function by

$$RMDO_1(x,X) = 1 - RMDE(x,X) = 1 - [1 + RSMD(x,F)]^{-1} = \frac{RSMD(x,X)}{1 + RSMD(x,X)}.$$

Instead of using the robust squared Mahalanobis distance $RSMD(x,X)$ to define the outlyingness function, one could use the robust (non-squared) Mahalanobis distance, $RMD(x,X)$, resulting in the a robust Mahalanobis depth outlyingness function, which we denote by $RMDO(.,X)$. This outlyingness function is used in Dang and Serfling (2010) (with MCD estimators) and will be used in our simulation.

$$RMDO(x,X) = \frac{RMD(x,X)}{1 + RMD(x,X)}$$

2.2 Spatial Depth and Outlyingness

The idea of spatial depth was formally introduced by Serfling (2002). It is based on the notion of a spatial quantile introduced by Chaudhuri (1996) and Koltshinskii (1997). Specifically, Chaudhuri (1996) shows that a unique quantile $Q_F(u)$ always exists for any $u \in \mathbb{B}(0,1)$, if the dimension of the data d is greater than 2, and F is not supported on a straight line. He also shows that $Q_F(u)$ may be represented as a solution $x = x_u$ of the following equation:

$$E\left\{\frac{x - X}{\|x - X\|}\right\} = u$$

The preceding equation shows, as Serfling (2002) points out, that each point $x = x_u \in \mathbb{R}^d$ can be viewed as the spatial quantile of $u_x \in \mathbb{B}(0,1)$, $x_u = Q_F(u_x)$ and that u_x can be viewed as the "average" of unit vectors pointing to x_u from a random point having distribution F. Then, it is not difficult to see that the quantile x corresponding to u is "more central" in case $\|u\|$ is close to 0 and it is "more extreme" in case $\|u\|$ is close to 1, with $0 \leq \|u\| < 1$. The idea is that when x is more "central", the averaging "balances out" while if x is more "extreme", it does not.

So, $\|u\| = \|Q_F^{-1}(x)\| = \left\|E\left\{\frac{x-X}{\|x-X\|}\right\}\right\|$ measures "outlyingness" and $1 - \|Q_F^{-1}(x)\|$ could measure the "spatial depth" of observation x with respect to the distribution F as follow:

$D_S(x,F) = 1 - \|Q_F^{-1}(x)\| = 1 - \left\|E\left\{\frac{x-X}{\|x-X\|}\right\}\right\|$ where $X \sim F$.

The vector sign function in $\mathbb{R}^d$ could be used in the formulation of the definition of the "Spatial depth"; the vector sign function in $\mathbb{R}^d$ is defined by:

$$S(x) = \begin{cases} \frac{x}{\|x\|}, & if\ x \neq 0 \\ 0, & if\ x = 0 \end{cases}.$$

We can then rewrite the spatial depth function $D_S(x,F) = 1 - \|E_F(S(x - X))\|$ and the spatial outlyingness function is given by $O_S(x,F) = \|E_F(S(x - X))\|$.

The sample version of the spatial depth function can be inferred from the formula $D_S(x,F) = 1 - \|E_F(S(x-X))\|$. It is given by:

$$D_S(x,F_n) = 1 - \left\| n^{-1} \sum_{i=1}^{n} S(x-X_i) \right\|.$$

Dang and Serfling (2010) pointed out that the spatial outlyingness function defined by $O_S(x,F) = \|E_F(S(x-X))\|$ is only orthogonally equivariant. They suggested that, in order to the make the spatial depth affine invariant, a "weak covariance functional" may be used. This is, any symmetric positive, matrix-valued functional $C(F)$ defined on distribution F on $\mathbb{R}^d$ that satisfy the "weak covariance equivariance" condition:

$$C(F_{AX+b}) = k(A,b,F_X)AC(F_X)A' ,$$

for any nonsingular dxd matrix A and any real number b with $k(A,b,F_X)$ a positive scalar function. Dang and Serfling (2010) then gave the affine invariant version the spatial outlyingness function as follow.

$$O_S(x,F) = \left\| E_F \left(S(C(F_X)^{-\frac{1}{2}}(x-X)) \right) \right\|$$

When the weak covariance functional is the sample covariance, this gives the classical Mahalanobis spatial outlyingness function. When the weak covariance functional is a robust measure of dispersion, like the MCD covariance matrix, the resulting outlyingness is called robust Mahalanobis spatial outlyingness, and is denoted RMSO hereafter. RMSO (with MCD) is used in the simulation study later. It is given by:

$$RMSO(x,F) = \left\| E_F \left(S(C(F_X)^{-\frac{1}{2}}(x-X)) \right) \right\| ,$$

where $C(F_X)$ being the MCD estimator of scatter (Serfling 2010).

2.3 Elliptical Depth and Outlyingness

Let X_1 and X_2 be two $i.i.d.$ random variables from a distribution F on $\mathbb{R}^d, d \geq 1$. Elmore (2005) introduced the elliptical depth as follows.

$$EDE(x, C_F) = P_F[x \in e(X_1, X_2)],$$

where $e(X_1, X_2)$ is a closed random hyper-ellipse formed by X_1 and X_2, and C_F a positive definite matrix. The elliptical region $e(X_1, X_2)$ is given by

$$e(X_1, X_2) = \{t: (X_1 - t)^T {C_F}^{-1} (X_2 - t) \leq 0\}.$$

He showed that the elliptical depth has desirable properties of a depth function (see section 1).

It can be seen that $EDE(x, C_F)$ can be estimated by the proportion of random hyper-ellipses $e(X_i, X_j)'s$ that contain x where X_i and X_j are simulated from F a large number of times. It not difficult to see, as Elmore (2005) pointed out, that the sample triangle depth function is defined by:

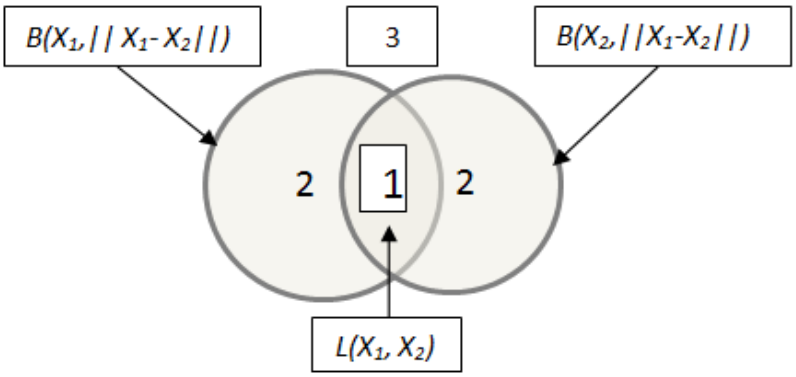
$$EDE\left(x, C_{F_n}\right) = \frac{1}{\binom{n}{2}} \sum_{i<j}^{n} I(x \in e(X_i, X_j)) ,$$

where $X_1, X_2, \ldots, X_n$ is a random sample from a distribution F on $\mathbb{R}^d$ and $I(A)$ is the usual indicator function of event A. Using this depth function, a robust elliptical depth outlyingness function, which we denote REDO and use later in the simulation study, can be defined as follows: $REDO\left(x, C_{F_n}\right) = 1 - EDE\left(x, C_{F_n}\right)$, where C_{F_n} is the MCD estimator of scatter.

2.4 Triangle Depth and Outlyingness

Liu and Modarres (2010) recently introduced the triangle depth. Let X_1 and X_2 be two $i.i.d.$ random variables from a distribution F on $\mathbb{R}^d, d \geq 1$. Consider the balls $B(X_1, \|X_1 - X_2\|)$

and $B(X_2, \|X_1 - X_2\|)$. Liu and Modarres (2010) pointed out that these balls partition the data into three regions, region 1 (the "lens area"), region 2 (the "lunar areas") and region 3 (outside the balls).



They observed that for a given point x, if t is in the "lens area" denoted $L(X_1, X_2)$, that is an indication of the "closeness" of x with respect to the distribution F. This led them to define of the triangle depth as follow.

$$TDE(x, F) = \Pr\big(x \in L(X_1, X_2)\big),$$

where X_1 and X_2 are $i.i.d.$ from F.

It can then be seen, as pointed out in Liu and Modarres (2010), that $TDE(x, F)$ can be estimated by the proportion of random hyper-lenses $L(X_i, X_j)'s$ that contains x, where X_i and X_j are simulated from F a large number of times. They observed, in addition, that $x \in L(X_i, X_j)$ is equivalent to $\|X_i - X_j\| > \max(\|x - X_i\|, \|x - X_j\|)$ which allows an easy computation of the sample triangle depth function defined by:

$$TDE(x, F_n) = \frac{1}{\binom{n}{2}} \sum_{i<j}^{n} I(x \in L(X_i, X_j)),$$

where $X_1, X_2, \ldots, X_n$ is a random sample from the distribution F on $\mathbb{R}^d$ and $I(A)$ is the usual indicator function of event A.

Liu and Modarres (2010) argued that replacing the Euclidian distance by the Mahalanobis distance in $\|X_i - X_j\| > \max(\|x - X_i\|, \|x - X_j\|)$ makes the triangle depth affine-invariant. Let the resulting depth function be denoted by $RTDE(x, F_n)$. A robust triangle depth outlyingness function could then defined as follow and used in the simulation study further.

$$RTDO(x, F_n) = 1 - RTDE(x, F_n)$$

We summarized in Table 6.1 (see appendix), the formulae of the outlyingness functions just discussed and that will be used in the simulation study below.

3. Simulation Study

There has been some work on comparing the outlier detection capabilities of the outlyingness functions. Hubert and van der Veeken (2008) proposed a modification of the Stahel-Donoho outlyingness, called the adjusted outlyingness. They then compared the outlier detection capability of their adjusted outlyingness to the Stahel-Donoho outlyingness with multivariate skew-normal data. Dang and Serfling (2010) compared the outlier detection capabilities of some of the outlyingness functions, including the robust Mahalanobis distance outlyingness and the robust Mahalanobis spatial outlyingness at multivariate normal data.

However, to the best of our knowledge, a study comparing the outlier detection capabilities of the robust affine outlyingness functions (invariant classifiers) reviewed in section 2, under multivariate skew-normal data, is not available in the current literature. This is what is undertaken in this section. Specifically, we compare the robust Mahalanobis distance outlyingness, RMDO, the robust Mahalanobis spatial outlyingness, RMSO, the robust Mahalanobis elliptical outlyingness, REDO and the robust triangle outlyingness, RTDO.

3.1 Simulation Plan

The plan of our simulation study is similar to that of Hubert and van der Veeken (2008). The regular or uncontaminated observations are generated from a multivariate skew-normal distribution given in Azzalini and Dalla Valle (1996). A p-dimensional random variable X is said to follow a multivariate skew-normal distribution with a vector of shape parameters α and with dependence parameter $\boldsymbol{\Omega}$, when its probability density function is given by

$$f_p(\boldsymbol{x}) = 2\phi_p(\boldsymbol{x}, \boldsymbol{\Omega})\Phi(\alpha^T \boldsymbol{x}) ,$$

where $\phi_p(\boldsymbol{x}, \boldsymbol{\Omega})$ is the p-dimensional normal density with mean zero and correlation matrix $\boldsymbol{\Omega}$ and Φ is the standard normal distribution function. Observe then that when the shape vector $\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\Omega} = \boldsymbol{I_p}$ then the skew-normal distribution reduces to the standard multivariate normal distribution. Thus, the term skew-normal distribution refers to a class of distributions that includes the standard multivariate normal distribution.

In the simulations performed here, we use $\boldsymbol{\Omega} = \boldsymbol{I_p}$ and take the vector of shape parameters $\boldsymbol{\alpha} = (10, 4)^T$ when $p = 2$ (bivariate skew-normal distribution) and $\alpha = (10, 4, 4)^T$ when $p = 3$ (trivariate skew-normal distribution). Our shape parameter value ($\boldsymbol{\alpha}$) are similar to the ones used by Hubert and Van Der Veeken (2008). Note that our simulations are intentionally limited to smaller dimensional data because skewness is a known to be a more critical issue in smaller dimension, as pointed out in Hastie et al. (2001) and confirmed by Hubert and Van der Veeken (2008).

In multivariate outlier studies, two outlier scenarios are often explored. The first scenario is referred to as "Cluster" and the second can be referred to as "Radial". Details about these scenarios are provided below along with Figures 6.1 and 6.2 where they are displayed. We focus on these two scenarios, with our underlying data being multivariate skew-normal.

Scenario 1 (Cluster): The outliers are randomly generated from a multivariate normal distribution with $\boldsymbol{I_p}/20$ (p is the dimension of the data) as covariance matrix and a center located along the $-\mathbf{1_p}$ direction. Specifically the center of the outlier generating distribution is $c = (-b, -b)^T$ when $p = 2$ and $c = (-b, -b, -b)^T$ when $p = 3$. Note that the parameter $b > 0$ controls how far the center of the outlier generating distribution is from the origin, which is the "center" of the distribution of the regular observations. This outlier generation scheme was used by Hubert and Van der Veeken (2008).

[Figure 6.1 here; see appendix]

Scenario 2 (Radial): Here, the regular observations in a sample of size n are contaminated with m outliers. To do this, first generate the regular observations from a skew normal distribution. Let $X_{(n-m+1)}, \ldots, X_{(n)}$ denote the m observations with the largest Euclidian norm (among the original regular observations). Next, replace $X_{(n-m+1)}, \ldots, X_{(n)}$ by $BX_{(n-m+1)}, \ldots, BX_{(n)}$ where B is some "inflation" factor ($B > 1$). Note that the parameter B controls how far the outliers are from the regular observations. Similar outlier generation scheme is used in Dang and Serfling (2010).

[Figure 6.2 here; see appendix]

In the simulation study, we consider situations where data sets of size $n = 100$ are generated from the bivariate and trivariate skew-normal distribution with 10% outliers according to the two scenarios just described.

As discussed earlier, observations with "large" outlyingness values are to be declared as outliers in the simulations but then the question is how large does the outlyingness of an observation have to be for this observation to be labeled an outlier? To answer this, one could apply the traditional boxplot of Tukey (1977), the adjusted boxplot procedure by Hubert and

vandervieren (2008) or the modified adjusted boxplot by Dovoedo and Chakraborti (2010) to the outlyingness values. However, our empirical investigation to this end revealed that this approach is not successful in the simulations undertaken here. This is because the upper fences of all these boxplots tend to be larger than 1, while the outlyingness values all lie between 0 and 1; consequently, no outliers are detected using the boxplots. Hence the approach of Dang and Serfling (2010) is adopted: simulate some uncontaminated data from the assumed underlying distribution, compute the outlyingness values corresponding to a given outlyingness function in Table 6.2 (see appendix) and then compute the estimated 99^{th}, 95^{th} and 90^{th} percentiles of the distribution of outlyingness values. We then use these percentiles as cut off values, above which the outlyingness of an observation from contaminated data has to fall, for the observation to be declared a potential outlier. However, in contrast to Dang and Serfling (2010), to increase the precision of the estimates, we averaged each percentile over 1000 simulated uncontaminated data instead of estimating them from a single uncontaminated dataset.

3.2 Simulation Results

First, we report in the Table 6.2 (see appendix), the 90^{th}, 95^{th} and 99^{th} percentiles of the distributions of the outlyingness values for each of the four outlyingness functions RMDO, RMSO, REDO, and RTDO, respectively, when the underlying distributions are the multivariate skew-normal distributions specified in the simulation plan. Recall that the outlyingness functions are respectively, the robust Mahalanobis depth outlyingness, the robust Mahalanobis spatial outlyingness, the robust elliptical depth outlyingness, and the robust triangle depth outlyingness(all based on MCD estimators of location and scale). The values reported in Table

6.2 are based on the average over 1000 simulations. The approximate 95% margin of error is at most 0.002.

The percentiles given in Table 6.2 are the cut-off (critical) values that are used to detect multivariate outliers. Specifically, in a contaminated data situation, any observation with outlyingness value larger than a critical value (percentile of interest) is declared as a potential outlier. Due to the computational burden however, it is customary in this kind of multivariate setting, to use a moderate number of simulations. For example Cerioli and Farcomeni (2011) used 200 simulations for some power comparison in multivariate settings similar to the ones being explored here. The results reported below are based on 1000 simulations.

For each of the four outlyingness functions, we report in Table 6.3 (see appendix), the percentage of outliers detected (POD) as well as the percentage of regular observations declared as outliers (PRDO) for various values of the parameters b-cluster (respectively B-radial) in Scenarios 1 and 2, respectively. The results are displayed in Figures 6.3, 6.4, and 6.5 (respectively Figures 6.6, 6.7, and 6.8).

3.2.1 Discussion of Results for Scenario 1: Cluster Outliers

-Using the 99th percentiles of outlyingness values to detect outliers for Scenario 1 (cluster):

In Table 6.3 (for the 99th percentiles), the approximate 95% margin of error is 2.21% at most, for the percentages of outliers detected (POD) and 0.052% for the percentages of regular observations declared as outliers (PRDO). The results in Table 6.3 (with the 99th percentiles) are displayed in Figure 6.3. It is seen that using the 99th percentiles, for both the two-dimensional and three-dimensional data with cluster outliers, (1) The RMDO clearly outperform others namely RMSO, RTDO, and REDO (2) RMSO is slightly better than RTDO and REDO.

We observe that the 99th percentiles used in case of RTDO, REDO and perhaps RMSO are high and it may be of interest to try using the 95th or even the 90th percentiles. That is what is done next.

[Figure 6.3 here; see appendix]

-Using the 95th percentiles of outlyingness values to detect outliers for Scenario 1 (cluster):

In Table 6.3 (for the 95th percentiles), the approximate 95% margin of error is 0.372% for the percentages of outliers detected (POD) and 0.098% for the percentages of regular observations declared as outliers (PROD). The results in Table 6.3 (with the 95th percentiles) are displayed on Figure 6.4. It is seen that using the 95th percentiles, for both the two-dimensional and three-dimensional data with cluster outliers: (1) The RMDO clearly outperform others, namely RMSO, RTDO, and REDO when taking into account the percentage of outliers detected. However, the percentage of regular observations falsely declared as outliers tend to be higher for the RMDO. (2) RMSO is better than RTDO and REDO when taking into the percentage of outliers detected. But The percentage of regular observations falsely declared as outliers is similar for RMSO and RTDO and tend to be slightly higher than that for REDO. (3) RMSO outlier detection capabilities (using the 95th percentile) is higher than in the previous case (using the 99th percentile).

The outlier detection capabilities of RTDO and REDO still remain low. Further, we investigate using the 90th percentiles as cut-off.

[Figure 6.4 here; see appendix]

-Using the 90th percentiles of outlyingness values to detect outliers for Scenario 1 (cluster):

In Table 6.3 (for the 90th percentiles), the approximate 95% margin of error is 0.558% at most, for the percentages of outliers detected (POD) and 0.143% for the percentages of regular

observations declared as outliers (PRDO). The results in Table 6.3 (with the 99th percentiles) are displayed on Figure 6.5. It appears that using the 90th percentiles, for both the two-dimensional and three-dimensional data with cluster outliers: (1) the percentage of regular observations declared as outliers is higher for all methods than using the 99th and 95th percentiles as cut-offs and (2) in order of preference, in this case, we have the methods RMDO, RMSO, RTDO, and REDO respectively.

[Figure 6.5 here; see appendix]

3.2.2 Discussion of Results for Scenario 2: Radial Outliers

-Using the 99th percentiles of outlyingness values to detect outliers for Scenario 2 (radial):

In Table 6.4 (for the 99th percentiles), the approximate 95% margin of error is 1.35% at most, for the percentages of outliers detected (POD) and 0.043% for the percentages of regular observations declared as outliers (PRDO). The results in Table 6.4 (with the 99th percentiles) are displayed on Figure 6.6. It is seen that using the 99th percentiles, for both the two-dimensional and three-dimensional data with radial outliers: (1) RMDO outperforms others namely RMSO, RTDO, and REDO. However, this comes with the cost that the percentage of regular observations falsely declared as outliers is slightly higher. (2) RMSO is better than RTDO and REDO. This is more pronounced for three dimensional data. (3) The percentages of regular observations declared as outliers are lower for RMSO, REDO and RTDO compared to RMDO.

Again, we observe that the 99th percentiles used in case of RTDO, REDO and RMSO are high. This may be why RTDO, REDO and RTDO do not perform that well. Next, we use the 95th or even the 90th percentiles.

[Figure 6.6 here; see appendix]

-Using the 95th percentiles of outlyingness values to detect outliers for Scenario 2 (radial):

In Table 6.4 (for the 95th percentiles), the approximate 95% margin of error is 1.1% at most for the percentages of outliers detected (POD) and 0.169% for the percentages of regular observations declared as outliers. The results in Table 6.4 (with the 95th percentiles) are displayed on Figure 6.7. It is seen that using the 95th percentiles, for both the two-dimensional and three-dimensional data with radial outliers: (1) RMDO and RMSO clearly outperform others, namely RTDO, and REDO when taking into account the percentage of outliers detected. However, the percentage of regular observations falsely declared as outliers tend to be higher for the RMDO than other procedures. Note also that the percentage of regular observations declared as outliers tend to be lower for RMSO. (2) RTDO is better than REDO, overall. The percentage of regular observations falsely declared as outliers is similar for RMSO and RTDO and tend to be slightly higher than that for REDO. (3) RMSO outlier detection capability (using the 95th percentile) is higher than in the previous case (using the 99th percentile).

The outlier detection capabilities of RTDO and REDO are still not high enough. Further, we investigate using the 90th percentiles.

[Figure 6.7 here; see appendix]

-Using the 90th percentiles of outlyingness values to detect outliers for Scenario 2 (radial):

In Table 6.4 (for the 90th percentiles), the approximate 95% margin of error is 1.084% at most, for the percentages of outliers detected (POD) and 0.247% for the percentages of regular observations declared as outliers (PRDO). The results in Table 6.4 (with the 90th percentiles) are displayed on Figure 6.8. It is seen that using the 90th percentiles, for both the two-dimensional and three-dimensional data with radial outliers: (1) Not surprisingly, the percentage of regular

observations declared as outliers is higher for all methods than using the 99^{th} and 95^{th} percentiles as cut-offs. (2) In order of preference, in this case, we have the methods RMSO, RMDO, RTDO, and REDO, respectively.

[Figure 6.8 here; see appendix]

To summarize, in general, the RMDO and the RMSO are preferred to the RTDO and the REDO in the situations studied.

4. Illustration

We illustrate the outlier detection capabilities of the four outlyingness functions RMSO, RMDO, RTDO, and REDO by applying them to a well known set of data, the so-called bushfire scars data, reported in Campbell (1989). Note that we declare as potential outliers, observations with the largest outlyingness values. The data set has $N = 38$ observations (pixels) of satellite measurements on $p = 5$ frequency bands used to locate bushfire scars. It is known that this data set contains 12 outliers. Maronna and Yohai (1985) classified these outliers in two groups. The group of observations 7-11, which, they pointed out, are easy to detect, and the group of observations 32-38 which are more difficult to detect because they tend to be masked by the first group.

The outlyingness values (RMDO, RMSO, RTDO and REDO) of the 38 points in the data set are reported in the Table 6.5 (see appendix). The criterion used here is that extreme clusters of outlyingness values (shown in bold) correspond to outlying observations. It can be seen from Table 6.5 that all four of the outlier functions studied detect the cluster 8-11. RMDO, RMSO, and RTDO detected the group of outliers 32-38 but REDO missed some outliers in that group. Table 6.6 (see appendix) lists for each outlier identifier, the outliers that were not detected and

the percentage of outliers not detected. It also lists the regular observations classified as outliers and the corresponding percentages.

It is seen that the RMDO suffers the least from swamping (fewer regular observations are declared as outliers), followed by the RMSO, whereas the RTDO and the REDO suffer the most. The RMDO and RMSO do not suffer from masking (all outliers are detected i.e., the presence of one group of outliers did not prevent from detecting the other group of outliers), while the RTDO suffers from some and the REDO suffers severely. Thus, for this data set, we have the following preferential order, from most preferred to least preferred: RMDO, RMSO, RTDO and finally REDO.

5. Summary and Conclusions

In case of the skew-normal distributions studied in the simulations, the following observations can be made:

Cluster outliers (Scenario 1):

(1) When using the procedure RMDO, it may be better to use the 99^{th} percentile, while the 90^{th} percentiles are recommended with procedures RMSO and RTDO in order to detect a reasonably high percentage of outliers.

(2) Even when using the 90^{th} percentile with REDO does not help detect a reasonably high percentage of outliers.

(3) In the case of Cluster outliers, in order of preference, we recommend the RMDO (using the 99^{th} percentile), the RMSO and the RTDO (using the 90^{th} percentiles).

Radial outliers (Scenario 2):

(1) When using the RMDO procedure, it is advisable to use the 99th percentile as the cut-off value while the 95th percentile is advisable for the RMSO. For the RMSO and RTDO, the 90th percentiles are recommended so as to detect a reasonably high percentage of outliers.

(2) Taking into account both the percentage of outliers detected and the percentage of regular observations declared as outliers, we recommend the following in order of preference among the four procedures: RMSO (using the 95th percentile), RMDO (using the 99th percentile), RTDO and REDO (using the 90th percentile).

APPENDIX

A-TABLES

Table 6.1: Formulae for outlyingness functions used in the simulation study

$RMDO(x, C_{F_n})$	$\frac{[(x-\mu_{1F_n})^T {C_{F_n}}^{-1}(x-\mu_{1F_n})]^{1/2}}{1+[(x-\mu_{1F_n})^T {C_{F_n}}^{-1}(x-\mu_{1F_n})]^{1/2}}$
$RMSO(x, C_{F_n})$	$\left\| E_{F_n}\left(S((C_{F_n})^{-\frac{1}{2}}(x-X))\right)\right\|$
$REDO(x, C_{F_n})$	$1-\frac{1}{\binom{n}{2}}\sum_{i<j}^{n} I(x \in \{t: (X_i-t)^T {C_{F_n}}^{-1}(X_j-t) \leq 0\})$
$RTDO(x, F_n)$	$1-\frac{1}{\binom{n}{2}}\sum_{i<j}^{n} I(x \in \{t: \|X_i-X_j\| > \max(\|t-X_i\|, \|t-X_j\|)\})$

Table 6.2: The 90th, 95th and 99th percentiles of the distributions of four selected outlyingness function estimates for $n = 100$ when the data come from the specified multivariate skew-normal distributions

		Dimension	
Outlyingness	Percentile	p=2	p=3
RMDO	90th	0.7319896	0.7509390
	95th	0.7652370	0.7787556
	99th	0.8239993	0.8275000
RMSO	90th	0.8584660	0.8395284
	95th	0.8974345	0.8757869
	99th	0.9529643	0.9300310
RTDO	90th	0.9006143	0.8984187
	95th	0.9455114	0.9433745
	99th	0.994392	0.9912996
REDO	90th	0.9277152	0.9477841
	95th	0.9562338	0.9663864
	99th	0.9790174	0.9792730

Table 6.3: Case of 2D-3D Cluster outliers: The performance of four outlyingness functions using the 99th, the 95th and the 90th Percentiles of outlyingness values of underlying uncontaminated distributions. POD stands for percent of outliers detected, and PRDO stands for percent of regular observations declared as outliers

		Parameter	RMDO		RMSO		RTDO		REDO	
Percentile	p	b	POD	PRDO	POD	PRDO	POD	PRDO	POD	PRDO
99	2	0.1	0	0.7878	0.01	0.55	0.04	0.6933	0.27	0.9867
		0.5	0.11	0.3367	0.64	0.4167	1.39	0.5344	2.58	0.7822
		1	21.07	0.6067	7.1	0.5533	7.23	0.5678	6.09	0.6889
		1.5	88.68	0.6633	11.77	0.5667	8.84	0.5822	7.31	0.6556
		2	99.97	0.6667	14.29	0.5556	9.23	0.5811	7.85	0.6433
		2.5	100	0.6678	15.74	0.5456	9.4	0.5778	8.04	0.6367
	3	0.1	0	0.7456	0	0.5567	0	0.7156	0.02	1.0133
		0.5	0.11	0.5178	0.49	0.51	0.94	0.6367	1.2	0.8667
		1	40.7	0.6289	9.71	0.5511	7.76	0.5678	5.01	0.6111
		1.5	98.59	0.6511	17.62	0.5489	9.11	0.5644	6.3	0.5733
		2	100	0.65	21.55	0.53	9.33	0.5611	6.8	0.5578
		2.5	100	0.6489	23.93	0.5167	9.43	0.5556	7.22	0.5478
95	2	0.1	0.03	5.2078	2.63	5.1089	3.27	4.81	5.51	4.97
		0.5	3.38	3.11	14.15	4.3522	16.02	4.1311	14.01	4.3567
		1	76.65	3.7067	34.21	3.6833	29.72	3.3889	17.96	3.2244
		1.5	99.54	3.9644	43.53	3.3567	31.5	3.1411	18.76	2.7622
		2	100	4.0444	47	3.3022	31.68	3.1489	19.04	2.6622
		2.5	100	4.0522	48.38	3.2844	31.53	3.1189	19.04	2.6044
	3	0.1	0	5.5411	0.14	5.4411	0.25	5.1689	0.82	5.4033
		0.5	4.28	4.1756	6.84	4.6244	8.83	4.4989	6.38	4.6133
		1	87.29	3.9578	38.09	3.5022	30.45	3.2456	11.14	2.92
		1.5	99.83	3.9767	52.57	3.1244	33.08	3.0133	10.91	2.3544
		2	100	3.9978	58.58	2.9889	33.07	2.9578	10.94	2.2244
		2.5	100	3.9589	61.9	2.8756	33.06	2.8889	10.88	2.0744
90	2	0.1	0.26	10.112	10.31	9.9644	10.96	9.6522	13.52	9.7756
		0.5	12.39	6.5411	30.68	8.8278	33.85	8.4522	28.32	8.79
		1	89.83	7.2411	56.44	7.1978	54.87	6.7944	32.13	6.6
		1.5	99.93	7.3922	66.4	6.4822	58.16	6.3	31.66	5.8489
		2	100	7.5	69.82	6.2122	58.76	6.16	31.58	5.5089
		2.5	100	7.2611	71.68	6.0622	58.76	6.1278	31.56	5.3022
	3	0.1	0.02	11.143	1.25	11.242	1.62	10.768	2.99	10.828
		0.5	12.89	8.2878	16.83	9.6567	20.54	9.3311	14.79	9.4844
		1	93.84	7.28	61.3	7.0178	55.93	6.7956	21.62	6.1989
		1.5	100	7.52	75.76	6.3	59.36	6.2489	21.53	5.1378
		2	100	7.5656	80.73	5.8533	59.58	6.0922	21.47	4.7689
		2.5	100	7.8156	84.25	5.7389	59.35	6.0089	21.19	4.5811

Table 6.4: Case of 2D-3D Radial outliers: Performance of four outlyingness functions using the 99th, the 95th and the 90th percentiles of outlyingness values of underlying uncontaminated distributions. POD stands for percent of outliers detected, and PRDO stands for percent of regular observations declared as outliers

		Parameter	RMDO		RMSO		RTDO		REDO	
Percentile	P	B	POD	PRDO	POD	PRDO	POD	PRDO	POD	PRDO
99	2	1.25	35.81	0.0433	10.15	0	8.32	0.0077	10.09	0.0355
		1.5	67.37	0.04	13.37	0.0011	9.11	0.0033	11.31	0.0144
		2	98.69	0.03	18.08	0	9.69	0.0011	11.62	0.0066
		3	100	0.0611	22.19	0	9.96	0.0033	12.16	0.0055
		4	100	0.0566	22.35	0	10.29	0.0011	12.01	0.0044
		5	100	0.0566	22.55	0	10.36	0.0011	12.17	0.0022
	3	1.25	36.32	0.1156	16.78	0	10.12	0.0022	12.08	0.0056
		1.5	73.56	0.12	29.23	0	11.62	0.0011	14.13	0.0033
		2	99.81	0.1244	51.22	0	12.68	0.0011	15.59	0.0011
		3	100	0.1255	71.24	0	13.3	0.0011	16.95	0
		4	100	0.1255	77.44	0	13.36	0.0011	17.58	0
		5	100	0.1244	79.44	0	13.26	0.0011	18.06	0
95	2	1.25	81.71	1.2911	62.69	0.3944	47.55	0.6466	42.22	0.7722
		1.5	97.74	1.1688	76.16	0.2733	49.65	0.4811	43.7	0.5411
		2	99.99	1.4588	87.78	0.1633	49.55	0.4255	44.74	0.3866
		3	100	1.2011	92.82	0.1222	45.77	0.41	43.59	0.2788
		4	100	1.41	93.76	0.1588	44.54	0.4166	43.72	0.2955
		5	100	1.36	93.61	0.1	43.45	0.3766	43.18	0.2322
	3	1.25	83.45	1.5422	74.16	0.2522	58.74	0.3767	47.3	0.456
		1.5	98.64	1.4088	93.33	0.1933	66.18	0.3022	50.88	0.2977
		2	100	1.5722	99.44	0.0744	65.84	0.2033	50.88	0.1422
		3	100	1.6077	99.97	0.0533	62.18	0.2022	48.23	0.0744
		4	100	1.5122	99.99	0.0466	60.16	0.1977	46.96	0.0477
		5	100	1.4033	99.97	0.0444	59.85	0.1767	46.68	0.0544
90	2	1.25	96.46	3.5388	90.92	2.1111	83.16	2.5855	74.26	2.7277
		1.5	99.88	3.7922	97.17	1.8455	86.8	2.3866	73.7	2.4677
		2	100	3.7822	99.78	1.5566	86.94	2.3266	73.03	2.1244
		3	100	3.6255	99.94	1.3188	85.01	2.2077	70.17	1.7922
		4	100	3.5277	99.93	1.3011	83.53	2.3	68.71	1.7688
		5	100	3.6355	99.94	1.1833	84.24	2.2288	68.39	1.5455
	3	1.25	96.79	4.0411	95.3	1.9344	90.19	2.2422	79.09	2.0889
		1.5	99.98	4.0233	99.57	1.3433	96.28	1.8433	80.76	1.6055
		2	100	4.0677	100	0.8344	96.26	1.5911	79.35	0.9988
		3	100	3.8733	100	0.62	95.31	1.5111	77.2	0.6111
		4	100	3.9211	100	0.5366	94.47	1.5022	76.76	0.4833
		5	100	3.8756	100	0.47	93.58	1.49	74.54	0.4567

Table 6.5: Outlyingness values for the bushfire data

Obs.	RMDO	RMSO	RTDO	REDO	Obs.	RMDO	RMSO	RTDO	REDO
1	0.678	0.574	0.644	0.734	20	0.647	0.342	0.458	0.552
2	0.614	0.470	0.509	0.657	21	0.633	0.403	0.477	0.602
3	0.629	0.547	0.626	0.701	22	0.725	0.596	0.664	0.727
4	0.608	0.491	0.569	0.669	23	0.612	0.520	0.550	0.691
5	0.681	0.452	0.535	0.622	24	0.613	0.577	0.615	0.755
6	0.743	0.494	0.538	0.605	25	0.582	0.530	0.590	0.708
7	**0.869**	0.674	0.609	0.674	26	0.596	0.581	0.654	0.758
8	**0.948**	0.864	**0.949**	**0.895**	27	0.601	0.561	0.630	0.734
9	**0.949**	0.905	**1.000**	**0.946**	28	0.672	0.554	0.623	0.710
10	**0.921**	0.753	**0.883**	**0.846**	29	0.736	0.590	0.666	0.706
11	**0.911**	0.704	**0.768**	**0.801**	30	0.769	0.512	0.586	0.583
12	0.784	0.580	0.619	0.671	31	**0.896**	0.579	0.691	0.647
13	0.729	0.668	**0.770**	**0.822**	32	**0.955**	**0.728**	**0.735**	0.681
14	0.707	0.696	**0.789**	**0.876**	33	**0.961**	**0.841**	**0.890**	**0.855**
15	0.702	0.710	**0.851**	**0.917**	34	**0.961**	**0.795**	**0.814**	0.761
16	0.677	0.623	**0.711**	**0.815**	35	**0.962**	**0.857**	**0.986**	**0.940**
17	0.584	0.412	0.486	0.605	36	**0.961**	**0.784**	**0.777**	0.725
18	0.682	0.429	0.486	0.589	37	**0.961**	**0.819**	**0.868**	**0.819**
19	0.678	0.340	0.428	0.538	38	**0.961**	**0.861**	**0.954**	**0.912**

Table 6.6: Performance of outlyingness functions under study on the bushfire data

	List [percent] of outlying observations Missed	List [percent] of regular observations classified as outliers
RMDO	[0%]	31 [3.85%]
RMSO	[0%]	13; 14; 15 [11.54%]
RTDO	7 [8.33%]	13; 14; 15; 16 [15.38%]
REDO	7; 32; 34; 36 [33.33%]	13; 14; 15; 16 [15.38%]

B-FIGURES

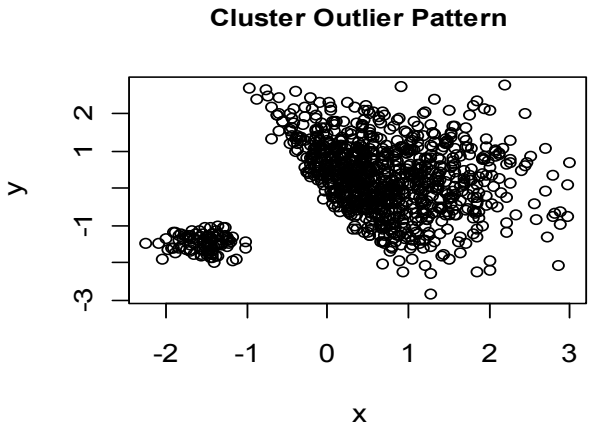


Figure 6.1: simulated bivariate skew-normal data with 10% *cluster* outliers

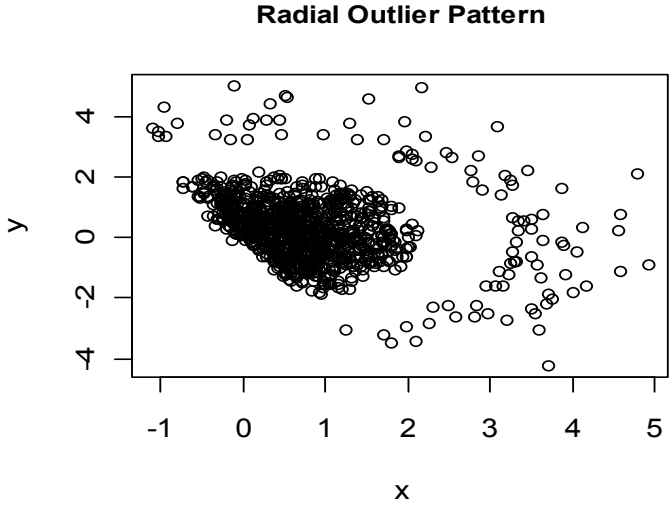


Figure 6.2: simulated bivariate skew-normal data with 10% *radial* outliers

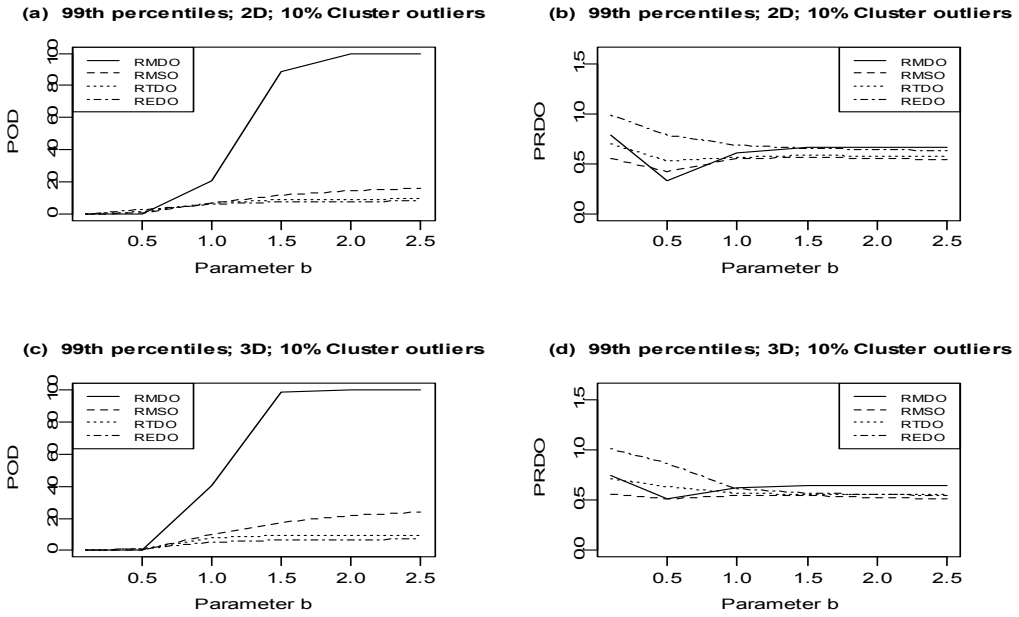


Figure 6.3: Cluster outliers. Simulation results for two-dimensional and three-dimensional data of size $n = 100$ using the 99th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD), whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

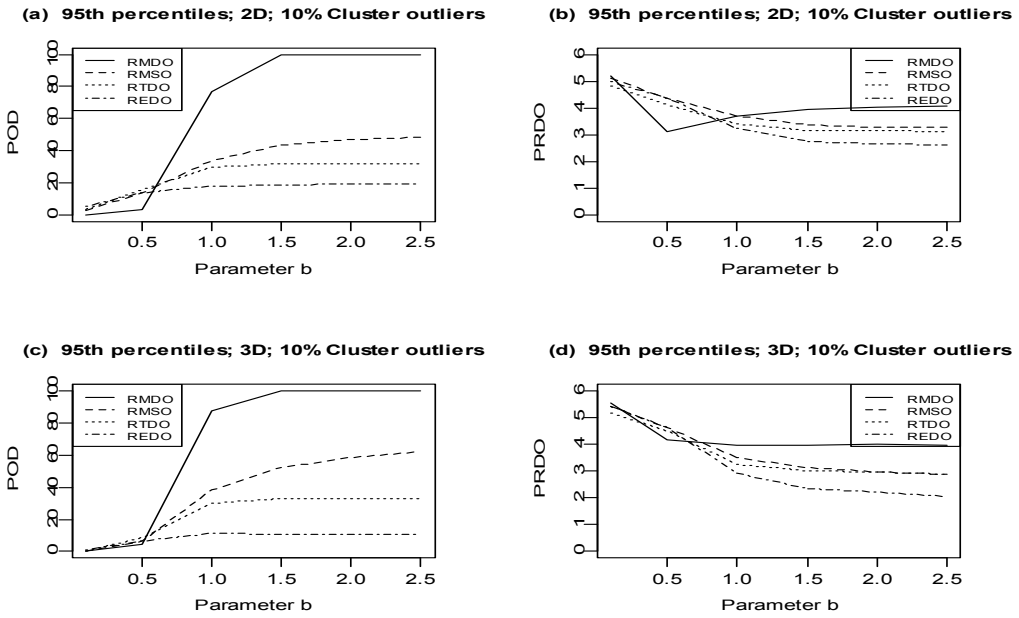


Figure 6.4: Cluster outliers. Simulation results for two-dimensional and three-dimensional data of size $n = 100$ using the 95th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD) whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

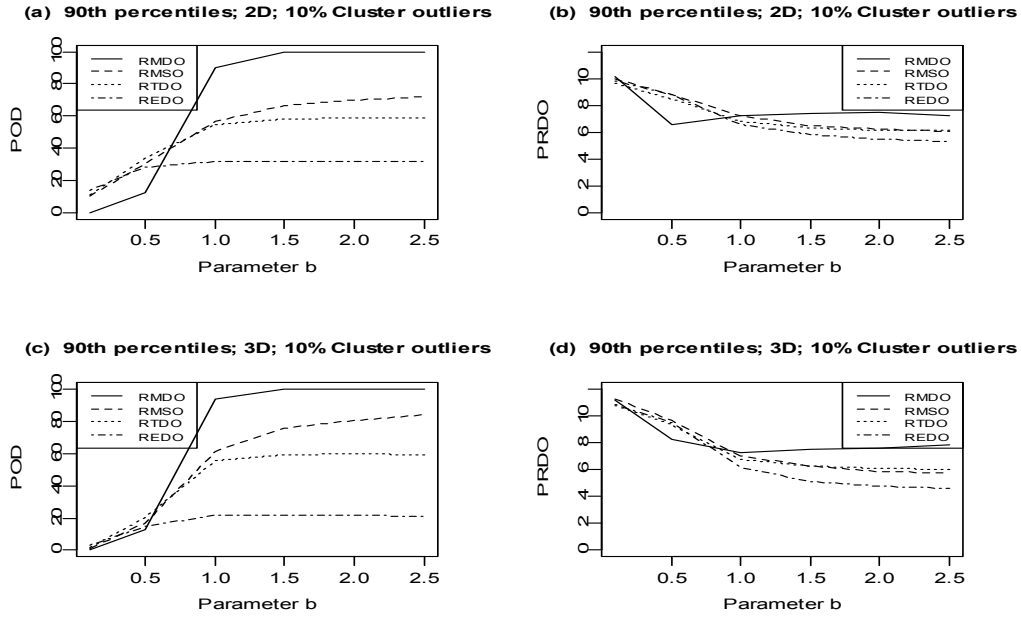


Figure 6.5: Cluster outliers. Simulation results for two-dimensional and three-dimensional data of size $n = 100$ using the 90th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD) whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

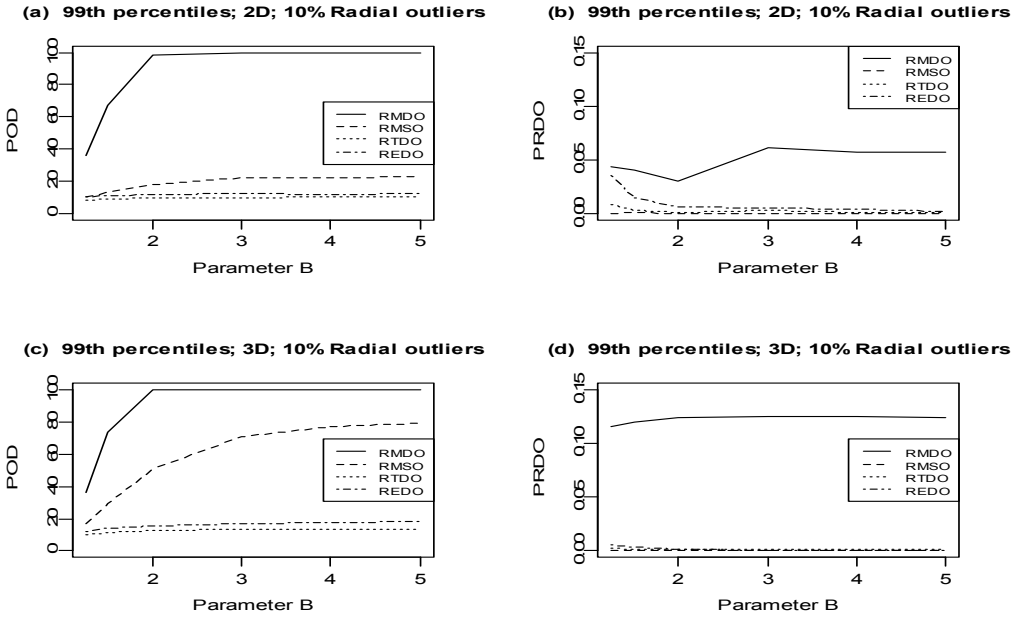


Figure 6.6: Radial outliers. Simulation results for two-dimensional and three-dimensional data of size $n = 100$ using the 99th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD) whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

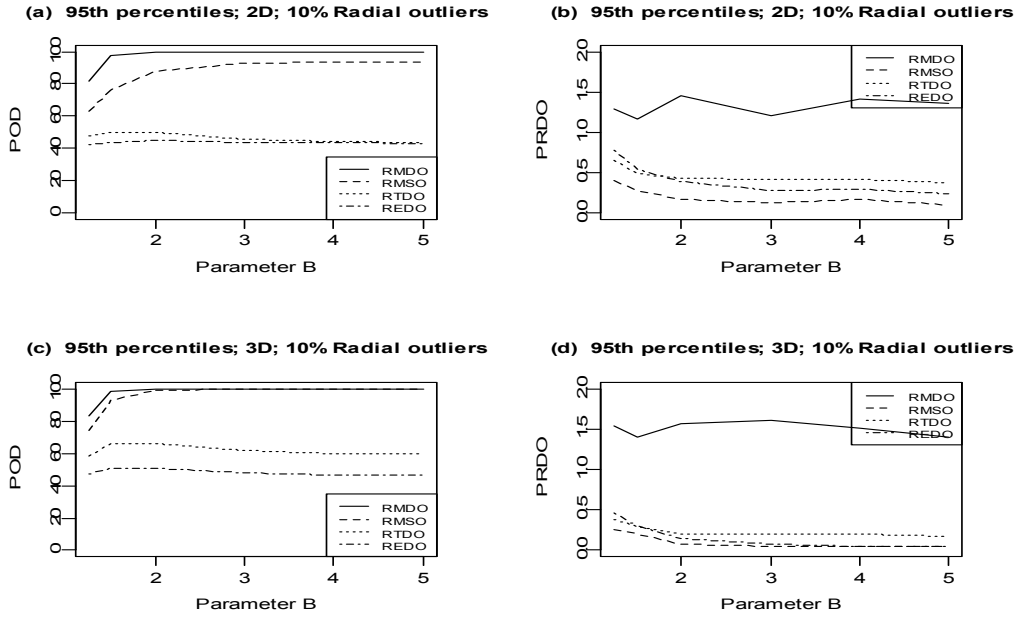


Figure 6.7: Radial outliers. Simulation results for two-dimensional and three-dimensional data of size $n = 100$ using the 95th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD) whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

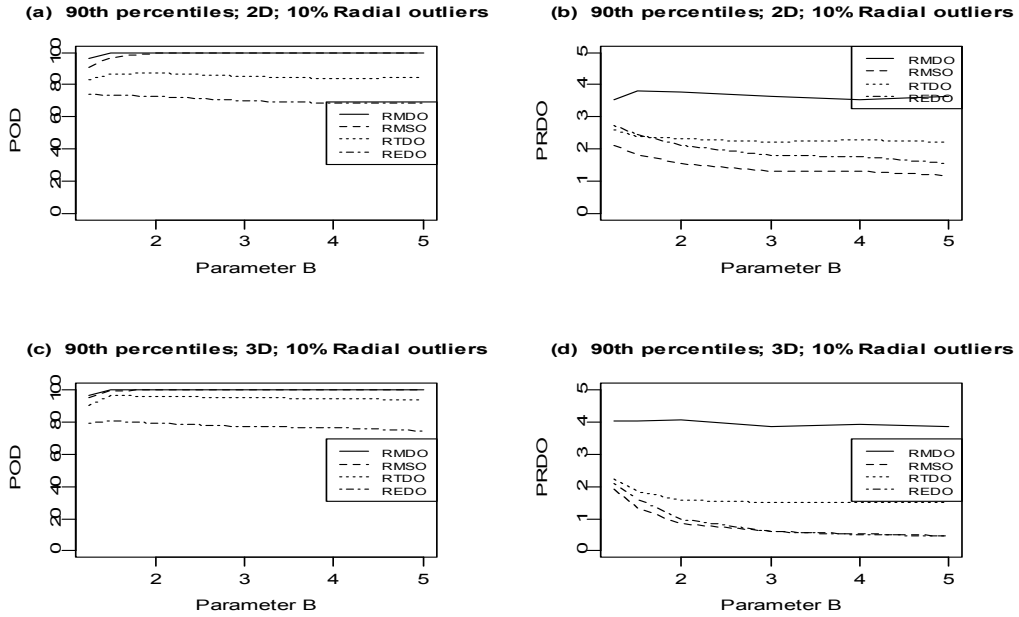


Figure 6.8: Radial outliers. Simulation results for two-dimensional and three-dimensional data of $size\ n = 100$ using the 90th percentiles of uncontaminated data outlyingness values. Panels (a) and (c) show the percentage of outliers detected (POD) whereas Panels (b) and (d) show the percentage of regular observations falsely classified as outliers (PRDO).

REFERENCES

Azzalini A., and Dalla Valle A. (1996), "The Multivariate Skew-normal Distribution," *Biometrika*, 83, 715–726.

Brys G., Hubert M., and Rousseeuw P. J. (2005), "A robustification of Independent Component Analysis," *Chemometrics*, 19, 364–375.

Campbell N. A. (1989), "Bushfire Mapping Using NOAA AVHRR Data," technical report, CSIRO.

Cerioli, A (2010), "Outlier Detection With High-breakdown Estimators," *Journal of the American Statistical Association*, 105, 147–156.

Chaudhuri, P. (1996), "On a Geometric Notion of Quantiles for Multivariate Data," *Journal of the American Statistical Association*, 91, 862–872.

Dang, X., and Serfling, R. (2006), "Nonparametric Depth-based Outlier Identifiers, and Robustness Properties," Preprint.

Dang X., and Serfling R. (2010), "Nonparametric Depth-based Multivariate Outlier Identifiers, and Masking Robustness Properties," *Journal of Statistical Planning and Inference*, 140, 198-213

Donoho D. L. (1982), "Breakdown Properties of Multivariate Location Estimators," Qualifying paper, Havard University.

Dovoedo, Y. H., and Chakraborti, S. (2010), "A Modified Adjusted Boxplot for Skewed Distributions," *American Statistical Association Proceedings of the Statistical graphics section*.

Elmore, R.T. (2005), "An Affine-invariant Data Depth Based on Random Hyperellipses," *Workshop*, Colorado State University.

Elmore, R.T., Hettmansperger T. P., and Xuan P. (2006), "Spherical Data Depth and Multivariate Median," *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, 72, 87-101.

Filzmoser P, Maronna RA, and Werner M. (2008), "Outlier Identification in High Dimensions," *Computational Statistics and Data Analysis*, 52, 1694–1711.

Hardin J, and Rocke D.M. (2005), "The Distribution of Robust Distances," *Journal of Computational and Graphical Statistics*, 14, 928–946.

Hubert, M., and Van der veeken, S., (2008), "Outlier Detection for Skewed Data," *Chemometrics*, 22, 235-246.

Hubert, M., and Vandervieren, E. (2008), "An Adjusted Boxplot for Skewed Distributions," *Computational Statistics and Data Analysis*, 52, 5186-5201.

Koltchinskii, V. (1997), "M-Estimation, Convexity and Quantiles," *Annals of Statistics*, 25, 435-477.

Liu R. Y. (1990), "On a Notion of Data Depth Based on Random Simplices," *Annals of Statistics*, 18(1), 405–414.

Liu, R. Y. (1992), "Data Depth and Multivariate Rank Tests," in: *L1-Statistics and Related Methods*, ed. Dodge Y., Amsterdam: North-Holland, pp. 279-294

Liu, Z., and Modarres, R. (2010-In review), "Triangle Data Depth and Median," *Journal of Nonparametric Statistics*.

Liu, R. Y., and Singh, K. (1993), "A Quality Index Based on Data Depth and Multivariate Rank Tests," *Journal of the American Statistical Association*, 88, 252–260.

Maronna, R. A., and Yohai, V. J. (1995), "The Behavior of the Stahel- Donoho Robust Multivariate Estimator," *Journal of the American Statistical Association*, 90, 330-341.

Rocke D. M., and Woodruff D. L. (1996), "Identification of Outliers in Multivariate Data," *Journal of the American Statistical Association*, 91, 1047–1061.

Rousseeuw P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871-880.

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

Rousseeuw, P. J., and van Zomeren, B. C. (1990), "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical Association*, 85, 633-651.

Serfling, R. (2002), "A Depth Function and a Scale Curve Based on Spatial Quantiles," in: *Statistical Data Analysis Based On the L1-Norm and Related Methods*, ed. Dodge Y., Basel: Birkhauser, pp. 25-38.

Stahel, W. A. (1981), "Robuste schatzungen: Infinitesimale Optimalitat und schatzungen von kovarianzmatrizen," *Ph.D. Thesis*, ETH, Zurich.

Tukey J. W. (1974), "Mathematics and Picturing Data," *Proceedings of the 1975 International Congress of Mathematics*, 2, 523–531.

Zuo, Y., and Serfling, R. (2000), "General Notions of Statistical Depth Function," *Annals of Statistics*, 28, 461–482.

CHAPTER 7

SUMMARY AND FUTURE RESEARCH

7.1 Summary

Chapter 1 gives a brief overview of outlier detection and Chapter 2 reviews briefly the literature on outlier detection.

In chapter 3, a boxplot procedure for location-scale distributions was proposed, in which the probability that at least one observation from a non-contaminated sample is identified as outlier, was controlled at a nominal value. An *exact expression* of that probability was derived. The proposed procedure is a modification of the boxplot outlier labeling rule by Sim, Gan and Chang, 2005. The two procedures are compared in a simulation study. The study shows that in the situations studied, the proposed procedure is comparable to the Sim, Gan and Chang, 2005 procedure, for low degrees of contamination. However, the proposed procedure is slightly more powerful when the degree of contamination is higher.

In chapter 4, boxplot-based one- and two-sided phase I control charts in which the overall false alarm rate is controlled are proposed and studied. A simulation study shows that the proposed charts are considerably more in-control robust and have out-control properties comparable to the competing charts by Jones and Champ, 2002.

In chapter 5, a modification of the adjusted boxplot for skewed distributions by Hubert and Vandervieren, 2008 is proposed and studied. The proposed boxplot fences are constructed from the median, using multiples of the semi-interquartile ranges. The multiples used in the

construction of the fences are functions of the medcouple, a robust measure of skewness introduced by Brys et al., 2003. The performances of the proposed boxplot, the adjusted boxplot and the traditional (Tukey's) boxplot are compared in a simulation study. The results based on simulated data from various distributions indicate, in most cases studied, in general, a clear advantage of the proposed procedure over the traditional boxplot with respect to (i) the percentages of outliers detected (in case of contaminated data) and (ii) with respect to the false alarm rate (in case of uncontaminated data). The results also indicate similar (but slight) advantages of the proposed procedure over the adjusted boxplot, particularly for moderately skewed distributions.

In chapter 6, an extensive simulation study is performed to compare the outlier detection performances of some robust outlier identifiers (outlyingness functions) with multivariate skew-normal data. The outlier identifiers included are, the robust Mahalanobis distance outlyingness (RMDO) based on the Mahalanobis depth function (Liu and Singh, 1993), the robust Mahalanobis spatial outlyingness (RMSO) (Chaudhuri, 1996; Dang and Serfling, 2010), the robust elliptical depth outlyingness (REDO) based on the elliptical depth function (Elmore, 2005), and the robust triangle outlyingness (RTDO) based on the triangle depth (Liu and Modarres, R 2010). Two outlier scenarios or settings were investigated, the "cluster outliers" and the "radial outliers". Based on observed results, the following recommendations are made, using certain percentiles from the uncontaminated data as critical values. For the cluster outliers scenario, we suggest using, in order of preference, the RDMO, the RMSO, and the RTDO. For the radial outliers, however, we suggest using, in order of preference, the RMSO, the RMDO, the RTDO and the REDO.

7.2 Future Research

In this dissertation, we proposed a boxplot procedure for skewed distributions. A similar approach could be used be used for heavy tailed distributions, using for example, measures of tail weights proposed in Brys et al., 2006.

More work should be done with respect to the comparison of the performance of the outlyingness functions, at outlier detection. The simulation study was performed, assuming the "clean" data were from some skew-normal distributions. The vector parameters of the skew-normal distribution should be varied to check for consistency with the results and observations reported in this dissertation. It may be interesting to study the outlier detection capability of the outlier identifiers of chapter 6 with other multivariate distributions like the multivariate t-distribution, the multivariate gamma distribution and the multivariate beta distribution. Also, in this dissertation, only some robust outlyingness functions were considered. This is because robust methods, based on high breakdown estimators are known to be computer intensive. Performance comparison should be performed for other outlyingness functions, at not only skew-normal distributions, but with other multivariate distributions mentioned above. In addition, the performance study was performed on bivariate and trivariate data; further work is needed for data with perhaps slightly higher dimensions (four and five) and for larger sample sizes.

Finally, Liu 1995 proposed three "completely nonparametric" multivariate control charts based on data depth. A comparison between those charts (using various depth functions) and their parametric counterparts is of interest and should be undertaken.

ADDITIONAL REFERENCES

Becker, C., and Gather, U. (1999), "The Masking Breakdown Point of Multivariate Outliers," *Journal of the American Statistical Association*, 94 (447), 947–955.

Becketti, S., and Gould, W. (1987), "Rangefinder Box Plots: A note," *The American Statistician*, 41 (2), 149.

Beckman, R.J., and Cook, R.D. (1983), "Outliers. . . . . .s," *Technometrics*, 25, 119–163.

Billor, N., Hadi, A. S., and Velleman, P. F. (2000), "BACON: Blocked Adaptive Computationally-Efficient Outlier Nominators," *Computational Statistics and Data Analysis*, 34, 279–298.

Brys, G., Hubert, M., and Struyf, A. (2006), "Robust measures of tail weight," *Computational Statistics and Data Analysis*, 50, 733–759.

Donoho, D. L., and Huber, P. J. (1983), "The Notion of Breakdown Point," in: *A Festschrift for Erich L. Lehmann*, eds. Bickel P. J., Doksum K. A., and Hodges J. L. Jr., Belmont: Wadworth, pp. 157-184.

Edgeworth, F. Y. (1887), "On Discordant Observations," *Philosophical Magazine*, 23(5), 364-375.

Ferguson, T. S. (1967), *Mathematical Statistics: A Decision Theoretic Approach*, New York: Academic Press.

Frigge, M., Hoaglin, D. C., and Iglewicz, B. (1989), "Some Implementations of the Boxplot," *Journal of the American Statistical Association*, 43, 50–54.

Gnanadesikan, R., and Kettenring, J. R. (1972), "Robust Estimates, Residuals, and Outlier Detection With Multiresponse Data," *Biometrics*, 28 81–124.

Goldberg, K. M., and Iglewicz, B. (1992), "Bivariate Extensions of the Boxplot," *Technometrics*, 34, 307-320.

Hadi A. S. (1992), "Identifying Multiple Outliers in Multivariate Data," *Journal of Royal Statistical Society, Series B*, 54 (3), 761-771.

Hastie T., Tibshirani R., and Friedman J. (2001), *The Elements of Statistical Learning*, New York: Springer.

Hoaglin, D. C., and Iglewicz, B. (1987), "Fine Tuning Some Resistant Rules for Outlier Labeling," *Journal of the American Statistical Association,* 82, 1147–1149.

Liu, R. Y. (1995), "Control Chart for Multivariate Processes," *Journal of American Statistical Association*, 90, 1380-1387.

Liu, R. Y., Parelius, J. M., and Singh, K. (1999), "Multivariate Analysis by Data Depth: Descriptive Statistics, Graphics and Inference," (with discussion), *Annals of Statistics*, 27, 783–858.

Mahalanobis, P. C. (1936), "On the Generalized Distance in Statistics," *Proceedings of the National Institute of Science of India*, 12 49–55.

Rousseeuw, P. J., Ruts, I, and Tukey, J. W. (1999), "The Bagplot: A Bivariate Boxplot," *The American Statistician*, 53 (4), 382-387

Serfling, R. (2008), "Survey on (Some) Nonparametric and Robust Multivariate Methods," in: *TheYear-book of the Finnish Statistical Society*, Proceedings of 2007 Conference of the Finnish Statisticians, pp. 11-41.

Serfling, R. (2009), "Equivariance and Invariance Properties of Multivariate Quantile and Related Functions, and the Role of Standardization," *Journal of Nonparametric Statistics*, 00(0), 1-22.

Singh, K. (1991), "A Notion of Majority Depth," Preprint.

Zuo, Y. (2003), "Projection-based Depth Functions and Associated medians," *Annals of Statistics*, 31, 1460–1490.

ADDITIONAL APPENDIX: R PROGRAMS USED IN SIMULATIONS

A.1 Programs for Chapter 3

```
boxpower_4<-function (nsim, sampsize, coef_l,coef_u,coef_ls,coef_us, numbout, rdist,...) {
#compares the proposed procedure to the procedure by SIM, Gan, and Chang 2005 (SGC)
#Trying to find the proportion of time the proposed procedure (resp. the SGC procedure)
#end up detecting less than or equal to 2, 3, 4 or more than 4 outliers.
#numbout=4. The number of outliers  introduced for this program is 4
#rdist is the distribution of outliers
#coef_l and coef_u come from our table 3.1
#coef_ls and coef_us come from SGC tables
#v<-0
#to count the number of times the proposed procedure detects less than 2 outliers
countp<-0
#to count the number of times the proposed procedure detects 3 outliers
countq<-0
#to count the number of times the proposed procedure detects 4 outliers
countr<-0
#to count the number of times the proposed procedure detects more than 4 outliers
counts<-0
##
#to count the number of times SGC procedure detects less than 2 outliers
countp_s<-0
#to count the number of times SGC procedure detects 3 outliers
countq_s<-0
#to count the number of times SGC procedure detects 4 outliers
countr_s<-0
#to count the number of times SGC proposed procedure detects more than 4 outliers
counts_s<-0
#this is where the outliers will come from
y<-rdist(200000,...) ##arbitrary 200000.
for (count in 1:nsim)  {
#this is the underlying distribution. Change accordingly
        x<-rexp(sampsize,1)
        #x<-rnorm(sampsize,0,1)
        #x<-rlogis(sampsize,0,1)
        x_s<-x
        x<-sort(x)
        x_s<-sort(x_s)
```

```
#
        if (sampsize%%4==0) {l=sampsize/4}
        else {l=floor(sampsize/4)+1}
        u=sampsize-l+1
        m=ceiling(sampsize/2)
        q0<-c(x[l],x[m],x[u])
#the proposed fences
        UF0<-q0[2]+coef_u*(q0[3]-q0[2])
        LF0<-q0[2]-coef_l*(q0[2]-q0[1])
#SGC Fences
        UF0_s<-q0[3]+coef_us*(q0[3]-q0[1])
        LF0_s<-q0[1]-coef_ls*(q0[3]-q0[1])
#contaminating the sample with respect to both procs
k<-0
i<-0
numb<-NA
##Here numbout is the number of ouliers that we want to introduce in the sample
        while(k<numbout & i<200000){
                i<-i+1
                if ( ((y[i]<LF0) | (y[i]>UF0))& ((y[i]<LF0_s) | (y[i]>UF0_s))  ){
            numb[i]<-y[i]
            k<-k+1
}
}
#here are the outliers
        numb<-na.omit(numb)[1:k]
#we need to make sure we introduced numbout outliers. This is returned to check.
        testnumbout=length(numb)
#sample the positions where regular observations will be replaced by outliers
        t<-sample(1:sampsize,k,replace=F)
#replace regular observations with outlers
        for (i in 1:k) {x[t[i]]<-numb[i]}
        x<-sort(x)
        x_s<-x
#recomputing the fences for the proposed procedure
        q1<-c(x[l],x[m],x[u])
        UF1<-q1[2]+coef_u*(q1[3]-q1[2])
        LF1<-q1[2]-coef_l*(q1[2]-q1[1])
#recomputing the fences for SGC procedure
        q1_s<-c(x_s[l],x_s[m],x_s[u])
        UF1_s<-q1_s[3]+coef_us*(q1_s[3]-q1_s[1])
        LF1_s<-q1_s[1]-coef_ls*(q1_s[3]-q1_s[1])
#keeping track of observations that fell beyond the fences
#with respect to the proposed procedure
        w<-cumsum((x>UF1 | x<LF1))
#keeping track of observations that fell beyond the fences
```

```
#with respect to SGC procedure
        w_s<-cumsum((x_s>UF1_s | x_s<LF1_s))
##counting the number of simuations that ends up finding at most 2 outliers (proposed
#procedure)
        if (w[sampsize]<=2) {countp<-countp+1}
##counting the number of simuations that ends up finding at most 3 outliers (proposed
#procedure)
        if (w[sampsize]==3) {countq<-countq+1}
##counting the number of simuations that ends up finding at most 4 outliers (proposed
#procedure)
        if (w[sampsize]==4) {countr<-countr+1}
##counting the number of simuations that ends up finding more than 4 outliers (proposed
#procedure)
        if (w[sampsize]>=5) {counts<-counts+1}
##counting the number of simuations that ends up finding at most 2 outliers (SGC procedure)
        if (w_s[sampsize]<=2) {countp_s<-countp_s+1}
##counting the number of simuations that ends up finding at most 3 outliers (SGC procedure)
        if (w_s[sampsize]==3) {countq_s<-countq_s+1}
##counting the number of simuations that ends up finding at most 4 outliers (SGC procedure)
        if (w_s[sampsize]==4) {countr_s<-countr_s+1}
##counting the number of simuations that ends up finding more than 4 outliers (SGC procedure)
        if (w_s[sampsize]>=5) {counts_s<-counts_s+1}
} ## end of nsim all simulations
#The respective proportions for the proposed procedure
probout1<-countp/nsim
probout2<-countq/nsim
probout3<-countr/nsim
probout4<-counts/nsim
#The respective proportions for the SGC procedure
probout1_s<-countp_s/nsim
probout2_s<-countq_s/nsim
probout3_s<-countr_s/nsim
probout4_s<-counts_s/nsim
#The results
v<-c(testnumbout,probout1,probout2,probout3,probout4,testnumbout,
probout1_s,probout2_s,probout3_s,probout4_s )
return(v)
} ## end of function.
```

A.2 Programs for Chapter 4

```
robust_one_tail_cc<-function (nsim, samp, coef_l,alpha, rdist,...) {
#compares the proposed and the Jones and Champs (2002) one-tail charts
#compares robustness for the charts
#nsim is the number of simulations
```

```
#coef_l is found in our table 4.1 of this chapter
#alpha is the overall false alarm rate
#rdist is an appropriate gamma distribution
#samp is the number of observations, for each simulation
#to hold the number of simulations that end up finding
#at least one out-of-control observation, with respect to the proposed chart
countp<-0
#to hold the number of simulations that end up finding
#at least one out-of-control observation, with respect JC chart
countq<-0
for (count in 1:nsim)  {
        x<-rdist(samp,...)
#from Proposed control chart perspective
#sorting x
        x<-sort(x)
        if (samp%%4==0) {l=samp/4}
        else {l=floor(samp/4)+1}
        u=samp-l+1
        m=ceiling(samp/2)
        q0<-c(x[l],x[m],x[u])
#the proposed chart lower control limit
        LCL<-q0[2]-coef_l*(q0[2]-q0[1])
#keeping track of observations that fell below LCL, with respect to the proposed chart
        w<-cumsum((x<LCL))
#counting the number of simulations that end up finding
#at least one out-of-control observation, with respect to the proposed chart
        if (w[samp]>=1) {countp<-countp+1}
#JC lower control limit
        LCL1<-(1-(1-alpha)^(1/(samp-1)))*mean(x)
#keeping track of observations that fell below LCL, with respect to JC chart
        w1<-cumsum((x<LCL1))
#Counting the number of simulations that end up finding
#at least one out-of-control observation with respect JC chart
        if (w1[samp]>=1) {countq<-countq+1}
} ## end of nsim simulations
#proportion of times there is at least one
#out-of-control observation with respect to the proposed chart
probout1<-countp/nsim
#proportion of times there is at least one
#out-of-control observation with respect to JC chart
probout2<-countq/nsim
#list the results
list (Prob_at_least_One_Proposed=probout1, Prob_at_least_One_JC=probout2)
} ## end of function
```

```
robust_two_tail_cc<-function (nsim, samp, coef_l,coef_u, tau, alpha, rdist,...) {
#compares the proposed, and the Jones and Champs (2002) Two-tail charts
#compares robustness for the charts
#nsim is the number of simulations
#coef_l and coef_u are found in our table 4.2
#alpha is the overall false alarm rate
#rdist is an appropriate gamma distribution
#samp is the number of observations, for each simulation
#0<tau<alpha/m (JC method requires tau)
#in simulation I use the center of the interval [0,alpha/m] for tau
#m=30 for the simulations. I used the following tau's
#alpha=0.01, alpha/m =0.000333333 , tau=0.000166667
#alpha=0.05, alpha/m =0.001666667 , tau= 0.000833333
#alpha=0.1, alpha/m = 0.003333333, tau= 0.001666667
#to hold the number of simulations that end up finding
#at least one out-of-control observation, with respect to the proposed chart
countp<-0
#to hold the number of simulations that end up finding
#at least one out-of-control observation, with respect JC chart
countq<-0
for (count in 1:nsim)  {
        x<-rdist(samp,...)
#from proposed control chart perspective
#important to sort x
        x<-sort(x)
        if (samp%%4==0) {l=samp/4}
        else {l=floor(samp/4)+1}
        u=samp-l+1
        m=ceiling(samp/2)
        q0<-c(x[l],x[m],x[u])
#the proposed chart control limits
        UCL<-q0[2]+coef_u*(q0[3]-q0[2])
        LCL<-q0[2]-coef_l*(q0[2]-q0[1])
#keeping track of obseravtions that fell beyond the limits, with respect to the proposed chart
        w<-cumsum((x>UCL | x<LCL))
##counting the number of simulations that end up finding
# at least one out-of-control observation
        if (w[samp]>=1) {countp<-countp+1}
#from JC perspective. JC control limits
        F1<-qf( 1-(alpha/samp)+tau , 2*(samp-1), 2, lower.tail = TRUE, log.p = FALSE)
        F2<-qf( tau , 2*(samp-1), 2, lower.tail = TRUE, log.p = FALSE)
        LCL1<-(samp*mean(x))/(1+(samp-1)*F1)
        UCL1<-(samp*mean(x))/(1+(samp-1)*F2)
#keeping track of observations that fell beyond the limits, with respect to JC chart
        w1<-cumsum((x<LCL1 | x>UCL1))
#counting the number of simulations that end up finding
```

```
#at least one out-of-control observation. case of the JC control chart
        if (w1[samp]>=1) {countq<-countq+1}
} ## end of nsim nsim simulations
#proportion of times there is at least one
#out-of-control observation with respect to the proposed chart
probout1<-countp/nsim
#proportion of times there is at least one
#out-of-control observation with respect to JC chart
probout2<-countq/nsim
#list the results
list (Prob_at_least_One_Proposed=probout1, Prob_at_least_One_JC=probout2)
} ## end of function
```

```
comp_perf_one_tail_cc<-function (nsim, samp,t,mu,delta,coef_l,alpha) {
#compares the performances of the one-tail Control Charts
#the one by Jones and Champ, 2002 (JC) and the proposed one
#mu is the in-control mean
#samp is the total number of samples of size 1, for each simulation
#out of samp we have t out-of control observations.
#mu is the in-control mean of the exponential distribution
#delta is the shift. So, the out-of-control mean is mu+delta*mu
#alpha is the overall false alarm rate
#coef_l is obtained from table table 4.1
#to hold the number of simulations that end up finding at least one out-of-control obs.,
#with respect to the proposed procedure
countp<-0
#to hold the number of simulations that end up finding at least one out-of-control obs.,
#with respect to the JC procedure
countq<-0
for (count in 1:nsim)  {
#the in-control observations, mean=mu
        x1<-rexp(samp-t,rate=1/mu)
#the out-of-control observations, mean=mu+delta*mu
        x2<-rexp( t,rate=1/(mu+delta*mu) )
        x<-c(x1,x2)
#the proposed chart computations
#important to sort x
        x<-sort(x)
        if (samp%%4==0) {l=samp/4}
        else {l=floor(samp/4)+1}
        u<-samp-l+1
        m<-ceiling(samp/2)
        q0<-c(x[l],x[m],x[u])
#the LCL for the proposed chart
        LCL<-q0[2]-coef_l*(q0[2]-q0[1])
```

```
#keeping track of observations that were declared out of control,
#by the proposed chart
        w<-cumsum((x<LCL))
#counting the number of simulations that end up with at least one signal
#case of the proposed control chart
        if (w[samp]>=1) {countp<-countp+1}
#Jones and Champ (2002) chart computations
#JC control chart lower control limit
        LCL1<-(1-(1-alpha)^(1/(samp-1)))*mean(x)
#keeping track of observations that were declared out of control,
#by JC one-tail chart
        w1<-cumsum((x<LCL1))
#counting the number of simulations that end up with at least one signal
#case of the JC control chart
        if (w1[samp]>=1) {countq<-countq+1}
} ## end of the nsim simulations
#proportion of times there is at least one
#out-of-control observation with respect to the proposed chart
probout1<-countp/nsim
#proportion of times there is at least one
#out-of-control observation with respect to JC chart
probout2<-countq/nsim
#The results
list (Prob_out_Proposed=probout1, Prob_out_JC=probout2)
} ## end of function.
```

```
comp_perf_two_tail_cc<-function (nsim, samp,t,mu,delta, coef_l,coef_u,tau,alpha) {
#compares the performances of the two-tail Control Charts
#the one by Jones and Champ, 2002 (JC) and the proposed one
#samp is the total number of samples of size 1
#out of samp we have t out-of control observations.
#mu is the in-control mean of the exponential distribution
#delta is the shift in the mean
#tau is a parameter needed to compute JC control limts
#tau should satisfy 0<tau<samp/alpha
#0<tau<alpha/m (JC method requires this tau)
#In simulations, I use the center of the interval [0,alpha/m] for tau
#m=30 for the simulations. I used the following tau's
#alpha=0.01, alpha/m =0.000333333 , tau=0.000166667
#alpha=0.05, alpha/m =0.001666667 , tau= 0.000833333
#alpha=0.1, alpha/m = 0.003333333, tau= 0.00166666
#alpha=0.2, alpha/m=0.006666667, tau= 0.003333333
#alpha=0.3, alpha/m=0.01, tau=0.005
#To hold the number of simulations that end up finding at least one out-of-control observation,
#with respect to the proposed two-tail chart
```

```
countp<-0
#To hold the number of simulations that end up finding at least one out-of-control observation,
#with respect to JC two-tail chart
countq<-0
for (count in 1:nsim)  {
#the in-control observations, mean=mu
        x1<-rexp(samp-t,rate=1/mu)
#the out-of-control observations, mean=mu+delta*mu
        x2<-rexp( t,rate=1/(mu+delta*mu) )
        x<-c(x1,x2)
#the proposed chart computations
#sort x
        x<-sort(x)
        if (samp%%4==0) {l=samp/4}
        else {l=floor(samp/4)+1}
        u<-samp-l+1
        m<-ceiling(samp/2)
        q0<-c(x[l],x[m],x[u])
#The control limts for the proposed two-tail chart
        UCL<-q0[2]+coef_u*(q0[3]-q0[2])
        LCL<-q0[2]-coef_l*(q0[2]-q0[1])
#keeping track of observations that were declared out of control,
#by the proposed two-tail chart
        w<-cumsum((x<LCL | x>UCL))
#counting the number of simulations that end up with at least one signal
#case of the proposed control chart
        if (w[samp]>=1) {countp<-countp+1}
#Jones and Champ chart computations.
        F1<-qf( 1-(alpha/samp)+tau , 2*(samp-1), 2, lower.tail = TRUE, log.p = FALSE)
        F2<-qf( tau , 2*(samp-1), 2, lower.tail = TRUE, log.p = FALSE)
#JC chart control limits
        LCL1<-(samp*mean(x))/(1+(samp-1)*F1)
        UCL1<-(samp*mean(x))/(1+(samp-1)*F2)
#keeping track of observations that were declared out of control,
#by JC two-tail chart
        w1<-cumsum((x<LCL1 | x>UCL1))
#counting the number of simulations that end up with at least one signal
#case of JC two-tail control chart
        if (w1[samp]>=1) {countq<-countq+1}
} ## end of the nsim simulations
#proportion of times there is at least one
#out-of-control observation with respect to the proposed chart
probout1<-countp/nsim
#proportion of times there is at least one
#out-of-control observation with respect to JC chart
probout2<-countq/nsim
```

```
#the results
list (Prob_out_Proposed=probout1, Prob_out_JC=probout2)
} ## end of function.
```

A.3 Programs for Chapter 5

```
modified_adjusted_boxplot<-function(x,a=-2,b=2,boxcol=-1,ylab="Values",title=" ") {
#this program draws the modified adjusted boxplot for
#a continuous univariate distribution
#this modified adjusted boxplot is a modification of the so-called
# adjusted boxplot of Hubert of Vandervieren, 2008
#this modified adjusted boxplot lower fence is given by LF=Q2-4exp(-2*MC)*(Q2-Q1)
#and its upper fence is given by UF=Q2+4exp(2*MC)*(Q3-Q2)
#where Q1, Q2, Q3 are sample first, second, third quartiles
#observations outside [LF,UF] are suspect observations (potential outliers)
#and are marked with a dot
#MC is the medcouple, a robust measure of skewness was introduced by Brys et al., 2004
#boxcol is the color of the box in the boxplot. The default is -1 (empty box)
#the following is a sample call
#Sample call: Modified_Adjusted_Boxplot(x,a=-2,b=2,boxcol=-1,
#ylab="Values",title="The modified Adjusted Boxplot for...")
#need to load the package robustbase to compute medcouple
#library(robustbase)
#ignore infinitevalues. can be turned if desired
#not necessary
        #x <- x[is.finite(x)==TRUE]
#computation of quartiles
        q <- quantile(x, c(0.25, 0.5, 0.75), names=F)
#the median and semi-interquartile ranges
        m<-q[2]
        siqrl <- q[2] - q[1]
        siqru <- q[3] - q[2]
#compute the medcouple
        m_c <- mc(x)
#compute the lower and upper fences
        low_cutoff<-m-4*exp(a*m_c)*siqrl
        high_cutoff<-m+4*exp(b*m_c)*siqru
#This delta is small quantity that will control how wide the boxplot will be
        delta<-0.1
#Identify the outliers
        outliers<-x[(x <low_cutoff) | (x > high_cutoff)]
#central location from which the box should be drawn
        CentLoc<-1
        xmin<-CentLoc-2*delta
        xmax<-CentLoc+2*delta
```

```
        ymin<-min(x)-5*delta
        ymax<-max(x)+0.8*delta
#marks for width of the box
        plot(c(CentLoc-delta/5,CentLoc+delta/5),c(m,m),xlab="",ylab=ylab,xaxt="n",
        xlim=c(xmin,xmax),ylim=c(ymin,ymax),pch=20,lwd=1)
        title(main=title)
        mtext( "Modified Adjusted Boxplot", side=1, line=1 )
#the number of outliers
        lout<-length(outliers)

        if (length(outliers)!=0){
                for (i in 1:lout) {
#mark the outliers
                        points(CentLoc,outliers[i], pch=20)
}
}
#draw the whiskers
#find observations that are not outliers
#one whisker goes from q3 to the largest observation not outlier
#the other goes from q1 the smallest observation not outlier
        Notoutliers<-x[x<=high_cutoff & x>=low_cutoff]
#drawing the upper whisker
        lines(c(CentLoc,CentLoc),c(q[3],max(Notoutliers)),lty=2)
#drawing the lower whisker
        lines(c(CentLoc,CentLoc),c(q[1],min(Notoutliers)),lty=2)
#put some horizontal lines at the end of the whiskers
        lines( c(CentLoc-delta/7, CentLoc+delta/7),c(max(Notoutliers),max(Notoutliers)) )
        lines( c(CentLoc-delta/7,CentLoc+delta/7),c(min(Notoutliers),min(Notoutliers)) )
#draw the box
        polygon(c(CentLoc-delta/5,CentLoc+delta/5,CentLoc+delta/5,CentLoc-delta/5),
        c(q[1],q[1],q[3],q[3]),col=boxcol,border=1,lwd=2)
#draw the median line
        lines(c(CentLoc-eps/5,CentLoc+eps/5),c(m,m),col=1,lwd=2)
        text(xmin+delta/4,ymax-delta/4, "MC =")
        text(xmin+0.7*delta,ymax-delta/4, round(m_c,3))
} #end of function
```

```
no_cont_boxpower<-function (nsim, sampsize, coef_a,coef_b, rdist,...) {
#for all the distributions to explore except Gg
#compute false alarm rates according to the three boxplot procedures
#no contamination
#for our boxplot, we use coef_a= -2 and coef_b= 2
v<-0
#to hold the percents of outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours<-0
```

```
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#now the distrib of outliers is normal stdv=4, mean=2. Vary this later
y<-rnorm(10000,2,4) ##arbitrary 10000.# Where outliers are going to come from
for (count in 1:nsim)  {
        x<-rdist(sampsize,...)
        x<-sort(x)
#computing the quartiles
        q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
##computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#no contamination in this simulation
        x_s<-x
q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#the fences do not change
        LF_Hub_1<-LF_Hub
        UF_Hub_1<-UF_Hub
#
        LF_Ours_1<-LF_Ours
        UF_Ours_1<-UF_Ours
#
        LF_Trad_1<-LF_Trad
        UF_Trad_1<-UF_Trad
#keeping track of observations declared as outliers
#(both tails, lower and upper tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails, lower and upper tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
```

```
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails, lower and upper tails) using our procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#percent of observations declared as outliers (in both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#percent of observations declared as outliers (in the lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsize
#percent of observations declared as outliers (in the upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#report the results
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low ,
OurPercent_up, HubPercent_up,TradPercent_up)
return(v)

} ## end of function
```

```
no_cont_boxpower_Gg<-function (nsim, sampsize, coef_a,coef_b, g) {
#for the Gg distributions I want to explore
```

```
#compute false alarm rates according to the three boxplot procedures
#no contamination simulations
#for our boxplot, we use coef_a= -2 and coef_b= 2
#to report results
v<-0
#to hold the percents of outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours<-0
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#now the distrib of outliers is normal stdv=4, mean=2, Vary this later
# where outliers are going to come from
y<-rnorm(10000,2,4) ##arbitrary 10000.
for (count in 1:nsim)  {
        x1<-rnorm(sampsize,0,1)
##the regular observations
        x<-(exp(g*x1)-1)/g # x follow the Gg Distribution.
        x<-sort(x)
#computing the quartiles
        q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
##computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#no contamination
        x_s<-x
        q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#the fences do not change
        LF_Hub_1<-LF_Hub
        UF_Hub_1<-UF_Hub
#
        LF_Ours_1<-LF_Ours
        UF_Ours_1<-UF_Ours
#
```

```
        LF_Trad_1<-LF_Trad
        UF_Trad_1<-UF_Trad
#keeping track of observations declared as outliers
#(both tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#percent of observations declared as outliers (in both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#percent of observations declared as outliers (in the lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsize
#percent of observations declared as outliers (in the upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#report the results
```

```
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low ,
OurPercent_up, HubPercent_up,TradPercent_up)
return(v)
} ## end of function
```

```
left_cont_boxpower<-function (nsim, sampsize, coef_a,coef_b, rdist,...) {
#for all the distributions to explore except Gg
#5% lower contamination
#find percent of outliers detected by the three boxplot procedures
#for our boxplot, we use coef_a= -2 and coef_b= 2
#to hold the percents of outliers from different perpectives (ours, HV's, Traditional)
v<-0
Percent_Out_Ours<-0
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#to hold the outliers
outliers<-0
#now the distrib of outliers is normal stdv=4, mean=0, Vary this later
# where outliers are going to come from
y<-rnorm(1000000,0,4) ## arbitrary 100000.
for (count in 1:nsim)  {
#this the distribution of regular observations
        x<-rdist(sampsize,...)
        x<-sort(x)
#computing the quartiles
        q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
##computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#contamination with lower outliers.
```

```
        k<-0
        i<-0
#to hold the outliers
        numb<-NA
#number of outliers
        numbout<-(0.05)*sampsize
##storing the outliers in numb
        while(k<numbout & i<1000000){
                i<-i+1
#the observation has to fall below all three lower fences to be
#considered outlier
                if ( (y[i]<LF_Hub) & (y[i]<LF_Ours) & (y[i]<LF_Trad) ) {
            numb[i]<-y[i]
            k<-k+1
}
}
#removing the NA's from the vector holding the outliers
        numb<-na.omit(numb)[1:k]
#return this to check the number of outliers
        test_numbout<-length(numb)
#sample the positions where the regular observations will be replaced with outliers
        t<-sample(1:sampsize,k,replace=F)
#replacing regular observations with outliers
        for (i in 1:k) {x[t[i]]<-numb[i]}
        x<-sort(x)
        x_s<-x
#the quartiles
        q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#computing the fences after the contamination (HV, ours, Traditional)
        LF_Hub_1<-q1[1]-1.5*exp(-4*mc(x_s))*(q1[3]-q1[1])
        UF_Hub_1<-q1[3]+1.5*exp(3*mc(x_s))*(q1[3]-q1[1])
#
        LF_Ours_1<-q1[2]-4*exp(coef_a*mc(x_s))*(q1[2]-q1[1])
        UF_Ours_1<-q1[2]+4*exp(coef_b*mc(x_s))*(q1[3]-q1[2])
#
        LF_Trad_1<-q1[1]-1.5*(q1[3]-q1[1])
        UF_Trad_1<-q1[3]+1.5*(q1[3]-q1[1])
#keeping track of observations declared as outliers
#(both tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
```

```
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails) using the traditional procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#check the number of outliers
        outliers[count]<-test_numbout
#percent of observations declared as outliers (both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#Percent of observations declared as outliers (lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsize
#Percent of observations declared as outliers (upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#checking the number of outliers
The_Outliers<-mean(outliers)
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low,
OurPercent_up, HubPercent_up,TradPercent_up,
The_Outliers)
return(v)
} ## end of function
```

```
left_cont_boxpower_Gg<-function (nsim, sampsize, coef_a,coef_b, g) {
#for the Gg distributions I want to explore
#5% left contamination
#find percents of outliers detected by the three boxplot procedures
v<-0
#to hold the percents of outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours<-0
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#to hold the outliers
outliers<-0
#
#now the distrib of outliers is normal stdv=4, mean=0, Vary this later
.# Where outliers are going to come from
y<-rnorm(10000,0,4) ##arbitrary 10000
for (count in 1:nsim)  {
        x1<-rnorm(sampsize,0,1)
#the regular observations
        x<-(exp(g*x1)-1)/g
# x follow the Gg Distribution.
        x<-sort(x)
#computing the quartiles
        q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
##computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#contamination with lower outliers
        k<-0
        i<-0
        numb<-NA
        numbout<-(0.05)*sampsize
```

```
#here numbout is the number of ouliers that we want to introduce in the sample
        while(k<numbout & i<200000){
                i<-i+1
#an observation has to fall below all three lower fences to be
#considered outlier
                if ( (y[i]<LF_Hub) & (y[i]<LF_Ours) & (y[i]<LF_Trad) ) {
            numb[i]<-y[i]
            k<-k+1
}
}
#removing the NA's from the vector holding the outliers
        numb<-na.omit(numb)[1:k]
#return this to check the number of outliers
        test_numbout<-length(numb)
#sample the positions where the regular observations will be replaced with outliers
        t<-sample(1:sampsize,k,replace=F)
#replacing regular observations with outliers
        for (i in 1:k) {x[t[i]]<-numb[i]}
        x<-sort(x)
        x_s<-x
#the quartiles
        q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#computing the fences after the contamination (HV, ours, Traditional)
        LF_Hub_1<-q1[1]-1.5*exp(-4*mc(x_s))*(q1[3]-q1[1])
        UF_Hub_1<-q1[3]+1.5*exp(3*mc(x_s))*(q1[3]-q1[1])
#
        LF_Ours_1<-q1[2]-4*exp(coef_a*mc(x_s))*(q1[2]-q1[1])
        UF_Ours_1<-q1[2]+4*exp(coef_b*mc(x_s))*(q1[3]-q1[2])
#
        LF_Trad_1<-q1[1]-1.5*(q1[3]-q1[1])
        UF_Trad_1<-q1[3]+1.5*(q1[3]-q1[1])
#
#keeping track of observations declared as outliers
#(both tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails) using the traditional procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
```

```
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#check the number of outliers, to make sure
        outliers[count]<-test_numbout
#percent of observations declared as outliers (in both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#percent of observations declared as outliers (in the lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsiz
#percent of observations declared as outliers (in the upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#checking the number of outliers
The_Outliers<-mean(outliers)
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low,
OurPercent_up, HubPercent_up,TradPercent_up,
The_Outliers)
return(v)
} ## end of function
```

```
right_cont_boxpower<-function (nsim, sampsize, coef_a,coef_b, rdist,...) {
#for all distributions to explore except Gg
#for our boxplot, we use coef_a= -2 and coef_b= 2
#rdist is the distribution of regular observations
```

```
#5% right contamination
#find percents of outliers detected by the three boxplot procedures
v<-0
#to hold the percents of outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours<-0
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#to hold the outliers
outliers<-0
#now the distrib of outliers is normal stdv=4, mean=2, Vary this later
# where outliers are going to come from
y<-rnorm(10000,0,4) ##arbitrary 10000.
for (count in 1:nsim)  {
#the regular observations.
        x<-rdist(sampsize,...)
        x<-sort(x)
#computing the quartiles
        q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
##computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#contamination with with upper outliers
        k<-0
        i<-0
        numb<-NA
        numbout<-(0.05)*sampsize
#here numbout is the number of ouliers that we want to introduce in the sample
        while(k<numbout & i<10000){
                i<-i+1
#the observation has to fall above all three upper fences to be
#considered outlier
                if ( (y[i]>UF_Hub) & (y[i]>UF_Ours) & (y[i]>UF_Trad) ) {
```

```
            numb[i]<-y[i]
            k<-k+1
}
}
#removing the NA's from the vector holding the outliers
        numb<-na.omit(numb)[1:k]
#return this to check the number of outliers
        test_numbout<-length(numb)
#sample the positions where the regular observations will be replaced with outliers
        t<-sample(1:sampsize,k,replace=F)
#replacing some regular observations with outliers
        for (i in 1:k) {x[t[i]]<-numb[i]}
        x<-sort(x)
        x_s<-x
#the quartiles after contamination
        q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#computing the fences after the contamination (HV, ours, Traditional)
        LF_Hub_1<-q1[1]-1.5*exp(-4*mc(x_s))*(q1[3]-q1[1])
        UF_Hub_1<-q1[3]+1.5*exp(3*mc(x_s))*(q1[3]-q1[1])
#
        LF_Ours_1<-q1[2]-4*exp(coef_a*mc(x_s))*(q1[2]-q1[1])
        UF_Ours_1<-q1[2]+4*exp(coef_b*mc(x_s))*(q1[3]-q1[2])
#
        LF_Trad_1<-q1[1]-1.5*(q1[3]-q1[1])
        UF_Trad_1<-q1[3]+1.5*(q1[3]-q1[1])
#keeping track of observations declared as outliers
#(both tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails) using the traditional procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#check the number of outliers
        outliers[count]<-test_numbout
#percent of observations declared as outliers (in both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
```

```
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#percent of observations declared as outliers (in the lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsize
#percent of observations declared as outliers (in the upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#check the number of outliers
The_Outliers<-mean(outliers)
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low,
OurPercent_up, HubPercent_up,TradPercent_up,
The_Outliers)
return(v)
} ## end of function.
```

```
right_cont_boxpower_Gg<-function (nsim, sampsize, coef_a,coef_b, g) {
#for our boxplot, we use coef_a= -2 and coef_b= 2
#for Gg distributions explored
#find percents of outliers detected by the three boxplot procedures
#to return the results
v<-0
#to hold the percents of outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours<-0
Percent_Out_Hub<-0
Percent_Out_Trad<-0
#to hold the percents of lower outliers from different perpectives (ours, HV's, Traditional)
```

```
Percent_Out_Ours_low<-0
Percent_Out_Hub_low<-0
Percent_Out_Trad_low<-0
#to hold the percents of upper outliers from different perpectives (ours, HV's, Traditional)
Percent_Out_Ours_up<-0
Percent_Out_Hub_up<-0
Percent_Out_Trad_up<-0
#to hold the outliers
outliers<-0
#now the distrib of outliers is normal stdv=4, mean=2, Vary this later
#where outliers are going to come from
y<-rnorm(20000,0,4) ##arbitrary 20000
for (count in 1:nsim)  {
        x1<-rnorm(sampsize,0,1)
#the regular observations
# x from the Gg Distribution.
        x<-(exp(g*x1)-1)/g
        x<-sort(x)
#computing the quartiles
                q0<-quantile(x,probs=c(0.25,0.5,0.75),names=F)
#computing the fences (HV)
        LF_Hub<-q0[1]-1.5*exp(-4*mc(x))*(q0[3]-q0[1])
        UF_Hub<-q0[3]+1.5*exp(3*mc(x))*(q0[3]-q0[1])
#computing the fences (Ours)
        LF_Ours<-q0[2]-4*exp(coef_a*mc(x))*(q0[2]-q0[1])
        UF_Ours<-q0[2]+4*exp(coef_b*mc(x))*(q0[3]-q0[2])
#computing the fences (Traditional)
        LF_Trad<-q0[1]-1.5*(q0[3]-q0[1])
        UF_Trad<-q0[3]+1.5*(q0[3]-q0[1])
#contamination with upper outliers.
        k<-0
        i<-0
#to hold the outliers
        numb<-NA
#number of outliers
        numbout<-(0.05)*sampsize
#here numbout is the number of ouliers that we want to introduce
        while(k<numbout & i<20000){
                i<-i+1
#an observation has to fall above all three upper fences to be
#considered outlier
                if ( (y[i]>UF_Hub) & (y[i]>UF_Ours) & (y[i]>UF_Trad) ) {
            numb[i]<-y[i]
            k<-k+1
}
}
```

```
#removing the NA's from the vector holding the outliers
        numb<-na.omit(numb)[1:k]
#return this to check the number of outliers
        test_numbout<-length(numb)
#sample the positions where the regular observations will be replaced with outliers
        t<-sample(1:sampsize,k,replace=F)
#replacing regular observations with outliers
        for (i in 1:k) {x[t[i]]<-numb[i]}
        x<-sort(x)
        x_s<-x
#the quartiles
        q1<-quantile(x_s,probs=c(0.25,0.5,0.75),names=F)
#computing the fences after the contamination (HV, ours, Traditional)
        LF_Hub_1<-q1[1]-1.5*exp(-4*mc(x_s))*(q1[3]-q1[1])
        UF_Hub_1<-q1[3]+1.5*exp(3*mc(x_s))*(q1[3]-q1[1])
#
        LF_Ours_1<-q1[2]-4*exp(coef_a*mc(x_s))*(q1[2]-q1[1])
        UF_Ours_1<-q1[2]+4*exp(coef_b*mc(x_s))*(q1[3]-q1[2])
#
        LF_Trad_1<-q1[1]-1.5*(q1[3]-q1[1])
        UF_Trad_1<-q1[3]+1.5*(q1[3]-q1[1])
#keeping track of observations declared as outliers
#(both tails) using HV procedure
        w_Hub<-cumsum((x_s>UF_Hub_1 | x_s<LF_Hub_1))
        w_Hub_low<-cumsum((x_s<LF_Hub_1))
        w_Hub_up<-cumsum((x_s>UF_Hub_1))
#keeping track of observations declared as outliers
#(both tails) using our procedure
        w_Ours<-cumsum((x_s>UF_Ours_1 | x_s<LF_Ours_1))
        w_Ours_low<-cumsum((x_s<LF_Ours_1))
        w_Ours_up<-cumsum((x_s>UF_Ours_1))
#keeping track of observations declared as outliers
#(both tails) using the traditional procedure
        w_Trad<-cumsum((x_s>UF_Trad_1 | x_s<LF_Trad_1))
        w_Trad_low<-cumsum((x_s<LF_Trad_1))
        w_Trad_up<-cumsum((x_s>UF_Trad_1))
#check the number of outliers
outliers[count]<-test_numbout
#percent of observations declared as outliers (in both tails)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours[count]<-w_Ours[sampsize]/sampsize
        Percent_Out_Hub[count]<-w_Hub[sampsize]/sampsize
        Percent_Out_Trad[count]<-w_Trad[sampsize]/sampsize
#percent of observations declared as outliers (in the lower tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_low[count]<-w_Ours_low[sampsize]/sampsize
```

```
        Percent_Out_Hub_low[count]<-w_Hub_low[sampsize]/sampsize
        Percent_Out_Trad_low[count]<-w_Trad_low[sampsize]/sampsize
#Percent of observations declared as outliers (in the upper tail)
#by the procedures (Ours, HV, traditional)
        Percent_Out_Ours_up[count]<-w_Ours_up[sampsize]/sampsize
        Percent_Out_Hub_up[count]<-w_Hub_up[sampsize]/sampsize
        Percent_Out_Trad_up[count]<-w_Trad_up[sampsize]/sampsize
} ## end of nsim simulations
#check the number of outliers
The_Outliers<-mean(outliers)
#results both fences
OurPercent<-mean(Percent_Out_Ours)*100
HubPercent<-mean(Percent_Out_Hub)*100
TradPercent<-mean(Percent_Out_Trad)*100
#results Lower fences
OurPercent_low<-mean(Percent_Out_Ours_low)*100
HubPercent_low<-mean(Percent_Out_Hub_low)*100
TradPercent_low<-mean(Percent_Out_Trad_low)*100
#results Upper fences
OurPercent_up<-mean(Percent_Out_Ours_up)*100
HubPercent_up<-mean(Percent_Out_Hub_up)*100
TradPercent_up<-mean(Percent_Out_Trad_up)*100
#
v<-c( OurPercent, HubPercent, TradPercent ,
OurPercent_low, HubPercent_low, TradPercent_low ,
OurPercent_up, HubPercent_up,TradPercent_up ,
The_Outliers)
return(v)
} ## end of function
```

A.3 Programs for Chapter 6

```
rob_mahal_depth_out<-function(data){
#this program computes the robust (MCD based) Mahalanobis Distance outlyingness (CMDO)
#for each point in the dataset "data" with respect to the data
#this is where to store the RMD Outlyingness of each point
RMDO<-0
#compute the robust (MCD) covariance matrix and mean for the data
#package "MASS" is required
#Store MCD estimators
A<-cov.mcd(data)
sigma<-A$cov #MCD covariance matrix
mu<-A$center #the mean vector
#compute the outlyingness of each point and store them
```

```
n<-nrow(data)
for (i in 1:n){
        x<-data[i,]
        Maha_x_squared<-mahalanobis(x,center=mu, cov=sigma,inverted=FALSE)
        Maha_x<-sqrt(Maha_x_squared)
        RMDO[i]<-(Maha_x)/(1+Maha_x)
}
return(RMDO)
}
```

```
L2norm<-function(x){
#needed to compute the expected sign vector of data
#computes the L2 norm of a vector
r<-sqrt(sum(x^2))
return(r)
}
```

```
L2norm_matrix<-function(X){
#compute the L2norm of each row (vector) in a data matrix
#Require the function "L2norm" i wrote
n<-nrow(X)
m<-ncol(X)
X<-as.matrix(X)
#Where to store the norm of each row
result<-0
for (i in 1:n){
        result[i]<-L2norm(X[i,])
}
return(result)
}
```

```
expected_sign_vector<-function(X){
#this program compute the expected value of the sign vectors in data X
#requires the function "L2norm"
#useful for computing the spatial depth
n<-nrow(X)
m<-ncol(X)
X<-as.matrix(X)
#standardize the rows of X (the vectors) and put the result in X1
X1<-matrix(nrow=n,ncol=m)
for (i in 1:n){
#In case a row vector is zero
        if (L2norm(X[i,])==0) {
```

```
                x<-rep(0,m)
}
        if (L2norm(X[i,])!=0) {
#Standardize the rows of X
                x<-X[i,]/L2norm(X[i,])
}
        X1[i,]<-x # Put them in X1
}
#now compute the average of the standardized row vectors
X2<-0
for (i in 1:n){
        X2<-X2+X1[i,]
}
#the expected value desired is AvStdRowVectors
AvStdRowVectors<-X2/n
return(AvStdRowVectors)
}
```

```
rob_mahal_spat_depth_out<-function(data){
#This program computes the robust Mahalanobis spatial depth outlyingness (RMSO)
#for each point in the dataset "data" with respect to the data.
#according to the definition of Dang and Serfling (2010)
#this function makes use the functions "expected_sign_vector" and "L2norm"
#requires the package "MASS" if we use "ginv" to compute the inverse of a matrix
#package "MASS" is required for cov.mcd
#requires the package "car" if we use "inv" to compute the inverse of a matrix
#requires the package "mgcv" for matrix square root (function mroot), defined as B such that
#B'B=A
#this is where to store the outlyingness value of each point
RMSO<-0
#now, compute the outlyingness of each point and store them
n<-nrow(data)
A<-cov.mcd(data)
# MCD covariance matrix
sigma<-A$cov
#accoring to the formula in Dang and Serfling,2010,
#I need to take the covariance matrix at power -1/2
#square root the covariance matrix
B<-mroot(sigma)
# take the inverse of the resulting matrix
B1<-ginv(B)
#important to transform this to matrix for computation
data1<-as.matrix(data)
for (i in 1:n){
        x<-data1[i,]
```

```
#I need to compute x-data. Be careful.Use t(data)
        C<-x-t(data1)
        TransProd<-B1%*%C
#transformed data
        Product<-t(TransProd)
#can now compute the expected sign vector of this new data
        Exp_S_Vector<-expected_sign_vector(Product)
        RMSO[i]<-L2norm(Exp_S_Vector)
}
return(RMSO)
}
```

```
rob_mahal_triangle_depth_out<-function(data){
#this program computes the robust triangle depth outlyingness (RTDO)
#according to the definition of Liu and Modarres (In press-2010)
#it returns the triangle depth outlyingness of each point
#in a data cloud with respect to the data
#to store the depth
TD<-0
#to store the outlyingnesses
TDO<-0
n<-nrow(data)
#cov.mcd requires the package "MASS"
A<-cov.mcd(data)
# MCD covariance matrix
sigma<-A$cov
#need to compute the mahalanobis distance of pairs of points
#requires the R package "MASS"
# take the inverse of the resulting matrix
B<-ginv(sigma)
#important to transform this to matrix for computation
data1<-as.matrix(data)
for (k in 1:n){
#Put counter at zero for each data point
        count<-0
        xk<-data1[k,]
        for (i in 1:n) {
        for (j in 1:i) {
                xi<-data1[i,]
                xj<-data1[j,]
                P1<-xi-xj
                P2<-xk-xi
                P3<-xk-xj
#computation of mahalanobis distances between pairs of points
                D1<-t(P1)%*%B%*%P1
```

```
                D2<-t(P2)%*%B%*%P2
                D3<-t(P3)%*%B%*%P3
                if ( D1>max(D2,D3) ) {count<-count+1}
}
}
TD[k]<-count/(n*(n-1)/2)
}
#now compute outlyingnesses using the depths
TDO<-1-TD
return(TDO)
}
```

```
rob_mahal_Ellipse_depth_out<-function(data){
#this program computes the robust triangle depth outlyingness (REDO)
#according to the definition of Elmore (2005)
#it returns the elliptical depth of each point in a data cloud with respect to the data
#to store the depths
ED<-0
#to store the outlyingnesses
EDO<-0
n<-nrow(data)
#cov.mcd requires the package "MASS"
A<-cov.mcd(data)
# MCD covariance matrix
sigma<-A$cov
#needs to compute the mahalanobis distance of pairs of points
#requires package "MASS"
# take the inverse of the resulting matrix. B=sigma^-1
B<-ginv(sigma)
#important to transform this to matrix for computation
data1<-as.matrix(data)
for (k in 1:n){
#put counter at zero for each data point
        count<-0
#pick up the kth data point
        xk<-data1[k,]
        for (i in 1:n) {
        for (j in 1:i) {
                xi<-data1[i,]
                xj<-data1[j,]
                P1<-xi-xk
                P2<-xj-xk
#mahalanobis type computation
                D<-t(P1)%*%B%*%P2
                if ( D<=0 ) {count<-count+1}
```

```
}
}
ED[k]<-count/(n*(n-1)/2)
}
#now compute outlyingnesses using the depths
EDO<-1-ED
return(EDO)
}
```

```
find_percentiles<-function (dim,nsim, sampsize) {
#compute percentiles of outlyingness values for "clean data" from the multivariate skew-normal
distribution of Azzalini and Dalla Valle (1996)
#only values 2, 3, 5 are allowed for dim in this program
#alpha a parameter of the skew-nomal distribution (vector of length p=dim)
#dim is the dimension of the data (number of columns). Can take only values 2,3,5
#this program makes use of the following programs
#make sure to have them available
#rob_mahal_depth_out()
#rob_mahal_Spat_depth_out()
#rob_mahal_Triangle_depth_out()
#rob_mahal_Ellipse_depth_out()
#the following R packages are required for this main program to run.
#"mnormt","sn","MASS","mgcv","robustbase","mvtnorm","depth","ExPD2D","car".
#library(mnormt)
#library(sn)
#library(MASS)
#library(mgcv)
#library(robustbase)
#library(mvtnorm)
#library(car)
#to store percentiles of outlyingnesses for each run
#case of robust mahalanobis distance outlyingness
RMDO_80<-0
RMDO_90<-0
RMDO_95<-0
RMDO_99<-0
#case of robust mahalanobis spatial distance outlyingness
RMSO_80<-0
RMSO_90<-0
RMSO_95<-0
RMSO_99<-0
#case of robust triangle deph outlyingness
RTDO_80<-0
RTDO_90<-0
RTDO_95<-0
```

```
RTDO_99<-0
#case of robust elliptical deph outlyingness
REDO_80<-0
REDO_90<-0
REDO_95<-0
REDO_99<-0
#parameters in generating skew-normal variates
Omega<-diag(dim)
mu<-rep(0,dim)
if (dim==2) { alpha<-c(10,4) }
else if (dim==3) { alpha<-c(rep(10,1),rep(4,2)) }
else if (dim==5) { alpha<-c(rep(10,2),rep(4,3)) }
# Number of observations (regular at start)
n<-sampsize
for (count in 1:nsim)  {
#generating regular observations
#x1 is a matrix with n rows and p columns. p-variate skew-normal observation
#package "sn" needed to generate multivariate skew-normal
        x1<-rmsn(n, mu, Omega, alpha)[1:n,]
#If we want to generate from MVN
#x1<-rmvnorm(n, mean=mu, sigma=Omega)
        x2<-rep(0,n)
#I tag the regular observations with zeros
        x<-cbind(x1,x2)
#Z has sampsize rows and dim+1 columns
# no outliers here
        Z<-x
#Need to compute "outlyingness" of each point with respect to the data now
#grab the first "dim" columns of Z. They will be used for outlyingness computations
        Z1<-Z[,1:dim]
#compute Robust Mahalanobis "outlyingness" with MCD (RMDO)of observations
#use first "dim" columns of Z and store results in column (dim+2)
#vector of length sampsize=n
        RMDO<-rob_mahal_depth_out(Z1)
        Z<-cbind(Z,RMDO) #add a column to Z
        A1<-quantile(RMDO,probs=c(0.80,0.90),type=6,names=F)
        A<-quantile(RMDO,probs=c(0.95,0.99),type=6,names=F)
        RMDO_80[count]<-A1[1]
        RMDO_90[count]<-A1[2]
        RMDO_95[count]<-A[1]
        RMDO_99[count]<-A[2]
#compute Robust Mahal Spacial "outlyingness" with MCD (RMSO) of observations
#use first "dim" columns of Z and store results in column (dim+3)
        RMSO<-rob_mahal_Spat_depth_out(Z1)
#add a column to Z, to hold the RMSO's
        Z<-cbind(Z,RMSO)
```

```
        A1<-quantile( RMSO ,probs=c(0.80,0.90),type=6,names=F)
        A<-quantile( RMSO ,probs=c(0.95,0.99),type=6,names=F)
        RMSO_80[count]<-A1[1]
        RMSO_90[count]<-A1[2]
        RMSO_95[count]<-A[1]
        RMSO_99[count]<-A[2]
#compute Robust (MCD) Mahalanobis Triangle depth outlyingness(RTDO) of observations
#use first "dim" columns of Z and store results in column (dim+4)
        RTDO<-rob_mahal_Triangle_depth_out(Z1)
#add a column to Z, to hold the RTDO's
        Z<-cbind(Z,RTDO)
        A1<-quantile( RTDO ,probs=c(0.80,0.90),type=6,names=F)
        A<-quantile( RTDO ,probs=c(0.95,0.99),type=6,names=F)
        RTDO_80[count]<-A1[1]
        RTDO_90[count]<-A1[2]
        RTDO_95[count]<-A[1]
        RTDO_99[count]<-A[2]
#compute Robust (MCD) Elliptical Triangle depth outlyingness(REDO) of observations
#use first "dim" columns of Z and store results in column (dim+5)
        REDO<-rob_mahal_Ellipse_depth_out(Z1)
#add a column to Z, to hold the REDO's
        Z<-cbind(Z,REDO)
        A1<-quantile( REDO ,probs=c(0.80,0.90),type=6,names=F)
        A<-quantile( REDO ,probs=c(0.95,0.99),type=6,names=F)
        REDO_80[count]<-A1[1]
        REDO_90[count]<-A1[2]
        REDO_95[count]<-A[1]
        REDO_99[count]<-A[2]
} ## end of nsim simulations
#compute averages of percentiles of outlyingness values
#for each outlyingness function
RMDO_80_av<-mean(RMDO_80)
RMDO_90_av<-mean(RMDO_90)
RMDO_95_av<-mean(RMDO_95)
RMDO_99_av<-mean(RMDO_99)
#
RMSO_80_av<-mean(RMSO_80)
RMSO_90_av<-mean(RMSO_90)
RMSO_95_av<-mean(RMSO_95)
RMSO_99_av<-mean(RMSO_99)
#
RTDO_80_av<-mean(RTDO_80)
RTDO_90_av<-mean(RTDO_90)
RTDO_95_av<-mean(RTDO_95)
RTDO_99_av<-mean(RTDO_99)
#
```

```
REDO_80_av<-mean(REDO_80)
REDO_90_av<-mean(REDO_90)
REDO_95_av<-mean(REDO_95)
REDO_99_av<-mean(REDO_99)

list (RMDO_80_av=RMDO_80_av,RMDO_90_av=RMDO_90_av,
   RMDO_95_av=RMDO_95_av, RMDO_99_av=RMDO_99_av,
#
        RMSO_80_av=RMSO_80_av, RMSO_90_av=RMSO_90_av,
   RMSO_95_av=RMSO_95_av, RMSO_99_av=RMSO_99_av,
#
        RTDO_80_av=RTDO_80_av, RTDO_90_av=RTDO_90_av,
   RTDO_95_av=RTDO_95_av, RTDO_99_av=RTDO_99_av,
#
        REDO_80_av=REDO_80_av, REDO_90_av=REDO_90_av,
   REDO_95_av=REDO_95_av, REDO_99_av=REDO_99_av)

} ## end of main function
```

```
compare_performances_1<-function (dim, dist_1,dist_2,nsim, sampsize,percent,pattern,
 cut_RMDO, cut_RMSO, cut_RTDO,cut_REDO) {
#main program
#we contaminated sample and compute the percent of outliers detected
#and the percent of regular observations declared as outliers
#pattern=1 means "Cluster of outliers".
#dist_1 control at what distance pattern 1 outliers are generated (b in paper)
#pattern=2 means "Radial outliers" .
#dist_2 control at what distance pattern 2 radial outliers are generated (B in paper)
#this program will not run if percent*sampsize=outsize (the number of outliers) is not integer!!
#alpha a parameter of the skew-nomal distribution (vector of length p=dim)
#percent is the percent of outliers in the sample (decimal number)
#dim is the dimension of the data (number of columns). Can take only values 2,3,5
#this program makes use of the following programs:
#rob_mahal_depth_out()
#rob_mahal_Spat_depth_out()
#rob_mahal_Triangle_depth_out()
#rob_mahal_Ellipse_depth_out()
#the following R packages need to be loaded for this program to run:
#"mnormt","sn","MASS","mgcv","robustbase","mvtnorm","car".
#library(mnormt)
#library(sn)
#library(MASS)
#library(mgcv)
#library(robustbase)
```

```
#library(mvtnorm)
#library(car)
#to hold the percent of outliers detected and
#to hold the Percent of regular observations declared as outliers with respect to different
#outlyingness functions.
RMDO_PercentOutDetected<-0
RMDO_PercentRegDecOut<-0
#
RMSO_PercentOutDetected<-0
RMSO_PercentRegDecOut<-0
#
RTDO_PercentOutDetected<-0
RTDO_PercentRegDecOut<-0
#
REDO_PercentOutDetected<-0
REDO_PercentRegDecOut<-0
#
#number of outliers. make sure this is an integer, otherwise the program will not run
outsize<-percent*sampsize
#parameters to use in generating skew-normal variate
Omega<-diag(dim)
mu<-rep(0,dim)
#alpha similar to that of Hubert and Van der Veeken, 2008
if (dim==2) {
        alpha<-c(10,4)
}
else if (dim==3) {
        alpha<-c(rep(10,1),rep(4,2))
}
else if (dim==5) {
        alpha<-c(rep(10,2),rep(4,3))
}
# number of regular observations
n<-sampsize-outsize
#cluster outliers generation
#position of center of outlier generating distribution
#
mu1<-rep(-dist_1,dim)
for (count in 1:nsim)  {
        if (pattern==1){
#generating regular observations from skew-normal distribution
#x1 is a matrix with n rows and p columns
#package "sn" needed to generate multivariate skew-normal
                x1<-rmsn(n, mu, Omega, alpha)[1:n,]
                x2<-rep(0,n)
#I tag the regular observations with zeros
```

```
                x<-cbind(x1,x2)
#generating outliers according to radial pattern.
#packages "mvtnorm" and "mnormt" needed
                out1<-rmvnorm(outsize, mean = mu1, sigma = diag(dim)/20) #the outliers
                out2<-rep(1,outsize)
#I tag the outliers with "ones"
                out<-cbind(out1,out2)
#create a matrix named Z to Stack regular observations on top of outliers.
#column 1 to Column (dim) contains the observations and outliers
#column (dim+1) contains 0 or 1 (0 for regular observations, 1 for outliers)
#the following Z has sampsize rows and dim+1 columns
# stack regular observations on top of outliers.
                Z<-rbind(x,out)
} #end pattern 1
#"radial" outliers generation
        if (pattern==2){
#generating Outliers according to "radial" pattern.
#first generate sampsize regular observations from skew-normal distribution
                x1<-rmsn(sampsize, mu, Omega, alpha)[1:sampsize,]
                a<-L2norm_Matrix(x1)
                b<-rank(a)
#match each observation with its L2norm rank
                x3<-cbind(x1,b)
#then replace outsize observations with observations with large L2norms with
#inflated observations
#n is the number of regular observations
        for (i in 1:n){
#handle the non outliers
#replace the rank with 0 meaning not outliers (in column b)
                x3[b==i,dim+1]<-0
}
        for (i in (n+1):sampsize){
#replace observations with large L2norm with outliers
#inflate those observations
                x3[b==i,1:dim]<-dist_2*(x3[b==i,1:dim])
#replace the rank (in b) with 1 (meaning outliers)
                x3[b==i,dim+1]<-1
#column (dim+1) contains 0 or 1 (0 for regular obs, 1 for outliers)
#as for for pattern 1, call Z the contaminated sample
                Z<-x3
}
} #end pattern 2
#need to compute "outlyingness" value for each point with respect to the data
#grab the first "dim" columns of Z. They will be used for outlyingness computations
#use this to compute outlyingness values
Z1<<-Z[,1:dim]
```

```
#
#compute Robust Mahalanobis "outlyingness" with MCD (RMDO) of observations
#use first "dim" columns of Z and store results in column (dim+2)
RMDO<-rob_mahal_depth_out(Z1) #vector of length sampsize
Z<-cbind(Z,RMDO) #add a column to Z
#
#compute Robust Mahal Spacial "outlyingness" with MCD (RMSO) of observations
#use first "dim" columns of Z and store results in column (dim+3)
RMSO<-rob_mahal_Spat_depth_out(Z1)
Z<-cbind(Z,RMSO) #add a column to Z. RMSO in column dim+3
#
#compute Robust Mahalanobis Triangle depth "outlyingness" with MCD (RTDO)
#of observations
#use first "dim" columns of Z and store results in column (dim+4)
RTDO<-rob_mahal_Triangle_depth_out(Z1)
Z<-cbind(Z,RTDO) #add a column to Z. RMSO in column dim+4
#
#compute Robust Mahalanobis Elliptical depth "outlyingness" with MCD (REDO)
#of observations
#Use first "dim" columns of Z and store results in column (dim+5)
REDO<-rob_mahal_Ellipse_depth_out(Z1)
Z<-cbind(Z,REDO) #add a column to Z. REDO in column dim+5
#End computation of outlyingness values
#using RMDO to find the outliers
#RMDO in column dim+2 of Z
#percent of outliers detected by RMDO (Row dim+2 of Z),
#robust Mahalanobis Distance Outlyingness
#the rows of Z where the elements declared as outliers
DecOut<-Z[ Z[,dim+2]>cut_RMDO, ]
#careful if the resulting matrix has one row it is converted to vector,
# and R cannot count the rows for some reason
if (is.vector(DecOut)==1) {
        DecOut<-matrix(DecOut,nrow=1)
}
#while they are outliers
DecOut1<-DecOut[ DecOut[,dim+1]==1,]
if (is.vector(DecOut1)==1) {
        DecOut1<-matrix(DecOut1,nrow=1)
}
#percent of outliers detected
RMDO_PercentOutDetected[count]<-(nrow(DecOut1)/outsize)*100
#regular Observations declared as outliers.
#
DecOut_2<-DecOut[ DecOut[,dim+1]==0,]
if (is.vector(DecOut_2)==1) {
        DecOut_2<-matrix(DecOut_2,nrow=1)
```

```
}
#recall n=sampsize-outsize
RMDO_PercentRegDecOut[count]<-(nrow(DecOut_2)/n)*100
#
###using RMSO to find the outliers
#RMSO in column dim+3 of Z
#percent of outliers detected by RMSO (Row dim+3 of Z),
#robust Mahal. spatial Outlyingness
#the rows of Z where the elements declared as outliers
DecOut<-Z[ Z[,dim+3]>cut_RMSO, ]
#careful if the resulting matrix has one row it is converted to vector,
# and R cannot count the rows for some reason
if (is.vector(DecOut)==1) {
        DecOut<-matrix(DecOut,nrow=1)
}
#and they are outliers
DecOut1<-DecOut[ DecOut[,dim+1]==1,]
if (is.vector(DecOut1)==1) {
        DecOut1<-matrix(DecOut1,nrow=1)
}
#percent of outliers detected
RMSO_PercentOutDetected[count]<-(nrow(DecOut1)/outsize)*100
# regular Observations declared as outliers.
#recall n=sampsize-outsize regular observations.
DecOut_2<-DecOut[ DecOut[,dim+1]==0,]
if (is.vector(DecOut_2)==1) {
        DecOut_2<-matrix(DecOut_2,nrow=1)
}
RMSO_PercentRegDecOut[count]<-(nrow(DecOut_2)/n)*100
#
##using RTDO to find the outliers
#RTDO in column dim+4 of Z
#percent of outliers detected by RTDO (column dim+4 of Z),
#Robust Triangle depth Outlyingness
#the rows of Z where the elements declared as outliers
DecOut<-Z[ Z[,dim+4]>cut_RTDO, ]
#careful if the resulting matrix has one row it is converted to vector,
# and R cannot count the rows for some reason
if (is.vector(DecOut)==1) {
        DecOut<-matrix(DecOut,nrow=1)
}
#and they are outliers
DecOut1<-DecOut[ DecOut[,dim+1]==1,]
if (is.vector(DecOut1)==1) {DecOut1<-matrix(DecOut1,nrow=1)}
#percent of outliers detected
RTDO_PercentOutDetected[count]<-(nrow(DecOut1)/outsize)*100
```

```
# regular Observations declared as outliers.
DecOut_2<-DecOut[ DecOut[,dim+1]==0,]
if (is.vector(DecOut_2)==1) {
        DecOut_2<-matrix(DecOut_2,nrow=1)
}
#recall n=sampsize-outsize regular observations.
RTDO_PercentRegDecOut[count]<-(nrow(DecOut_2)/n)*100
#
###Using REDO to find the outliers
#REDO in column dim+5 of Z
#percent of outliers detected by REDO (column dim+5 of Z)
#the rows of Z where the elements declared as outliers
DecOut<-Z[ Z[,dim+5]>cut_REDO, ]
#careful if the resulting matrix has one row it is converted to vector,
# and R cannot count the rows for some reason
if (is.vector(DecOut)==1) {
        DecOut<-matrix(DecOut,nrow=1)
}
#and they are outliers
DecOut1<-DecOut[ DecOut[,dim+1]==1,]
if (is.vector(DecOut1)==1) {
        DecOut1<-matrix(DecOut1,nrow=1)
}
#percent of outliers detected
REDO_PercentOutDetected[count]<-(nrow(DecOut1)/outsize)*100
#regular Observations declared as outliers.
DecOut_2<-DecOut[ DecOut[,dim+1]==0,]
if (is.vector(DecOut_2)==1) {
        DecOut_2<-matrix(DecOut_2,nrow=1)
}
#recall n=sampsize-outsize regular observations.
REDO_PercentRegDecOut[count]<-(nrow(DecOut_2)/n)*100
                } ## end of simulations
#compute averages of percents of outliers detected
#and percents of regular observations classified as outliers
#
RMDO_PercentOutDetected_av<-mean(RMDO_PercentOutDetected)
RMDO_PercentRegDecOut_av<-mean(RMDO_PercentRegDecOut)
#
RMSO_PercentOutDetected_av<-mean(RMSO_PercentOutDetected)
RMSO_PercentRegDecOut_av<-mean(RMSO_PercentRegDecOut)
#
RTDO_PercentOutDetected_av<-mean(RTDO_PercentOutDetected)
RTDO_PercentRegDecOut_av<-mean(RTDO_PercentRegDecOut)
#
REDO_PercentOutDetected_av<-mean(REDO_PercentOutDetected)
```

```
REDO_PercentRegDecOut_av<-mean(REDO_PercentRegDecOut)
#
list (RMDO_PercentOutDetected_av=RMDO_PercentOutDetected_av,
        RMDO_PercentRegDecOut_av=RMDO_PercentRegDecOut_av,
#
    RMSO_PercentOutDetected_av=RMSO_PercentOutDetected_av,
        RMSO_PercentRegDecOut_av=RMSO_PercentRegDecOut_av,
#
    RTDO_PercentOutDetected_av=RTDO_PercentOutDetected_av,
        RTDO_PercentRegDecOut_av=RTDO_PercentRegDecOut_av,
#
    REDO_PercentOutDetected_av=REDO_PercentOutDetected_av,
        REDO_PercentRegDecOut_av=REDO_PercentRegDecOut_av)

} ## end of main function
```