May 2022

Robust Estimation of Ornstein-Uhlenbeck Parameters

Timon Sebastian Kramer
University of Wisconsin-Milwaukee

Robust Estimation of Ornstein-Uhlenbeck Parameters

by

Timon Kramer

A Thesis Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Master of Science
in Mathematics

at

The University of Wisconsin-Milwaukee
May 2022

ABSTRACT

Robust Estimation of Ornstein-Uhlenbeck Parameters

by

Timon Kramer

The University of Wisconsin-Milwaukee, 2022
Under the Supervision of Professor Daniel Gervini

The standard estimators of the parameter of the Ornstein-Uhlenbeck process are vulnerable to contamination in the data sets. In this thesis more robust estimators for the parameter of the Ornstein-Uhlenbeck process are proposed which use medians instead of means. The scaling for these estimators is more complex and numerical methods must be used. A possible numerical implementation is described. The performance of the standard estimators and the proposed robust estimators are compared on data sets with different levels of contamination and different kind of errors. This thesis shows that the proposed robust estimators can be considerably better than the standard estimators on contaminated data sets.

Table of Contents

List of Figures

List of Tables

Introduction

Contamination in data sets can easily happen and it can be hard to reverse it. Sample means are very sensitive to contamination in data sets, whereas sample medians can be more resistant to them.

The standard estimators of the parameters of an Ornstein-Uhlenbeck process are vulnerable to contamination of the data sets as these are based on sample means. To make these estimators more resistant to contamination in data sets, the means can be replaced by medians. This yields a more robust set of estimators for the Ornstein-Uhlenbeck process.

In this master thesis, robust estimators of the parameters of an Ornstein-Uhlenbeck process are introduced based on [Falk, 1997] and a possible implementation of these is described. The performance of the robust estimators is then compared to the performance of the standard estimators on data sets with different contaminations.

The computation of the robust estimators is more complex than the computation of standard estimator. The medians have to be scaled for consistency. Especially the scaling of the robust estimators for the autocorrelation is complicated as it has no closed form and numerical methods must be used. In this thesis, the Monte Carlo method and a root-finding algorithm is used to compute an approximation of the robust autocorrelation estimator.

For the comparison of the performance of the robust estimators and the standard estimators, a data model is created using an Ornstein-Uhlenbeck process with one set of parameters. From this data model contaminated data sets are created and a comparison is conducted on each of them. The thesis shows that the robust estimators can be considerably better than the standard estimators on contaminated data sets.

The structure of the thesis is as follows: In Chapter 2, the theoretical basics of an Ornstein-Uhlenbeck process are provided and the standard and the robust estimators for the Ornstein-Uhlenbeck parameters are introduced. Chapter 3 provides a description of the implementation of the estimators. Especially, the implementation of the robust autocorrelation estimator which is needed for the robust estimation of λ and σ is described in more detail. In Chapter 4, the performance of the standard and robust estimators is analyzed on data sets with different contaminations. Chapter 5 concludes the thesis with a summary of the findings and an outlook for further research.

Theoretical Basics

This chapter states the theoretical basics of an Ornstein-Uhlenbeck process and introduces the estimation of its parameters based on a sample of observations using the standard and a robust approaches.

I.1 Ornstein-Uhlenbeck Process

This section introduces the Ornstein-Uhlenbeck process. It is based on [Rieder, 2012].

An Ornstein-Uhlenbeck process $X = (X_t)_{t\geq 0}$ is a stochastic process which evolves according to the following stochastic differential equation (SDE)

$$dX_t = \lambda(\mu - X_t)dt + \sigma dB_t$$

with parameters $\lambda, \sigma \in \mathbb{R}^+$, $\mu \in \mathbb{R}$ and $(B_t)_{t\geq 0}$ denoting the standard Brownian motion. The solution of this SDE is given by

$$X_t = e^{-\lambda t}X_0 + \mu(1 - e^{-\lambda t}) + \sigma \int_0^t e^{\lambda(s-t)}dB_s$$

where $\int_0^t e^{\lambda(s-t)}dB_s \sim \mathcal{N}(0, \frac{1-e^{-2\lambda t}}{2\lambda})$. In the following it is assumed that the initial distribution is the limiting distribution for $t \to \infty$, i.e. $X_0 \sim \mathcal{N}(\mu, \frac{\sigma^2}{2\lambda})$. In addition, it is assumed that X_0 is independent of $(\mathcal{F}_k)_{k\geq 0}$ with $\mathcal{F}_k = \sigma(\int_0^t e^{\lambda(s-t)}dB_s | t \leq k)$. Then, the process is Gaussian with $X_t \sim \mathcal{N}(\mu, \frac{\sigma^2}{2\lambda})$ and the covariance is given by $Cov(X_t, X_u) = \frac{\sigma^2}{2\lambda}e^{-\lambda|t-u|}$ for $t, u \geq 0$.

I.2 Estimating the Parameters

Assume an Ornstein-Uhlenbeck process as described above is observed on a regular time grid with $0 \leq t_1 < ... < t_n$ where $t_i - t_{i-1} = d$ with $i \in \{2, ..., n\}$ and $d \in \mathbb{R}^+$. The observed values of the Ornstein-Uhlenbeck process X_t at $t \in \{t_1, ..., t_n\}$ are denoted by $(X_{t_1}, ..., X_{t_n})$. The goal is to estimate the parameters μ, σ^2 and λ of the observed Ornstein-Uhlenbeck process using the observations. From above it is known that the observations $X_{t_1}, ..., X_{t_n}$ satisfy

$$\mathbb{E}(X_{t_i}) = \mu, \quad Cov(X_{t_i}, X_{t_{i-1}}) = \frac{\sigma^2}{2\lambda} e^{-\lambda d}, \quad Var(X_{t_i}) = \frac{\sigma^2}{2\lambda}$$

I.2.1 Standard Estimators

The standard estimators for μ, λ, σ^2 are based on estimations of the mean, variance, and correlation. These are given by

$$\hat{\mu} = \bar{X}, \quad \hat{\lambda} = -\frac{\ln(\hat{\rho})}{d}, \quad \hat{\sigma}^2 = 2\hat{\lambda}\hat{s}^2 \tag{I.2.1}$$

where $\bar{X}$ is the sample mean, $\hat{\rho}$ is the autocorrelation at lag 1 and $\hat{s}^2$ is the sample variance, i.e.,

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_{t_i}, \quad \hat{\rho} = \frac{\sum_{i=2}^{n}(X_{t_i} - \bar{X})(X_{t_{i-1}} - \bar{X})}{\sum_{i=1}^{n}(X_{t_i} - \bar{X})^2}, \quad \hat{s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_{t_i} - \bar{X})^2$$

A disadvantage of these estimators is that they are not resistant to contaminations in the data.

I.2.2 Robust Estimators

To gain more outlier resistant estimators for Ornstein-Uhlenbeck parameters, we replace the standard mean, variance and correlation estimators by robust alternatives. We propose the

following more robust estimators for the Ornstein-Uhlenbeck parameters. There, instead of using the mean in the formulas of the estimators, we use medians.

The sample mean $\bar{X}$ of the observations is replaced by the sample median, i.e.,

$$\tilde{X} = \text{med}_{i\in\{1,\dots,n\}}(X_{t_i})$$

Instead of using the sample standard deviation $\hat{s}$, the scaled median absolute deviation $\tilde{s}$ is used, which is given by

$$\tilde{s} = \frac{\text{med}_{i\in\{1,\dots,n\}}(X_{t_i} - \tilde{X})}{\Phi^{-1}(3/4)}$$

where Φ^{-1} is the quantile function of the standard normal distribution. This scaling factor is needed for consistency under the normal distribution (cf. [Falk, 1997][p. 616]).

In a similar way, a robust autocorrelation estimator $\tilde{\rho}$ can be obtained. However, the scaling for consistency is more complicated.

First, replacing mean with median in the standard autocorrelation formula yields the coefficient

$$\hat{\delta} = \frac{\text{med}_{i=\{2,\dots,n\}}((X_{t_i} - \tilde{X})(X_{t_{i-1}} - \tilde{X}))}{\text{med}_{i=\{1,\dots,n\}}(|X_{t_i} - \tilde{X}|)^2} \tag{I.2.2}$$

For scaling, we will use the same procedure as in [Falk, 1997] where it is used in a robust estimation of the correlation coefficient of a bivariate normal distribution. The robust estimator for the autocorrelation has then the form

$$\tilde{\rho} = g^{-1}(g(1)\hat{\delta}) \tag{I.2.3}$$

where the function $g(\rho)$ is defined as the median of XY with (X, Y) being a bivariate normal distribution with mean 0, unit variance and correlation ρ.

As shown in [Falk, 1997], $g(\rho)$ is such that it holds

$$P(XY \leq g(\rho)) = \frac{1}{2}$$

The function $g(\rho)$ defined on $\rho \in [-1, 1]$ is a strictly monotone increasing, symmetric and continuous function. It holds that

$$\begin{aligned} g(\rho_1) < g(\rho_2), \quad -1 \leq \rho_1 < \rho_2 \leq 1 \\ g(-\rho) = -g(\rho), \quad -1 \leq \rho \leq 1 \\ g(1) = \Phi^{-1}(3/4)^2 \end{aligned}$$

However, $g(p)$ has no closed form and must therefore be evaluated by Monte Carlo simulations.

After obtaining $\tilde{\rho}$ the formulas for the robust estimators of μ, λ and σ^2 are then given by

$$\tilde{\mu} = \tilde{X}, \quad \tilde{\lambda} = -\frac{\ln(\tilde{p})}{d}, \quad \tilde{\sigma}^2 = 2\tilde{\lambda}\tilde{s}^2 \tag{I.2.4}$$

Implementation of the Estimators

The standard estimators $\hat{\mu}$, $\hat{\lambda}$ and $\hat{\sigma}^2$ and the robust estimator $\tilde{\mu}$ can be easily implemented by using built-in functions in the statistics software R [R Core Team, 2020]. To compute the robust estimators $\tilde{\lambda}$ and $\tilde{\sigma}^2$, the robust autocorrelation estimator $\tilde{\rho}$ must be calculated. This is more complex as a closed formula for the functions g and g^{-1} does not exist.

To compute $\tilde{\rho}$ as given in Formula (I.2.3), g^{-1} must be evaluated at $g(1)\hat{\delta}$. This value can be calculated from the observations of the Ornstein-Uhlenbeck process using Formula (I.2.2) and the fact that $g(1) = \Phi^{-1}(3/4)^2$.

Evaluating $g^{-1}(g(1)\hat{\delta})$ is equivalent to finding $\tilde{\rho}$ such that $g(\tilde{\rho}) = g(1)\hat{\delta}$. Therefore, this can be reformulated as a root-finding problem. Define the function $f(\tilde{\rho}) := g(1)\hat{\delta} - g(\tilde{\rho})$. The function has exactly one root for $\hat{\delta} \in [-1, 1]$ as $g(\rho)$ is strictly increasing, continuous and $g(\rho) \in [-\Phi^{-1}(3/4)^2, \Phi^{-1}(3/4)^2]$.

Therefore, evaluating $g^{-1}(g(1)\hat{\delta})$ includes two tasks. A root-finding algorithm must be applied on the function $f(\rho)$ to find ρ such that for a given tolerance $\epsilon > 0$, it holds $|f(\rho)| = |g(1)\hat{\delta} - g(\tilde{\rho})| < \epsilon$. To be able to calculate $f(\rho)$, $g(\rho)$ must be computed for different values of ρ. For this Monte Carlo method is used.

I.3 Evaluation of the Function g

To be able to calculate $g(\rho)$ in the root-finding algorithm, $m \in \mathbb{N}$ independent bivariate standard normal distributed random variables are created, i.e.,

$$\begin{pmatrix} X_i \\ Y_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right), \text{ for } i \in \{1, ..., m\}$$

For this the R function "rmvnorm" is used from the package "Rfast" [Manos Papadakis and Chatzipantsiou, 2021]. During the whole root-finding algorithm and for the computations of g for different ρ, the same set of X_i, Y_i with $i \in \{1, ..., m\}$ is used.

For a given $\rho \in [-1, 1]$, the covariance matrix is defined as

$$\Sigma_\rho = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Using Cholesky decomposition $\Sigma_\rho^{\frac{1}{2}}$ is calculated from the covariance matrix. This is done by the build in function "chol" in R.

Bivariate normal random variables with covariance ρ can be created using $\Sigma_\rho^{\frac{1}{2}}$ and the bivariate standard normal random variables X_i, Y_i, i.e.,

$$\begin{pmatrix} X_{i,\rho} \\ Y_{i,\rho} \end{pmatrix} := \Sigma_\rho^{\frac{1}{2}} \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_\rho \right), \text{ for } i \in \{1, ..., m\}$$

Therefore, $(X_{i,\rho}, Y_{i,\rho})_{1,...,m}$ is a set of observations of a bivariate Normal distributed random variable with mean 0, unit variance and correlation ρ.

From this, $g(\rho)$ can be estimated as

$$g(\rho) = \text{med}_{i \in \{1,...,m\}}(X_{i,\rho} Y_{i,\rho})$$

In this application, the number of bivariate normal variables was set to $m = 10,000$. This yields an appropriate trade-off between accuracy and computational effort concerning the application of this thesis.

I.4 Implementation of the Root-Finding algorithm

As described above, finding $g^{-1}(g(1)\hat{\delta})$ can be formulated as a root-finding problem for the function $f(\tilde{\rho}) = g(1)\hat{\delta} - g(\tilde{\rho})$. The function f is continuous, strictly decreasing and has exactly one root on the interval $[-1, 1]$.

To find the root of f, a numerical method is used. Applying the regula-falsi method on a continuous function has the advantage that it will converge to a root and it does not need the derivative of the function which can be costly to evaluate. However, it converges slowly on an interval where the function is convex or concave. There, one of the end-points is always kept which leads to a first order asymptotic convergence (cf. [Dowell and Jarratt, 1971]). This problem is faced by the Illinois method which is a modification of the Regula-Falsi method. The Illinois method still has a guaranteed convergence. It is used in the implementation of this thesis. The following description of the Illinois method is based on [Dowell and Jarratt, 1971].

To find a root of the function f, the Illinois method needs two starting points $\alpha_1, \beta_1 \in [-1, 1]$ with $f(\alpha_1)f(\beta_1) < 0$. This makes sure that there is a root between α_1 and β_1. Set $f_{\alpha_1} := f(\alpha_1)$ and $f_{\beta_1} := f(\beta_1)$. The next point is calculated by the rule

$$\gamma_l = \beta_l - f_{\beta_l} \frac{\beta_l - \alpha_l}{f_{\beta_l} - f_{\alpha_l}}, \quad l \in \mathbb{N} \tag{I.4.1}$$

If the stop condition is triggered by γ_l, i.e., $|f(\gamma_l)| < \epsilon$, then an approximation of the root is found and $\tilde{\rho} = \gamma_l$ is set. Otherwise, there are two cases. If $f(\gamma_l)f_{\beta_l} < 0$ set

$$\beta_{l+1} := \gamma_l, \quad f_{\beta_{l+1}} := f(\gamma_l), \quad \alpha_{l+1} := \beta_l, \quad f_{\alpha_{l+1}} := f_{\beta_l}$$

and compute γ_{l+1} by Formula (I.4.1). If $f(\gamma_l)f_{\alpha_l} < 0$ set

$$\beta_{l+1} := \gamma_l, \quad f_{\beta_{l+1}} := f(\gamma_l), \quad \alpha_{l+1} := \alpha_l, \quad f_{\alpha_{l+1}} := \frac{1}{2} f_{\alpha_l}$$

and compute γ_{l+1} according to Formula (I.4.1).

The function g satisfies $g(-\rho) = -g(\rho)$. This implies, $g(0) = 0$. Since, g is also strictly increasing, it holds that $g_{|[0,1]} : [0,1] \to [0, \Phi^{-1}(3/4)^2]$ and $g_{|[-1,0]} : [-1,0] \to [-\Phi^{-1}(3/4)^2, 0]$. Therefore, if $g(1)\hat{\delta} > 0$ set as starting values for the Illinois method $\alpha_1 := 0$ and $\beta_1 = 1$. If $g(1)\hat{\delta} < 0$ set as starting values $\alpha_1 := 0$ and $\beta_1 = -1$. This will skip several steps in the beginning. As a root of f is found by the Illinois method, an approximation $\tilde{\rho}$ of $\rho = g^{-1}(g(1)\hat{\delta}$ is found which satisfies the condition $|g(1)\hat{\delta} - g(\tilde{\rho})| < \epsilon$. In the implementation of this thesis, the tolerance is set to $\epsilon = 10^{-6}$.

Performance of the Estimators

In this chapter, the robust estimators for the Ornstein-Uhlenbeck parameters are compared to the standard estimators. This comparison is done on a numerical example. First, the data sets on which the comparison is performed are described. Data sets are created from an Ornstein-Uhlenbeck process. Those data sets are then contaminated in different ways. The standard and robust estimators are then applied on the data sets without any contaminations, the so-called clean data sets, and the contaminated data sets. The performance of the standard and the robust estimators on the clean data sets and the contaminated data sets is then analyzed.

I.5 Description of the Data Sets

First, 5000 independent replications of 1000 observations of an Ornstein-Uhlenbeck process as described in Section (I.1) to the parameters $\mu = 0$, $\sigma = 1$, and $\lambda = 0.5$ and distance of $d = 1$ are created. Those data sets are referred to as the clean data sets in the following.

From those data sets, nine different sets of data sets with contaminations were created by looking at three different error types to three different levels of contaminated data each. The three levels were 1%, 10% and 20%.

The first types of contaminated data sets were created by randomly picking 1%, 10% and 20% of data points of each trial and assign them a value created by a Cauchy distribution with parameters location = 0 and scale = 0.5. This type of error is in the following called Cauchy error.

The second types of contaminated data sets were created by picking randomly 1%, 10%, 20% of consecutive data points of each trial and add to their value five times the standard deviation of X_i, i.e., $5\sqrt{\frac{\sigma^2}{2\lambda}}$. This type of error is in the following referred to as consecutive gross error.

The third types of contaminated data sets were created by picking randomly 1%, 10%, 20% of non-consecutive data points of each trial and add to their value five times the standard deviation of X_i. This type of error is in the following referred to as non-consecutive gross error.

For each trial in the contaminated data sets, the parameters of the underlying Ornstein-Uhlenbeck process, i.e., μ, σ, λ, were estimated using the robust estimators and the standard estimators.

In the following, the results of the standard estimators and the robust estimators for the different data sets are compared. First, the performance on the clean data sets is analyzed. Then the performance on the different contaminated data sets is compared.

I.6 Performance of the Estimators on Clean Data

The mean squared error (MSE) between the estimated value and the true value over all 5000 trials for the robust and the standard estimators on the clean data sets to each parameter are listed in the table below.

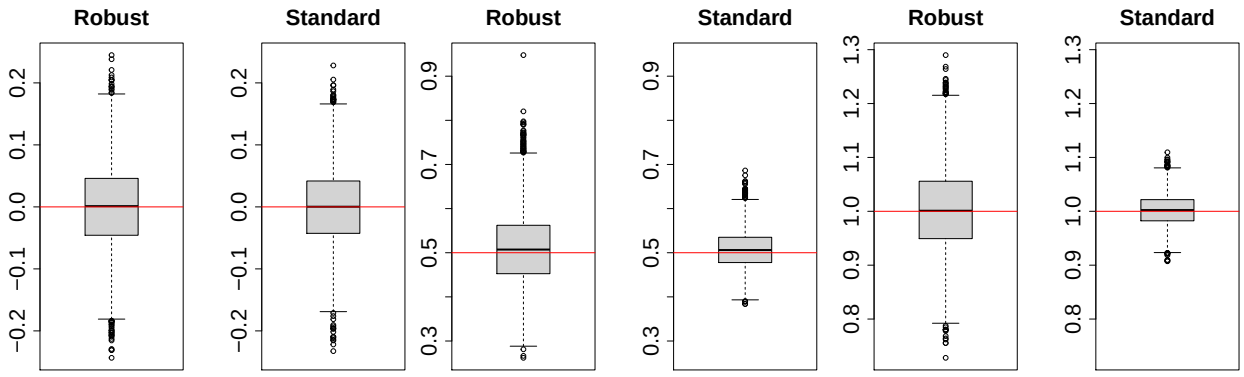
Parameter	μ	λ	σ
MSE Robust Estimator	0.004715332	0.007078571	0.00631273
MSE Standard Estimator	0.004036685	0.00189052	0.0008186828

Table I.1: MSE of robust estimators and standard estimators on the clean data sets

The MSE from the standard estimators is lower for each parameter compared to the robust estimators. Larger differences can be seen for estimating λ and σ. There, the MSE of the robust estimators is around 4 times respectively around 8 times larger than the MSE of the

standard estimators. Suggesting that the standard estimators are performing considerably better for clean data sets than the robust estimators.

In Figure I.1, the estimated values for the parameters μ, λ and σ using the robust and standard estimators of each trial of the clean data sets are compared in boxplots. In all following boxplots, the red line represents the true value from the simulated Ornstein-Uhlenbeck process.



(a) Estimation of μ (b) Estimation of λ (c) Estimation of σ

Figure I.1: Boxplots for estimating parameters using robust estimators and standard estimators on the clean data sets

The median of the estimated values to each parameter and estimation method are very close to the true value. The boxplots for the estimated μ values are very similar for the robust and standard estimation. For estimations of λ and σ the boxplots from the robust and standard estimation are more different. There, the interquartile range of the standard estimation is considerably smaller compared to the robust estimation and the minimum and maximum values are also closer to the true value. This shows, that the estimations using the standard estimatorsare closer to the real value compared to the robust estimators.

From the MSE and the boxplots, it can be seen that the robust and standard estimators are good estimators for the parameters of the Ornstein-Uhlenbeck process in this setting as the MSE is low and the median is near the true parameter values. Especially for estimating λ and σ the standard estimators is considerably better than the robust estimators. For μ

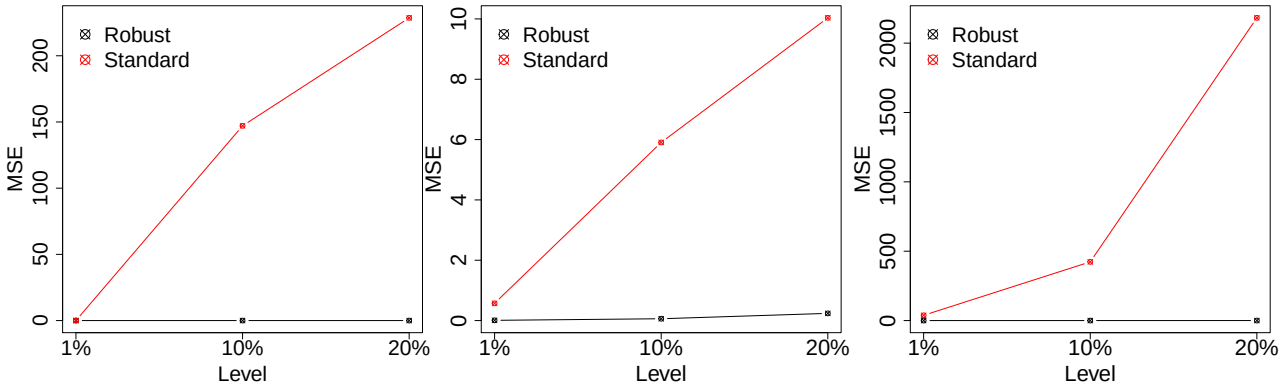
the standard estimator is also better but the differences are smaller.

I.7 Performance of the Estimators on Contaminated Data

I.7.1 Cauchy Error

In the following, the three contaminated data sets using the Cauchy error on the different contamination levels are analyzed.

The MSE over all 5000 trials between the estimated value and the true value using the robust and standard estimators on the Cauchy Error data sets to each parameter and contamination level is shown in the Figure I.2. The values are listed in Table I.2.



(a) MSE Estimation of μ (b) MSE Estimation of λ (c) MSE Estimation of σ

Figure I.2: MSE of robust estimators and standard estimators on the Cauchy error data sets

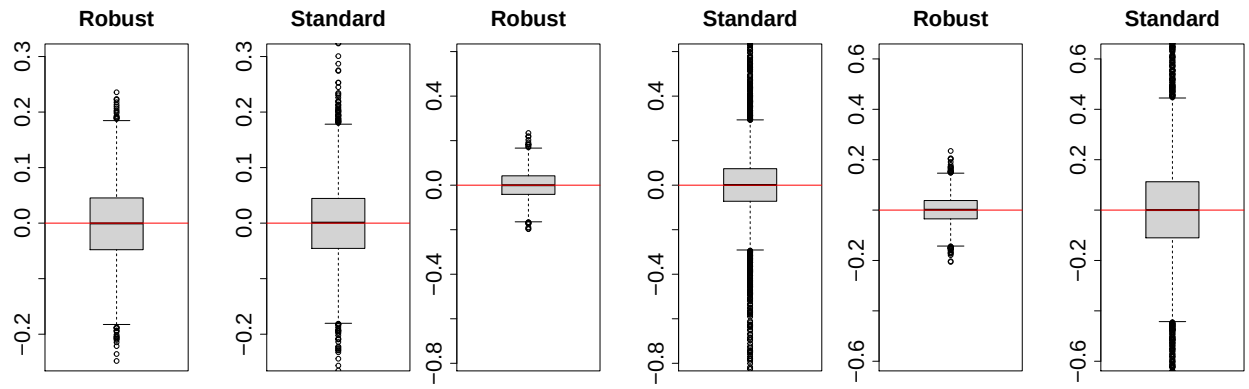
Parameter	μ	λ	σ
MSE Level 1 - RE	0.004682336	0.008237459	0.007009895
MSE Level 1 - SE	0.09105189	0.6275789	43.77125
MSE Level 2 - RE	0.003776902	0.06036973	0.0365415
MSE Level 2 - SE	6.01104	5.780263	475.7216
MSE Level 3 - RE	0.002928552	0.2390789	0.1130848
MSE Level 3 - SE	408.4647	10.39043	1234.749

Table I.2: MSE of robust estimators (RE) and standard estimators (SE) on the Cauchy error data sets

The difference in the MSEs are very considerably between the robust and the standard estimators for each parameter and contamination level. The MSE of the standard estimators is between around 19 and 139476 times higher than the MSE of the robust estimators. The highest difference can be seen in σ and μ. The MSE of the robust estimators is the lowest for parameter μ. The MSE increases over the contamination levels except for estimating μ with the robust estimator. The robust μ estimator is especially robust as between the different level almost no difference can be seen. Where as the MSE for λ and σ increases however considerably less than for the standard estimators. The MSE of the standard estimators is already in the first contamination level way higher than for the robust estimators.

The extreme difference is caused by the extreme values of the Cauchy distribution. The standard estimator uses means for estimation. The means however can explode due to the extreme values of the distribution.

In Figure I.3, the estimated values of μ using the robust and the standard estimators over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

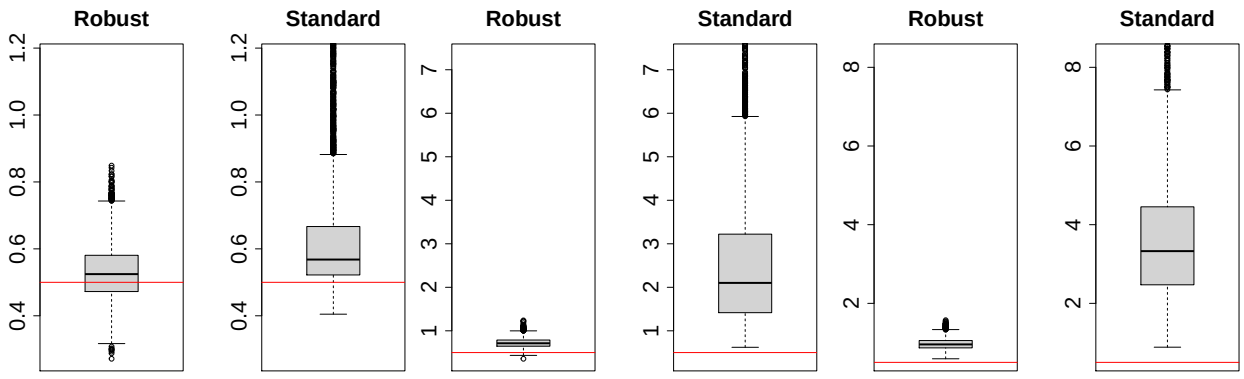
Figure I.3: Boxplots for estimating parameter μ using robust estimator and standard estimator on the Cauchy error data sets

For the 1% level, the boxes of the robust and the standard estimator are very similar. The median of both lies on the real μ value, the 25%- and 75%-quantile and the whiskers are on a very similar level, too. However, the number of outliers differs. The standard estimator has considerably more outliers beyond the whiskers.

For the 10%, 20% levels, the boxes of the standard estimator are larger than the boxes of the robust estimator. More striking is, however, the number of outliers. The robust has a few outliers near its whiskers. The standard estimator has a high number of outliers starting at the whiskers and up to multiple interquantile ranges away from the whiskers. The median for the robust and standard estimator still lies near the real μ value.

By contaminating the data using the Cauchy distribution, the mean of each trial can increase or decrease significantly as the Cauchy distribution can yield extremely large positive or negative values.

Figure I.4 shows the estimated values of λ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

Figure I.4: Boxplots for estimating parameter λ using robust estimators and standard estimators on the Cauchy error data sets

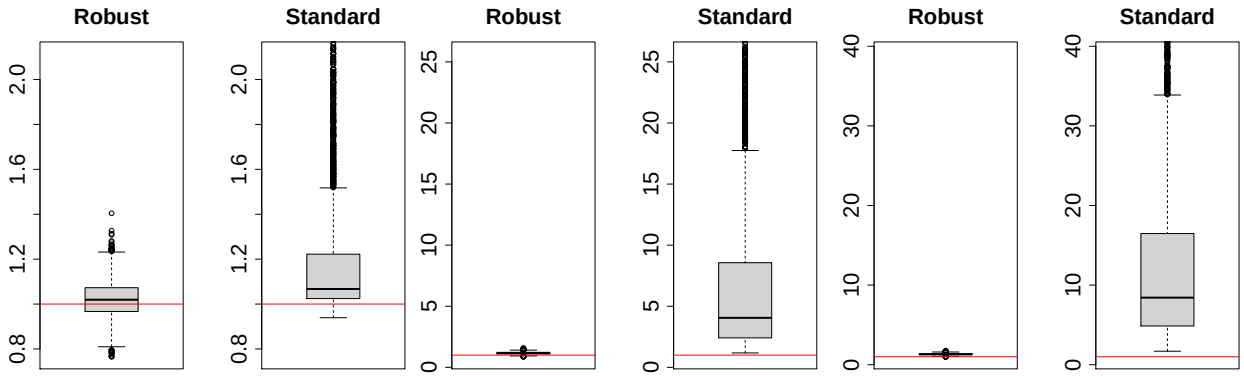
For the 1% level, the real value of λ is at the robust estimator between the 25%-quantile and the median. For the standard estimator, it is between the lower whisker and the 25%-quantile, meaning that in most of the cases the standard estimator overestimates λ. The box of the standard estimator is larger than the box of the robust estimator. Again the number of outliers differs considerably. Whereas the robust estimator has a few outliers near the lower whiskers and the higher whiskers, the standard estimator only has outliers above the higher whiskers. These range from at the higher whisker up to several multiples of the interquartile range away.

For the 10% and 20% level, robust and standard estimators are both overestimating λ, however, the standard estimator does far worse. For the robust estimator, the real λ value is between the lower whisker and the 25%-quantile or slightly below the lower whisker. For the standard estimator the real value is way below the lower whisker. In addition, the boxplots of the standard estimator is multiple times larger than the boxplots of the robust estimator. As before, the standard estimator has a high number of outliers above the upper whisker which can be multiple interquartile ranges away from the upper whisker.

By contaminating the data by randomly selecting data points in the trials and replace them with a value from the Cauchy distribution, the covariance between data points de-

creases. A lower covariance is represented by a higher λ. This can explain the overestimation of λ.

In Figure I.5, the estimated values of σ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are shown in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

Figure I.5: Boxplots for estimating parameter σ using robust estimators and standard estimators on the Cauchy error data sets

For the 1% level, the real value of σ is for the robust estimator very close to the median. Whereas, for the standard estimator the real value is between the lower whisker and the 25%-quantile. Meaning that the standard estimator overestimates σ. For the 10%, 20% levels, both estimators overestimate σ. For the robust estimator, the real σ value is between the lower whisker and the 25%-quantile or slightly below the lower whisker. For the standard estimator the real value is way below the lower whisker. In addition, the boxplots of the standard estimator is multiple times larger than the boxplots of the robust estimator. The standard estimator has a high number of outliers above the upper whisker which can be multiple interquartile ranges away from the upper whisker.

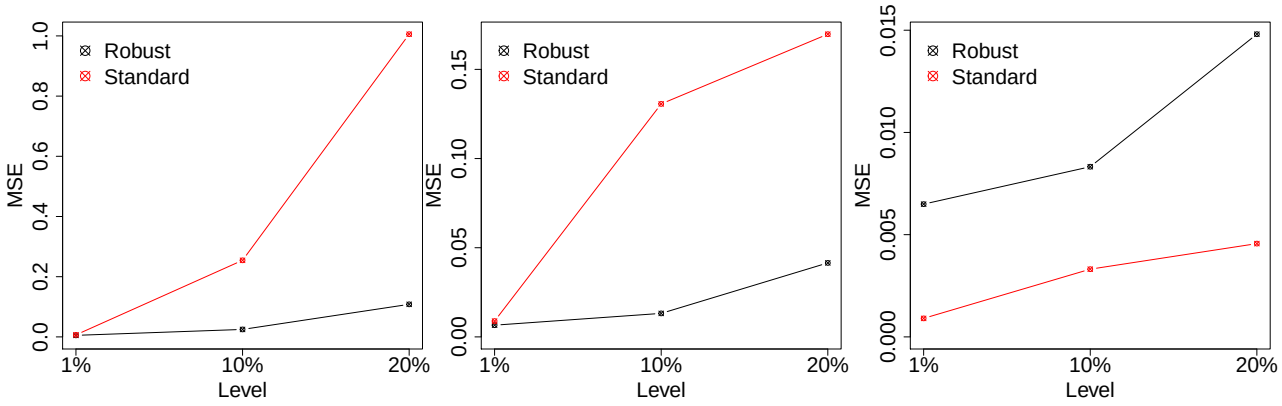
The contamination of the data using the Cauchy distribution increases the sample variance as the Cauchy distribution can lead the extreme large values. An increased variance leads to a higher σ. This can explain the overestimation of σ.

For the Cauchy error, the robust estimators are performing considerably better than

the standard estimators. For estimating μ using the robust and standard estimator, the median of the trials is near the real value. The boxes are also quite similar, however, the robust estimator has considerably less outliers. For λ and σ, both the standard and robust estimators overestimate the true value. However, the standard estimators perform way worse with values several times larger than the values of the robust estimators. The boxplots of the robust estimators are still near the true value of λ and σ.

I.7.2 Consecutive Gross Error

The MSE over all 5000 trials for the robust and standard estimators on the consecutive gross error data sets to each parameter and contamination level is shown in the Figure I.6. The values are listed in Table I.3.



(a) MSE Estimation of μ (b) MSE Estimation of λ (c) MSE Estimation of σ

Figure I.6: MSE of robust estimators and standard estimators on the consecutive gross error data sets

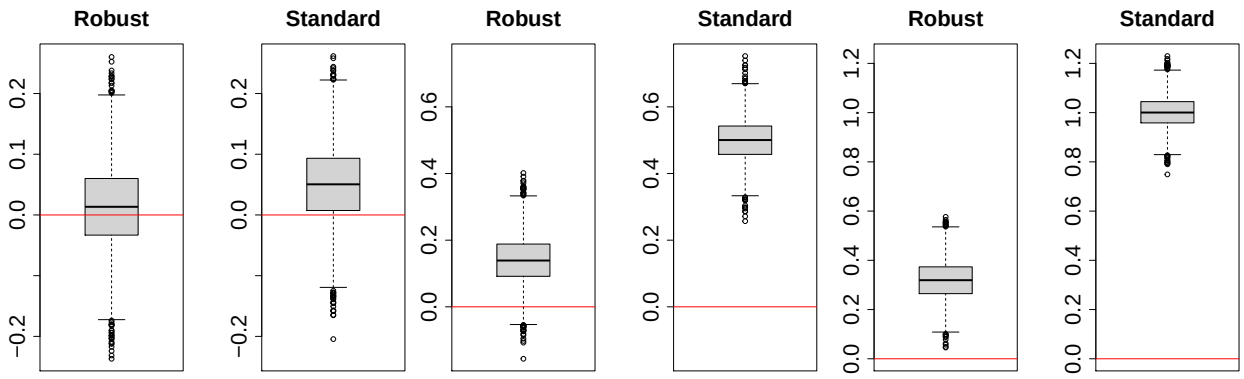
Parameter	μ	λ	σ
Consecutive Gross Error Level 1 - RE	0.00487433	0.006781321	0.006488062
Consecutive Gross Error Level 1 - SE	0.006545356	0.008580982	0.0009002971
Consecutive Gross Error Level 2 - RE	0.02477688	0.01291377	0.008568548
Consecutive Gross Error Level 2 - SE	0.2542559	0.1304651	0.003281597
Consecutive Gross Error Level 3 - RE	0.1065416	0.04188863	0.01436949
Consecutive Gross Error Level 3 - SE	1.001763	0.169762	0.004510387

Table I.3: MSE of robust estimators (RE) and standard estimators (SE) on the consecutive error data sets

The MSE for estimating μ and λ is for the robust estimators smaller than for the standard estimators across all contamination levels. For σ, the standard estimator lead to a smaller MSE over all contamination levels. The MSE increases with a higher contamination level for all parameters and both types of estimators.

The MSE of the standard estimators and robust estimators for estimating μ and λ at the 1% level are quite similar. The MSE of the robust estimators is only slightly lower. For the 10%, 20% levels, the MSE of the standard estimators for estimating μ and λ is around 4 to 10 times higher than the MSE of the robust estimators. For σ the MSE of the robust estimator is around 2.5 to 7 times larger than the MSE of the standard estimator.

In Figure I.7, the estimated values of μ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

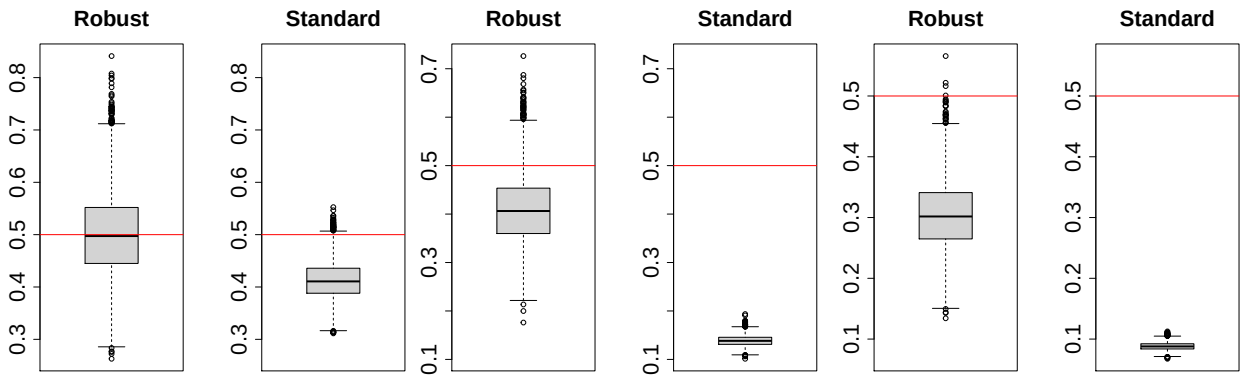
Figure I.7: Boxplots for estimating parameter μ using robust estimator and standard estimator on the consecutive gross error data sets

For the 1% level, the median of the robust estimator is slightly above the true value of μ. For the standard estimator, the true value of μ lies slightly below the 25%-quantile. Meaning that in most cases, the standard estimator overestimates μ. The boxplot of the robust estimator are quite similar, except that the boxplot for the standard estimator is slightly shifted up.

For the 10% and 20% levels, the robust and standard estimators are overestimating μ. However, the standard estimator does far worse. The boxplots of the robust estimator are slightly wider than the boxplots of the standard estimator. The boxplot of the robust estimator is with its lower whisker near the true value of μ, the boxplot of the standard estimator is shifted way up and is not near the true value.

By contaminating the data in that way, the mean and median of each trial increases as to 1%, 10%, 20% of the data points a fixed value is added. Therefore, μ is overestimated. The estimators for μ yield similar results as for the non-consecutive gross error data sets as for this estimator the order of the contaminated data does not matter.

In Figure I.8, the estimated values of λ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

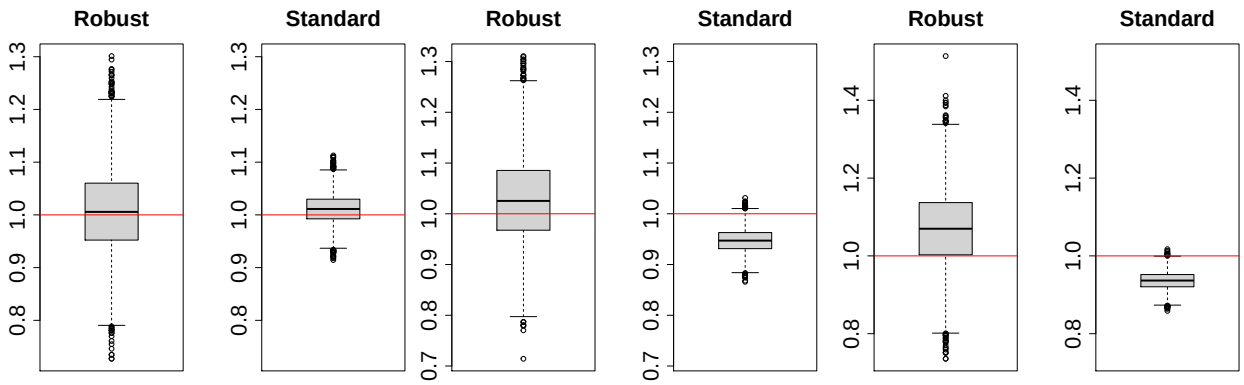
Figure I.8: Boxplots for estimating parameter λ using robust estimator and standard estimator on the consecutive gross error data sets

At the 1% level, the median of the robust estimator lies on the true λ value. The standard estimator underestimates the true λ value in most cases as its upper whisker is close to the true λ value. The range of the boxplot of the robust estimator is larger than the range of the standard estimator. Most of the values of the robust estimator, however, still lie closer to the true value.

For the 10% level, the standard estimator underestimates λ in all cases. For the robust estimator, the true value lies between the upper whisker and the 75%-quantile. For the 20% level, the robust and standard estimators are both underestimating λ. For the 10%, 20% levels, the robust estimator has a far larger range of values than the standard estimator. However, almost all values of the robust estimator are closer to the true λ value.

The underestimating of λ can be explained by the way of how the data is contaminated. By choosing 1%, 10%, or 20% of consecutive data points and adding five times the standard deviation to them, the covariance between the data points increase. A higher covariance is represented by a lower λ.

In Figure I.9, the estimated values of σ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



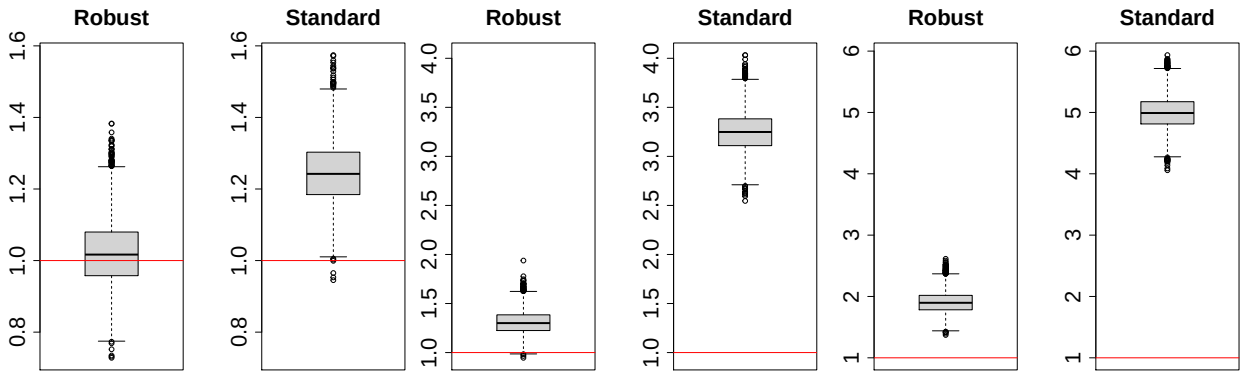
(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

Figure I.9: Boxplots for estimating parameter σ using robust estimator and standard estimator on the consecutive gross error data sets

For the 1% level, the median of the robust estimator is close to the true σ value. For the standard estimator, the true σ value lies between the 25%- quantile and the median. The value range of the robust estimator is far larger than the range of the standard estimator. Most of the values from the standard estimator lie closer to the true value than the values from the robust estimator. For the 10% level, the true value lies between the 25% quantile and the median for the robust estimator. The true value lies near the upper whisker for the standard estimator. For the 20% level, the true value lies on the 25%-quantile of the robust estimator and lies on the upper whisker for the standard estimator. The standard estimator underestimates σ in most cases whereas the robust estimator tends to overestimate σ. In both cases, the range of the values of the robust estimator is far greater than the range of values for the standard estimator. Most of the values of the standard estimator lie closer to the true value of σ than the values of the robust estimator.

These observations are in line with the larger mean squared error of the robust estimator compared to the standard estimator. To analyze the reason for the better performance of the standard estimator, the formula for σ must be investigated. By Formulas I.2.1 and I.2.4 the estimator of σ is the product of λ and s^2 which is the sample variance for the standard estimator respectively the squared scaled median absolute deviation for the robust estimator.

In Figure I.10, the estimated values of s^2 using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

Figure I.10: Boxplots for estimating s^2 using robust estimator and standard estimator on the consecutive gross error data sets

The true s^2 value of the original data sets is equal to 1. The standard estimator overestimates s^2 over all three contamination levels considerably. In the 1% level for the robust estimator the true value is between the 25%-quantile and the median. For the 10% and 20% levels, the robust estimator also overestimates the true level considerably. However, the robust estimator does far better than the standard estimator.

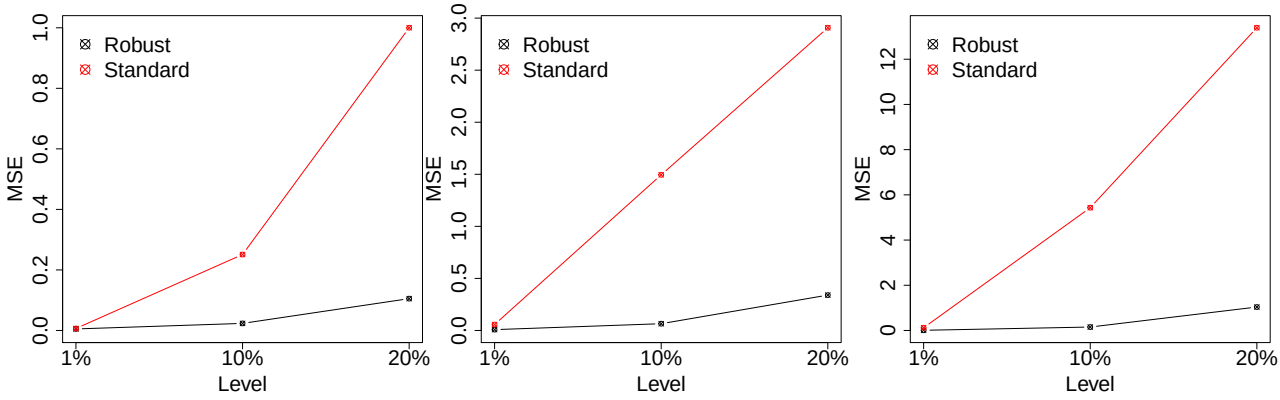
The standard estimator is a worse estimator for λ than the robust estimator. Here it underestimate the true value. For s^2 the standard estimator does also worse than the robust estimator. Here it overestimates the true value. However, judging by the MSE, the standard estimator performs better for estimating σ than the robust estimator. The errors of the standard estimator in estimating λ and s^2 seem to cancel each other out to some extend by multiplying these two values for estimating σ. The robust estimator also underestimates λ and overestimates s^2 for 10% and 20% cases and is in both cases a better estimator than the standard estimator. These two errors also cancel each other out to some extend but not as much as for the standard estimator.

For the consecutive gross error, the robust estimators are performing considerably better

for estimating λ and μ than the standard estimator. For σ, the standard estimator performs better than the robust estimator. σ is estimated by multiplying λ and s^2. The robust estimators perform better in estimating each of those. However, the standard estimators underestimate λ and overestimate s^2. These error cancel then each other out to some extend which then lead to a better performance in estimating σ.

I.7.3 Non-Consecutive Gross Error

The MSE over all 5000 trials for the robust and standard estimators on the non-consecutive gross error data sets to each parameter and contamination level is shown in the Figure I.11. The values are listed in Table I.4.



(a) MSE Estimation of μ (b) MSE Estimation of λ (c) MSE Estimation of σ

Figure I.11: MSE of robust estimators and standard estimators on the non-consecutive gross error data sets

The MSE for estimating μ, σ and λ is for the robust estimators smaller than for the standard estimators across all contamination levels. The MSE increases with a higher contamination level for all parameters and both types of estimators. The values of the MSE for μ are very similar to the MSE for μ on the continuous gross error data sets as for estimating μ, the order of the added error does not matter.

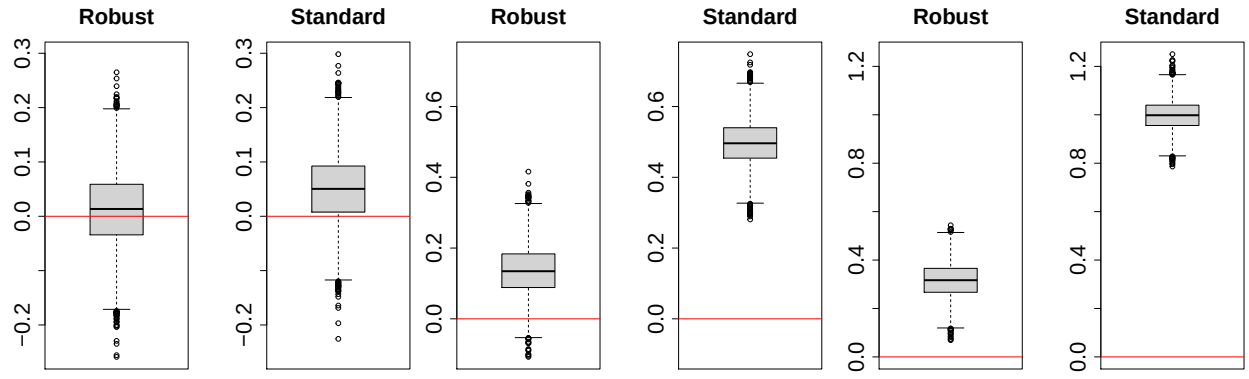
The smallest relative difference in the MSE between the robust and the standard estimators is for μ at the 1%-level. There the MSE of the standard estimator is around 30%

higher. For the rest, the MSE of the standard estimators is around 7 to 35 times larger than the MSE of the robust estimators.

Parameter	μ	λ	σ
Non-Consecutive Gross Error Level 1 - RE	0.004886138	0.008052146	0.007692788
Non-Consecutive Gross Error Level 1 - SE	0.006470841	0.05692788	0.1210731
Non-Consecutive Gross Error Level 2 - RE	0.02466052	0.06631387	0.1536988
Non-Consecutive Gross Error Level 2 - SE	0.2551777	1.503825	5.441474
Non-Consecutive Gross Error Level 3 - RE	0.1059201	0.3430731	1.037767
Non-Consecutive Gross Error Level 3 - SE	1.002822	2.900362	13.39157

Table I.4: MSE of robust estimators (RE) and standard estimators (SE) on the non-consecutive error data sets

In Figure I.12, the estimated values of μ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

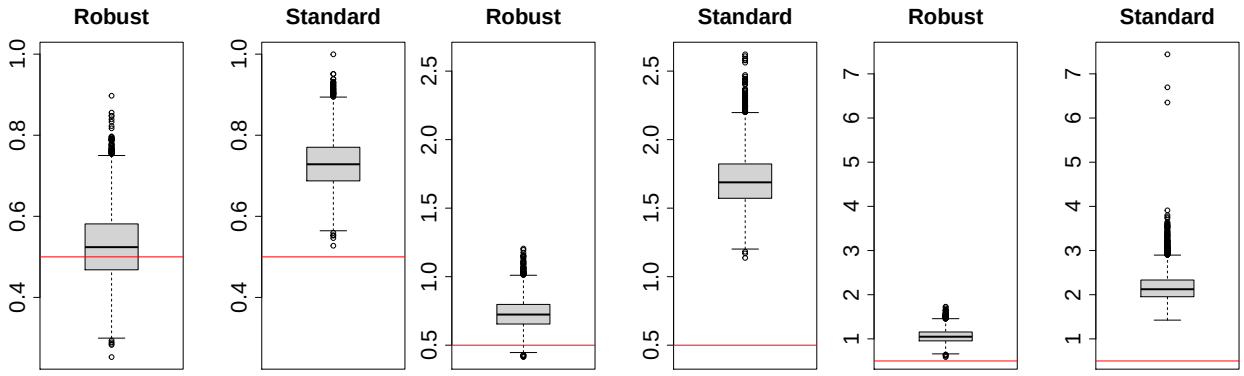
Figure I.12: Boxplots for estimating parameter μ using robust estimator and standard estimator on the non-consecutive gross error data sets

The results of estimating μ with the standard and robust estimators for the non-consecutive error are very similar to the results of the consecutive error. The reason here is, that the estimator for μ do not depend on the order of the contaminated data points, i.e., here it does not matter if the error are in consecutive data points or at randomly selected data points.

For the 1% level, the median of the robust estimator is slightly above the true value of μ. For the standard estimator, the true value of μ lies slightly below the 25%-quantile. For the 10% and 20% levels, the robust and standard estimators are overestimating μ. However, the standard estimator does far worse. The boxplot of the robust estimator is with its lower whisker near the true value of μ, the boxplot of the standard estimator is shifted way up and is not near the true value.

By contaminating the data in that way, the mean and median of each trial increases as to 1%, 10%, respectively 20%, of the data points a fixed value is added. Therefore, μ is overestimated.

In Figure I.13, the estimated values of λ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 3 - 20%

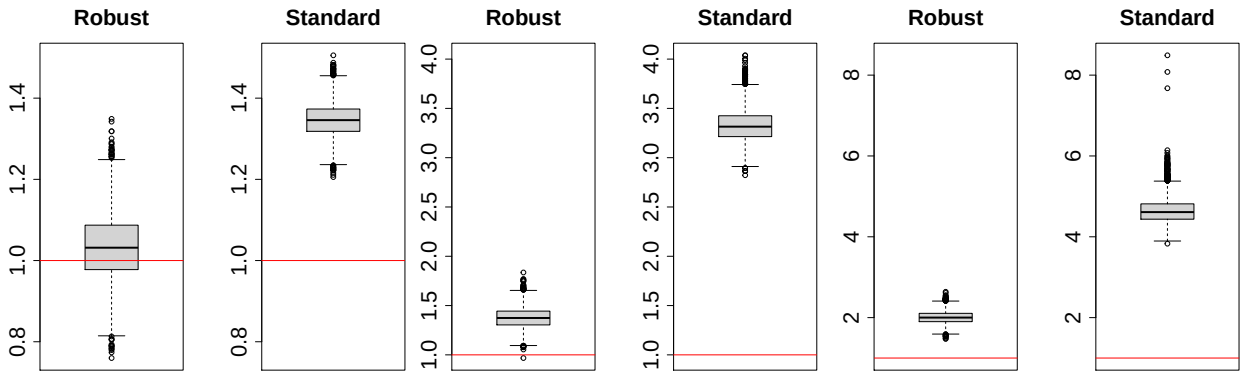
Figure I.13: Boxplots for estimating parameter λ using robust estimator and standard estimator on the non-consecutive gross error data sets

For the 1% level, the true value of λ lies between the 25%-quantile and the median. The standard estimator overestimates λ as the true value lies below the lower whisker. For the 10% and 20% level, the robust and the standard estimator overestimate λ. The standard estimator does worse than the robust estimator. For the robust estimator, the lower whisker is near the true value. The standard estimator is far above the true value. The standard estimator also has a larger number of outliers above the upper whisker. The outliers above

the upper whisker can be several interquartile ranges away from the upper whisker. Whereas the outliers of the robust estimator are closer to the whiskers.

By contaminating the data by non-consecutive gross errors, the covariance between data points decreases. A lower covariance is represented by a higher λ. This explains the overestimating of λ of both estimators.

In Figure I.14, the estimated values of σ using the robust and the standard estimator over the 5000 trials for each of the three contamination levels are plotted in boxplots.



(a) Contamination Level 1 - 1% (b) Contamination Level 2 - 10% (c) Contamination Level 1 - 20%

Figure I.14: Boxplots for estimating parameter σ using robust estimator and standard estimator on the non-consecutive gross error data sets

For the 1% level, the true value is between the 25%-quantile and the median for the robust estimator. Whereas the boxplot of the standard estimator is way above the true value. For the 10% and 20% levels, the robust estimator and the standard estimator both overestimate σ. The standard estimator performs worse and is far above the true value of σ. The standard estimator also has more outliers, mostly above the upper whisker. The outliers are up to several interquatile ranges away from the upper whisker for the standard estimator. Whereas the outliers of the robust estimator are close to the whiskers.

The overestimation of σ can be explained. By contaminating the data by non-consecutive gross errors, the variance between data points of the data sets increases. A higher variance in the data sets leads to a higher estimated s^2. As described before, the non-consecutive

contamination also leads to a higher λ. Both effects combined lead to a higher σ.

I.8 Summary of Results

The robust estimators perform considerably better than the standard estimators when applied on the contaminated data sets to estimate μ, λ and σ. Only in the case of consecutive errors and estimating σ, the standard estimator performed better. This was because the errors in estimating λ and s^2 canceled each other out. The robust estimator, however, performed better to estimate λ and s^2 separately. But the errors did not cancel out as much as for the standard estimator and therefore performed worse in estimating σ.

For the 1% level the robust estimators worked still very well as the median over all trials was near the true value. Whereas the standard estimators already had problems and mostly over- or underestimated the parameters considerably. For the 10% and 20% level both the robust estimators or standard estimators mostly over- or underestimate the true value. However, the robust estimators performed in most cases clearly better than the standard estimators with the exception for σ for the consecutive error.

For the clean data sets the standard estimators performed better than the robust estimators for estimating μ, λ and σ. The robust estimators still performed satisfactory.

Conclusions

This thesis shows how robust estimators for the parameters of an Ornstein-Uhlenbeck process can be implemented. In addition, it examines the performance of the robust estimators compared to the standard estimators by considering data sets created from an Ornstein-Uhlenbeck process with different levels of contamination and different contamination types. The robust estimators performs better than the standard estimators when the data sets were contaminated, except for the estimation of one parameter for one type of contamination. The standard estimators performed better on the clean data sets. The performance of the standard estimators on contaminated data was in most cases considerably worse than the performance of the robust estimators. The difference on the clean data sets was not that striking.

On the lowest level of contamination, the robust estimators still performed very well. Its median was near the true value of the parameter. Whereas the standard estimators often already over- or underestimated the parameter. For the higher contamination levels, the robust and the standard estimators over- or underestimated the true value of the parameter in most cases. However, the robust estimators performed in most cases considerably better than the standard estimators.

The comparison between the standard and robust estimators was conducted on data sets obtained from one specific set of parameters μ, σ, λ of the Ornstein-Uhlenbeck process on a regular time grid and settings of d, n to obtain samples of data points of the process. It shows that the robust estimators can be implemented and it can be more robust to contamination than the standard estimators.

Bibliography

[Dowell and Jarratt, 1971] Dowell, M. and Jarratt, P. (1971). A modified regula falsi method for computing the root of an equation. *BIT Numerical Mathematics*, 11(2):168–174.

[Falk, 1997] Falk, M. (1997). On mad and comedians. *Annals of the Institute of Statistical Mathematics*, 49(4):615–644.

[Manos Papadakis and Chatzipantsiou, 2021] Manos Papadakis, Michail Tsagris, M. D. S. F. I. T. M. F. G. B. J. B. C. Z. K. L. and Chatzipantsiou, C. (2021). *Rfast: A Collection of Efficient and Extremely Fast R Functions*. R package version 2.0.3.

[R Core Team, 2020] R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[Rieder, 2012] Rieder, S. (2012). Robust parameter estimation for the ornstein–uhlenbeck process. *Statistical Methods & Applications*, 21(4):411–436.

Appendix

I.8.1 R Code

```
library(Rfast)

g <- function(p,n){
  Sigma <- matrix(c(1,p,p,1),2,2)
  X_Y <- mvrnorm(n=n, rep(0,2),Sigma)
  g_value <- median(X_Y[,1]*X_Y[,2])
  return(g_value)
}
g_Cholesky <- function(p,n){
  Sigma <- matrix(c(1,p,p,1),2,2)
  X_Y <- mvrnorm(n=n, rep(0,2),matrix(c(1,0,0,1),2,2))
  X_Y <- X_Y%*%chol(Sigma)
  g_value <- median(X_Y[,1]*X_Y[,2])
  return(g_value)
}



g_inv <- function(value,m,tol=1e-06){
  iteration <- 0
  X_Y <- rmvnorm(n=m, rep(0,2), matrix(c(1,0,0,1),2,2))
  if(value>=-qnorm(3/4)^2 && value<= qnorm(3/4)^2){
    if(value>0){
      beta<-1
      alpha<-0
      h_alpha <- value
      h_beta <- value -qnorm(3/4)^2
      gamma <- 1 - h_beta/(h_beta - h_alpha)
      Sigma <- matrix(c(1,gamma,gamma,1),2,2)
      X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
      h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
      while(abs(h_gamma)>tol){
        iteration <- iteration +1
        if(h_gamma*h_alpha<0){
          beta <- gamma
          h_beta <- h_gamma
          gamma <- beta - h_beta*(beta-alpha)/(h_beta - h_alpha)
          Sigma <- matrix(c(1,gamma,gamma,1),2,2)
          X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
          h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
```

```
        }else{
          alpha <- gamma
          h_alpha <- h_gamma
          gamma <- beta - h_beta*(beta-alpha)/(h_beta - h_alpha)
          Sigma <- matrix(c(1,gamma,gamma,1),2,2)
          X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
          h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
        }
        if(iteration > 1000){
          print("Max Int")
          return(0)
        }
      }
    }else if(value<0){
      value <- -value
      beta<-1
      alpha<-0
      h_alpha <- value
      h_beta <- value -qnorm(3/4)^2
      gamma <- 1 - h_beta/(h_beta - h_alpha)
      Sigma <- matrix(c(1,gamma,gamma,1),2,2)
      X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
      h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
      while(abs(h_gamma)>tol){
        iteration <- iteration +1
        if(h_gamma*h_alpha<0){
          beta <- gamma
          h_beta <- h_gamma
          gamma <- beta - h_beta*(beta-alpha)/(h_beta - h_alpha)
          Sigma <- matrix(c(1,gamma,gamma,1),2,2)
          X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
          h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
        }else{
          alpha <- gamma
          h_alpha <- h_gamma
          gamma <- beta - h_beta*(beta-alpha)/(h_beta - h_alpha)
          Sigma <- matrix(c(1,gamma,gamma,1),2,2)
          X_Y_gamma <- mat.mult(X_Y,chol(Sigma))
          h_gamma <- value - median(X_Y_gamma[,1]*X_Y_gamma[,2])
        }
        if(iteration > 1000){
          print("Max Int")
          return(0)
        }
      }
      gamma <- - gamma
    }else{
      gamma <- 0
    }
    return(gamma)
  }else{
    return("Value is out of range")
  }
}
```

```
95
96  RV_for_g_inv <- function(m){
97    X_Y <- mvrnorm(n=m, rep(0,2),matrix(c(1,0,0,1),2,2))
98    return(X_Y)
99  }
100
101
102
103 Simulate_X <- function(mu,sigma,lambda,d,n){
104   X0 <- rnorm(1,mu,sd=sigma)
105   X<-c(X0,rep(0,(n-1)))
106   for(i in 2:n){
107     Y <- rnorm(1,mean=0,sd=sqrt((1-exp(-2*lambda*d))/(2*lambda)))
108     X[i]<-X[i-1]*exp(-lambda*d)+mu*(1-exp(-lambda*d))+sigma*Y
109   }
110   return(X)
111 }
112 Calc_delta <- function(X){
113   X_hat <- median(X)
114   estimate <- median((X[-1]-X_hat)*(X[-length(X)]-X_hat))
115   /median(abs(X-X_hat))^2
116   return(estimate)
117 }
118
119
120 Calc_rho <- function(X,m,tol=1e-06){
121   delta <- Calc_delta(X)
122   rho <- g_inv(delta*qnorm(3/4)^2,m=m,tol=tol)
123   return(rho)
124 }
125
126 Robust_estimators <- function(X,d,m,tol=1e-06){
127   mu <- median(X)
128   s <- median(abs(X-mu))/qnorm(3/4)
129   rho <- Calc_rho(X,m,tol)
130   lambda <- -log(rho)/d
131   sigma_2 <- 2*lambda*s^2
132   return(c(mu,sqrt(sigma_2),lambda))
133 }
134
135
136
137
138
139
140 Standard_estimators <- function(X,d){
141   mu <- mean(X)
142   s <- sd(X)
143   rho <- acf(X,lag.max=1,plot=FALSE)
144   lambda <- -log(rho[[1]][2])/d
145   sigma_2 <- 2*lambda*s^2
146   return(c(mu,sqrt(sigma_2),lambda))
147 }
148
```

```r

mu_1 <- 0
sigma_1 <- 1
lambda_1 <- 0.5

#Test for 5000 different Trails - Clean Data
k<-5000
Result <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1 <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,d=1,n=1000)
  RE1 <- Robust_estimators(X_1,1,m=10000)
  SE1 <- Standard_estimators(X_1,1)
  Result[i,c(1,2,3)]<-RE1
  Result[i,c(4,5,6)]<-SE1
}


#Test for 5000 different Trails - Contaminated Data - Cauchy
#_ Level 1 (1%)

k<-5000
Result_Cont_Cauchy_Level_1 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,
                         d=1,n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.01)
  X_1_cont[random_cont_1] <- rcauchy(length(random_cont_1),
                                     loc=0,scale=0.5)
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cauchy_Level_1[i,c(1,2,3)]<-RE1
  Result_Cont_Cauchy_Level_1[i,c(4,5,6)]<-SE1
}


#Test for 5000 different Trails - Contaminated Data - Cauchy _
#Level 2 (10%)

k<-5000
Result_Cont_Cauchy_Level_2 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,d=1,
                         n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.1)
  X_1_cont[random_cont_1] <- rcauchy(length(random_cont_1),loc=0,
                                     scale=0.5)
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cauchy_Level_2[i,c(1,2,3)]<-RE1
  Result_Cont_Cauchy_Level_2[i,c(4,5,6)]<-SE1
}
```

```r

#Test for 5000 different Trails - Contaminated Data - Cauchy _
#Level 3 (20%)

k<-5000
Result_Cont_Cauchy_Level_3 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,d=1,
                         n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.2)
  X_1_cont[random_cont_1] <- rcauchy(length(random_cont_1),loc=0,
                                     scale=0.5)
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cauchy_Level_3[i,c(1,2,3)]<-RE1
  Result_Cont_Cauchy_Level_3[i,c(4,5,6)]<-SE1
}
#############
#Test for 5000 different Trails - Contaminated Data -
#Consecutive Gross Error -
#Level 1 (1%) - 5 Sigma

k<-5000
Result_Cont_Cons_Gross_Level_1 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,d=1
                         ,n=1000)
  cont <- 1000*0.01
  random_cont_1 <- sample(1:(1000-cont), 1)
  X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] <-
    X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] +
    5*sigma_1
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cons_Gross_Level_1[i,c(1,2,3)]<-RE1
  Result_Cont_Cons_Gross_Level_1[i,c(4,5,6)]<-SE1
}
#Test for 5000 different Trails - Contaminated Data -
#Consecutive Gross Error
#Level 2 (10%)

k<-5000
Result_Cont_Cons_Gross_Level_2 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,
                         d=1,n=1000)
  cont <- 1000*0.1
  random_cont_1 <- sample(1:(1000-cont), 1)
  X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] <-
    X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] +
    5*sigma_1
```

```
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cons_Gross_Level_2[i,c(1,2,3)]<-RE1
  Result_Cont_Cons_Gross_Level_2[i,c(4,5,6)]<-SE1
}
#Test for 5000 different Trails - Contaminated Data -
#Consecutive Gross Error
#_ Level 3 (20%)

k<-5000
Result_Cont_Cons_Gross_Level_3 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1,
                         d=1,n=1000)
  cont <- 1000*0.2
  random_cont_1 <- sample(1:(1000-cont), 1)
  X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] <-
    X_1_cont[seq(random_cont_1,random_cont_1+(cont-1),by=1)] +
    5*sigma_1
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Cons_Gross_Level_3[i,c(1,2,3)]<-RE1
  Result_Cont_Cons_Gross_Level_3[i,c(4,5,6)]<-SE1
}
##########
#Test for 5000 different Trails - Contaminated Data -
#Gross Error _ Level 1 (1%)

k<-5000
Result_Cont_Gross_Level_1 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,
                         lambda=lambda_1,d=1,n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.01)
  X_1_cont[random_cont_1] <- X_1_cont[random_cont_1] +
    5*sigma_1
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Gross_Level_1[i,c(1,2,3)]<-RE1
  Result_Cont_Gross_Level_1[i,c(4,5,6)]<-SE1
}

#Test for 5000 different Trails - Contaminated Data - Gross Error _
#Level 2 (10%)

k<-5000
Result_Cont_Gross_Level_2 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=
                           lambda_1,d=1,n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.1)
```

```
  X_1_cont[random_cont_1] <- X_1_cont[random_cont_1] + 5*sigma_1
  RE1 <- Robust_estimators(X_1_cont,1,m=10000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Gross_Level_2[i,c(1,2,3)]<-RE1
  Result_Cont_Gross_Level_2[i,c(4,5,6)]<-SE1
}
#Test for 5000 different Trails - Contaminated Data - Gross Error _
#Level 3 (20%)

k<-5000
Result_Cont_Gross_Level_3 <- matrix(data=1,k,6)
for(i in 1:k){
  cat("Simulation ",i," of ",k,"\n")
  X_1_cont <- Simulate_X(mu=mu_1,sigma=sigma_1,lambda=lambda_1
                         ,d=1,n=1000)
  random_cont_1 <- sample(1:1000, 1000*0.2)
  X_1_cont[random_cont_1] <- X_1_cont[random_cont_1] + 5*sigma_1
  RE1 <- Robust_estimators(X_1_cont,1,m=100000)
  SE1 <- Standard_estimators(X_1_cont,1)
  Result_Cont_Gross_Level_3[i,c(1,2,3)]<-RE1
  Result_Cont_Gross_Level_3[i,c(4,5,6)]<-SE1
}
```