University of Al-Qadisiyah

College of Administration & Economics

Statistics department

Feature selection in ARMA models with application

A thesis submitted to the council of the college of Administration & Economics\ University of Al-Qadisiyah as partial fulfillment of the requirements for the degree of master in Statistics

By

Alaa Qasim Yaseen

Supervised by

Prof. Dr. Tahir R. Dikheel

1440 A.H. 2019 A.D.

The decision if the defense committee

We are the head and members of the defense committee certify that we have been looked at the thesis entitled(Feature selection in ARMA models with application) and we have debated the student (Alaa Qasim Yaseen). As a result, the student has defended her thesis and all its content. So that we have found the thesis is worthy to be accepted to award the degree of Master in statistics sciences.

Prof. Dr. Salah Hamza Abid

(The head of the committee)

Prof. Dr. Rahim Jabbar Alhamzawi

(Member)

Asst. Prof. Dr. Jassim N. Hussain

(Member)

Prof. Dr. Tahir Reisan Dikeel

(Member and supervisor)

Approval of the college committee

The council of the college of administration and economics at the University of Al-qadissiah have been approved on the decision of the defense committee.

Prof. Dr. Ali Jawad Kadhim

The Dean if the college of administration and economics / Al-qadissiah university

Abstract

The penalized least square methods that have been used in time series are suitable and common methods for dealing with high-dimensional data, especially when the number of explanatory variables is larger than the sample size. One of advantages of these methods is to ensure high accuracy prediction as well as to get the estimation process and variable selection in one time, where they reduce some coefficients and make of others equal to zero. They produces a model sparse, a model that contains as few variables as possible and is easily interpretable. In spite of these advantages enjoyed by the penalized least square methods, they are non robust methods, in the sense of being affected by the outliers. In order to overcome this problem, the loss function of the penalized least square is replaced by robust loss function to obtain the penalized robust methods. The resulting estimator is the penalized robust estimator that can deal with the problems of dimensions and outliers and also takes the lag effect. In this thesis, it is suggested that the penalized robust estimators are M-lag weighted lasso and MM-lag weighted lasso to obtain high-efficiency estimators in time series. In order to determine the superiority of these estimators, the simulation is performed based on the weighted relative prediction error (RPE^w) criterion. It is concluded that the proposed estimators achieved high efficiency compared to the other estimators for different samples sizes. As for the applied side, real medical data are used for people who are suffering from lung cancer. Data were collected from Diwaniyah Hospital. One of the reasons of lung cancer is the presence of chemical pollutants in water. It has been concluded that the proposed estimators are best in determining the coefficients affecting lung cancer based on the lowest RPE^w value.

بِسْمِ اللّهِ الرَّحْمَنِ الرَّحِيمِ

الرَّحْمَنُ ﴿١﴾ عَلَّمَ الْقُرْءَانَ ﴿٢﴾ خَلَقَ الإِنسَانَ ﴿٣﴾ عَلَّمَهُ الْبَيَانَ
﴿٤﴾ الشَّمْسُ وَالْقَمَرُ بِحُسْبَانٍ ﴿٥﴾ وَالنَّجْمُ وَالشَّجَرُ يَسْجُدَانِ ﴿٦﴾
وَالسَّمَاءَ رَفَعَهَا وَوَضَعَ الْمِيزَانَ ﴿٧﴾ أَلاَّ تَطْغَوْا فِي الْمِيزَانِ ﴿٨﴾ وَأَقِيمُوا
الْوَزْنَ بِالْقِسْطِ وَلا تُخْسِرُوا الْمِيزَانَ ﴿٩﴾

صَدَقَ اللهُ العلي العَظِيم

سورة الرحمن المباركة (الآيات ١–٩)

Dedication

To my God, may my work benefit me on the day of roses

To the Seal of the Prophets Muhammad (peace be upon him)

To the righteous servants of Allah

To my supervisor and teacher

To my illustrious teachers

To my paradise in the earth (my mother)

To my support and My destiny (my father)

To my family (my brother and sisters)

To all whom helped me to complete this thesis

Thanks and appreciation

This work is nearing completion, and before finish it, I must commend him for his great thanks and appreciation to those who had great credit for his accomplishment and start them with those who were present with me wherever I found my God, my Creator and my Lord, and then the supervisors and teachers of Professor Dr. Tahir R. Dikheel who had the magical touches that do not there stands a dilemma, yes teacher, teacher and father.

I also extend my thanks and appreciation to my professors who enlightened me through science and knowledge until I became what I am now. I do not forget the scientific seminar committee which played a big role in drawing the general lines of this thesis.

I would also like to thank Mr. Chairman and the members of the discussion committee who will have the greatest impact in evaluating this thesis to show its final form.

I will not forget to extend my thanks and great gratitude to my mother and father of expensive, who have been willing and willing to endure what I cannot reach to what I am now. I hope that I have fulfilled even a small amount of fulfillment for them.

I would also like to thank my dear brother and sisters for those who stood beside me in all stages of my life, and they never gave me anything.

And to all those who have helped me and forgot to mention it, I extend to all my thanks.

and thank Allah the god of everything.

Contents

Content tables

Content figures

Chapter one

Introduction

Literature Review

Thesis aims

1-1 Introduction

Variable selection plays a significant role in building a time series model. This technique provides shrinkage for good estimation of parameters by penalized regression, good prediction and identification of the relevant variables. The statistical procedures for selecting variables are characterized by the provision of interpretable models. Variable selection methods such as stepwise and best subset selection may suffer from a lack of stability. To overcome this problem, Tibshirani (1996) had introduced the "least absolute shrinkage and selection operator" (lasso). This process provides shrinkage coefficients toward zero and makes some coefficients exactly zero, thus it tries to keep the relevant variables with strong influences. Researchers have found that the "lasso estimator" is sometimes inefficient and the results of the variables selection are inconsistent (Fan & Li, 2001; Yuan & Lin, 2007; Zou, 2006). To overwhelm this problem, Zou (2006) has proposed the adaptive least absolute shrinkage and selection operator (alasso), which penalizes different regression coefficients by different weights. These disciplines reveal the size of the coefficient to define the correct model in the regression. However, these methods cannot deal with the lag effect, which is a base stone in the time series models. Recently, alasso has been used in time series (see, for example, Nardi & Rinaldo,2011;Chen &Chan, 2011;Liu, 2014; Medeiros & Mendes, 2015). However, the alasso in time series cannot reflect certain properties of a time series model, such as the lag effect. To enhance the exactitude of a time series model, Park & Sakaori (2013) introduced a "lag-weighted lasso" (lwlasso). The (lwlasso) requires different penalizes on each coefficient based on weights that indicate not the only coefficient size but also the lag effects. Under the lwlasso, the regression coefficient vector can be measured as follows:

$$\beta_{lwlasso} = arg\,min_{\beta}\{\|y_t - X\beta\|^2 + \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} w_{j,l} \left|\beta_{j,l}\right|, \qquad (1-1)$$

where x_t is the matrix of explanatory variables, β is the regression coefficient vector, λ is the tuning parameter and w is a weighted function.

Equation (1-1) depends on the following three types of weight:

$$\hat{w}_{j,l}^{(1)} = \frac{1}{[\alpha(1-\alpha)^l]^\gamma} \qquad (1-2)$$

$$\hat{w}_{j,l}^{(2)} = \frac{1}{\alpha(1-\alpha)^l \left[\left|\hat{\beta}_{j,l}\right|\right]^\gamma} \qquad (1-3)$$

$$\hat{w}_{j,l}^{(3)} = \frac{1}{\left[\alpha(1-\alpha)^l \left|\hat{\beta}_{j,l}\right|\right]^\gamma} \qquad (1-4)$$

Where $0 < \alpha < 1, \gamma > 0$. Previous studies and all types of data, including time series, have shown that the presence of outliers leads to a major defect in the construction of the appropriate model and also has the greatest impact on the methods used in the estimation. There are many types of outliers, for instance the outliers in the data may be the result of an error in collecting, transferring or recording data or there are outliers such as periods of wars, floods, emergency economic conditions, free waves and other causes, as well as the practical practitioners of the statistic should be aware that the existence of one outliers may have a significant impact on coefficients of the time series model, so researchers were interested in statistical study of this problem, which took a wide range in applications of regression models, outliers and parameters estimation in time series models because of reliability or correlation between time series observations.

The researchers indicate that there are two types of outliers that can be seen in time series data: additive outliers (AO) whose effect in the time series is very rapidly disappearing, and innovational outliers (IO) subsequent periods. So unfortunately, the lwlasso method is sensitive to outliers because it depends on ordinary least square (OLS), which is non robust to outliers in the observations and non normal errors. Robust regression such as Huber's criterion (Lambert& Zwald, 2011), LAD (Wang & Jiang, 2007) and quantile regression (Wu & Liu, 2009) has been used recently in variable selection. In this thesis, we have suggested a robust lag-weighted lasso method by replacing quadratic loss function with the robust function.

1-2 Literature Review

In this section, we review the most important studies that we have studied to be the closest to the subject of this research and the interests of the researcher, which were divided into two parts: The first part deals with research that included the penalized methods used in the time series. The second part relates to the research that used the penalized robust methods. The following is a summary of the most important of these Research.

Wang et al in 2007 proposed a penalized robust estimator called lad-lasso abbreviated to ("least absolute deviation for lasso"). In this study, the OLS loss function was replaced by the absolute variable in the penalized regression to give the lad-lasso. This estimation is characterized by a heavy-tailed distribution, or outlier. They also discussed the theoretical characteristics of the proposed estimate.

Park & Sakaori in 2011 depended on the lag weighted lasso method in VAR models to suggest the lag weighted penalization methods. The lag weighted penalization methods impose the deferent penalties on deferent variable and length of lag. Simulation studies show that the lag weighted penalization methods outperform than existing penalization methods in situation that variable effect decreases as the lag increases.

Lambert-Lacroix, & Zwaldin 2011 suggested using the Huber function with the alasso function to give the Huber alasso that is robust against outliers. The researcher used the simulation for the purpose of demonstrating the efficiency of the proposed estimate with the other estimations as (Lad adaptive lasso). The researcher used the simulation in the comparison process and the preferred estimate was obtained.

In the same year, Jung suggested a penalized robust estimator named Wlad-lasso, a brief of the weighted least absolute deviation lasso estimator, indicating that the lad-lasso is influenced by outliers, therefore, the estimator (lad-lasso) has been non robust in the case outliers by a weight function depending on the space of variables.

Medeiros, & Mendes in 2012 had investigated the asymptotic features of the alasso in sparse, high-dimensional, linear time-series models. They have supposed both the number of covariates variables in the model and candidate variables could be able to increase with the number of measurements and the number of candidate variables are, possibly, larger than the number of observations. they manifested the alasso consistently chooses the relevant variables as the number of observations increases (model selection consistency) and has the oracle property, even when the errors are non-Gaussian and hypothetically heteroskedastic. A simulation study and authentic data showed the method performance well in very general settings.

Park & Sakaori in 2013 studied the penalized methods in time series by proposing a lag weighted lasso methods based on weights. These weights reflect the coefficients effect depending on OLS estimators and the lag effect that is necessary in the time series models, subsequently thus reflect the time series properties. The simulation was used to compare these estimators with lasso and alasso estimators that consider the effect of coefficient size only. Through the study simulations and real data, they found that their proposed approach gives better results than the methods that had been compared.

Also Konzen & A. Ziegelmannin 2013 studied the penalized methods in time series where the (weighted lag alasso) method was introduced. This method assigns different weights to each coefficient (depending on the ridge estimators) and lag periods. They found that this method is superior to Monte Carlo applications and mainly to small samples in terms of choice of common variables, parameter accuracy and prediction, as compared to other shrinkage methods. They also found that this method gives better results in practical application .

In the same year Alfons, Croux and Gelper proposed a penalized robust estimator sparse least trimmed (Sparse LTS), which used a loss function with a lasso function to give an estimated Sparse LTS, and BP of the proposed estimate was derived. This estimate has a high BP as well as a decisive estimate against the outliers. An algorithm was also proposed to calculate the proposed estimate. The researchers used the simulation to make a comparison between the efficiency of the suggested estimation with other estimators, and the advantage of the proposed estimation was compared with the rest of the other estimations.

Darwish et alin 2015 proposed a new method called MM Lasso, which could be used in case of dimensions problems and outliers.They used sparse LTS as an initial estimation of the method. The simulation was used to show the preference of the proposed method to the rest of the estimators.

Liu et al in 2016 had been introduced a new penalized method in estimating the parameters of the ARMA model declared as the "doubly adaptive lasso" (dalasso) or PLAC-weighted adaptive lasso, for modelling stationary vector autoregressive processes. The scheme is twofold adaptive in the sense that its adaptive weights that have formulated as functions of the norms of the partial lag autocorrelation matrix function and yule–walker or OLS estimates of a vector time series. Whereby getting order testimony, subset selection and parameter estimation was done in one action, as presented in Monte Carlo examples and real data analysis.

Yi and Huang in 2017 proposed Huber elastic net for the linear regression model and the Quantile regression in the case of high-dimensional data in which the amount of explanatory variables are bigger than the sample size. An algorithm was also recommended for calculating the estimate. The two proposed estimator compared with the elastic net estimator and based on the coordinate majorization descent algorithm, and the proposed estimated advantage was reached.

1-3 Thesis aims

There are two main objectives for this thesis.

1-The first goal is to obtain the best estimation of the parameters of the AR (P) model by comparing a number of penalized robust estimator by simulation and proving the performances of estimators based on the RPE^{w} criterion .

2-The second goal is to determine the most important coefficients affecting lung cancer, depending on the best coefficients obtained in the experimental side.

3- Finally, proposed penalized robust methods to deal with high-dimensional problems, lag effects, and outliers.

Chapter two

The theoretical part

2-1 Introduction

This chapter includes the theoretical part of the ARMA (q, s) model, the high-dimensional ARMA (q, 0) model, and the penalized methods of the ARMA (q, s) model, in case of dimensional problem. Where s is the order of the MA model, and q is the order of the AR model. As well as the presentation of the most important theoretical characteristics of panelized estimator. It also offers a penalized methods capable to dealing with the lag effect. We proposed penalized robust methods to deal with high-dimensional problems, lag effects, and outliers.

2-2 Time series

Time series y_t is a set of observations arranged according to their occurrence in time such as years, seasons, months, days. ..etc. Therefore, it is a historical record that is adopted to build future expectations. In business, we find weekly interest rates, daily stock prices, monthly price indices, annual sales figures, and so forth. In meteorology, we observe high and low daily temperatures, precipitation rates, annual drought indicators, and wind speed are recorded over time. In agriculture, we record annual crop and livestock production figures, soil erosion and export sales are recorded. In biological science, we observe the electrical activity of the heart in periods of a second are recorded. In ecology, we record an abundance of animal species, a list of areas where the study of the time series is endless are recorded.

The purpose of time series analysis is generally to understand or model a random process leading to the emergence of a series of observations and predicting future values of the series based on the history of that series, and perhaps another related series. (Chan & Cryer 2008)

There are two types of time series, the first type is the discrete time series and the second type is the continuous time series. The time series is either to be non stationary in the mean and can be converted into stationary one using differences or is non stationary in variance and can be converted to stationary using transformations. The stationary, time series can be divided into two common types, stationary of the second order and strictly stationary. Time series is said to be stationary of the second order if the first and second moment are known and the mean of the series is constant independent of time and the covariance depends only on the lag l:

$$Cov(y_t, y_{t+l}) = \gamma_l \qquad (2-1)$$

γ_k is the covariance function.

The time series is said to be strictly stationary if for each integer $k \geq 1$, and for any partial set of time $t_1, t_2, \dots, t_l$ and the $y_{t_1}, y_{t_2}, \dots, y_{t_l}$ common distribution is constant by time difference. This means that for any positive integer l and for any integer r:

$$F(y_{t_{1+r}}, y_{t_{2+r}}, \dots, y_{t_{l+r}}) = F(y_{t_1}, y_{t_2}, \dots, y_{t_l}) \qquad (2-2)$$

F is the distribution function (Box and Jenkins 1976, Box and Jenkins 2008,Chatfield 2004, Maronna 2006).

2-3 ARMA models

Stationary time series models was introduced by Yule in (1926). He studied the autoregressive model AR (q). Wlker in (1931) introduced the general model of the autoregressive models. Stutzky (1937) studied models of moving average MA(s) and put it’s general formula. Then completed his way to find the model in a mixed and complete way by Wold in (1938), where he developed these two models with a series of operations into three directions in the estimation procedure and called it the autoregressive_ moving average (ARMA) models. This model is used if data is stationary.

ARMA models are mathematical formulas that represent the continuity pattern of the phenomenon, or the type of correlation between the time series and itself. It is widely used in many sciences such as economics, geography, aviation, agricultural sciences, physical systems and other fields. Models can help us understand how the system works by detecting the properties associated with this system.

ARMA models can be described through a series of equation; these equations are easier if the mean time series is zero by subtracting the sample mean. Therefore, the modified series is treated with the sample mean, meaning that:

$$y_t = Y_t - \bar{Y} \qquad (2-3)$$

Where

Y_t represents the time series.

y_t represents the modified time series.

Thus the ARMA model can be written as:

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_q y_{t-q} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_s e_{t-s} \qquad (2-4)$$

Where

q:represent the order of autoregressive model

s:represent the order of moving average model

$\hat{\beta}$:represent the parameter vector of autoregressive model

$\hat{\theta}$:represent the parameter vector of moving average model

e_t:represent the random error

q and s can be found by looking at the form of the autocorrelation (ACF) function and the partial autocorrelation (PACF) function. When the autocorrelations decay exponentially to zero, this means that the model is an AR model and its order is determined by a number of PACF that are significantly different from zero. If the PACF decay exponentially to zero, the model is an MA model whose order is determined by the number of ACF with statistical significance. If ACF and PACF decay to zero exponentially, this ARMA model is determined by AR & MA. See the ACF and PACF function, if the ACF function does not give up quickly with increasing degrees of delay, it means that the time series is non stationary, and you need to take the differences. Summed up the diagnostic process through the following:(Box and Jenkins 1976, Box and Jenkins 2008,Chatfield 2004, Maronna 2006)

table(2-1): Determine models order according to the behavior of ACF, PACF for the time series stationary.

Model	ACF	$PACF$
$AR(q)$	-Exponential decay, if q=1; -A sine-wave shape pattern or a set of exponential decays, if $q \geq 2$.	-Spike at lag 1,no correlation for other lags, if q=1; -Spikes at lags 1 to p, no correlation for other lags if $q \geq 2$.
$MA(s)$	-Spike at lag 1,no correlation for other lags, if s=1; -Spikes at lags 1 to p, no correlation for other lags if $s \geq 2$.	-Damps out exponentially, if s=1; -A sine-wave shape pattern or a set of exponential decays, if $s \geq 2$.

$ARMA(\mathrm{q}, \mathrm{s})$	-Exponential decay starting at lag q.	-Exponential decay starting at lag s.

the problem here in the case of mixed models, to determine q and s. Because the ACF and PACF functions in this case behave in a similar manner. There are several criteria have been developed to compare the models in the selection process of the model order. Selecting a order lower than the actual order of the model results in the inconsistency of the model parameters While choosing a order higher than the actual order in the model increases the variation of the model, This leads to loss of accuracy. There are several criteria to select the model order such as: Akaike information criterion (AIC), Bayesian Information Criterion (BIC) and etc. (Chan and Cryer 2008)

2-4 Autoregressive distributed lag (ADL) models

There are several models that can be used to describe time series, such as AR, MA and ARMA. AR model is the most common model in time series because most phenomena follow it in practice. Furthermore, the errors in MA models have a non-linear functions of the parameters, so iterative estimation methods are need to minimize the residual sum of squares (Chatfield 1995).

Consider the series y_t, the model of AR(q) can be written with response lag variables and a disturbance error term as:

$$y_t = a + \sum_{l=1}^{q} \beta_l L^l y_t + e_t \,, \qquad (2-5)$$

where a is the intercept, β_l is the l^{th} regression coefficient for $l = 0,1,2,\ldots,q$, e_t refers to white noise term with zero mean, constant variance σ^2 and $cov(\mathrm{e}_t, y_{t-l}) = 0$ for all $l \neq 0$ and L represents the lag operator ($i.e\ L^0 y_t = y_t, L^1 y_t = y_{t-1}$). Sometimes the past series y_{t-l} cannot describe the present value y_t. To overcome this problem and to improve the forecast accuracy, more explanatory variables $x_{j,t-l}(\ j = 1,\ldots,p\ and\ \ l = 0,\ldots,q_j)$ are added (Pesaran & Shin, 1997). Specifically, Pesaran & Shin (1998) proposed an autoregressive distributed lag (ADL) model with response lag variables, current and lagged explanatory variables. The ADL$(q_0, q_1, q_2, \ldots, q_p)$ model can be formulated as:

$$y_t = a + \sum_{l=0}^{q_0} \beta_{0,l} L^l y_t + \sum_{l=0}^{q_1} \beta_{1,l} L^l x_{1,t} + \cdots + \sum_{l=0}^{q_p} \beta_{p,l} L^l x_{p,t} + e_t. \qquad (2-6)$$

Equivalently:

$$y_t = a + \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} + e_t. \qquad (2-7)$$

Or can be written in the form of matrices as follows:

$$\mathrm{Y} = \mathrm{X}\beta + \mathrm{e} \qquad (2-8)$$

Where

$$X = \left[\begin{matrix} x_{0,max(q_0 q_1,\ldots,q_p)} & x_{0,(max(q_0 q_1,\ldots,q_p)-1)} & \cdots & x_{0,(max(q_0 q_1,\ldots,q_p)-q_0)} \\ x_{0,(max(q_0 q_1,\ldots,q_p)+1)} & x_{0,(max(q_0 q_1,\ldots,q_p))} & \cdots & x_{0,(max(q_0 q_1,\ldots,q_p)-q_0+1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,T-1} & x_{0,T-2} & \cdots & x_{0,T-q_0} \end{matrix}\right.$$

$$\begin{matrix} x_{1,max(q_0 q_1,\ldots,q_p)} & x_{1,(max(q_0 q_1,\ldots,q_p)-1)} & \cdots & x_{1,(max(q_0 q_1,\ldots,q_p)-q_0)} & \cdots \\ x_{1,(max(q_0 q_1,\ldots,q_p)+1)} & x_{1,(max(q_0 q_1,\ldots,q_p))} & \cdots & x_{1,(max(q_0 q_1,\ldots,q_p)-q_0+1)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ x_{1,T-1} & x_{1,T-2} & \cdots & x_{1,T-q_1} & \cdots \end{matrix}$$

$$\begin{array}{cccc} x_{p,max(q_0q_1,\ldots,q_p)} & x_{p,(max(q_0q_1,\ldots,q_p)-1)} & \cdots & x_{p(max(q_0q_1,\ldots,q_p)-q_0)} \\ x_{p,(max(q_0q_1,\ldots,q_p)+1)} & x_{p,[(max(q_0q_1,\ldots,q_p))} & \cdots & x_{p,(max(q_0q_1,\ldots,q_p)-q_0+1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p,T-1} & x_{p,T-2} & \cdots & x_{p,T-q_p} \end{array}\Bigg] \quad (2-9)$$

As well as the response lag variable vector (Y) agencies:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} \qquad (2-10)$$

The random error (e) is as follows:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_T \end{bmatrix} \qquad (2-11)$$

$$\beta = \begin{bmatrix} \beta_{0,0} \\ \beta_{0,1} \\ \vdots \\ \beta_{0,q_0} \\ \beta_{1,0} \\ \beta_{1,1} \\ \vdots \\ \beta_{1,q_1} \\ \vdots \\ \beta_{p,0} \\ \beta_{p,1} \\ \vdots \\ \beta_{p,q_p} \end{bmatrix} \qquad (2-12)$$

In the above model, $\beta_{0,0} = 0$ and $x_{0,t} = y_t$ are assumed. In addition, equation (2-7) satisfies the following assumptions: (Park & Sakaori 2013)

1. $E(e_t|y_{t-1}, y_{t-2}, \ldots, x_{1,t}, x_{1,t-l}, \ldots, x_{p,t-1}, x_{p,t-2}, \ldots) = 0$.

2. $(y_t, x_{1,t}, \ldots, x_{p,t})$ are stationary.

3. The correlation coefficients between $(y_t, x_{1,t}, \ldots, x_{p,t})$ and $(y_{t-l}, x_{1,t-l}, \ldots,$

$x_{p,t-l})$ decline as l increase.

2-5 Variables Selection

The statistical model can include a large number of explanatory variables to reduce the bias of the model, because we do not know which of the explanatory variables can affect the dependent variable. However, there are many explanatory variables that may have little or no effect on the dependent variable. Therefore, the main task is to create a model with a few important explanatory variables that match the correct model (Hesterberg et al 2008).

On the other hand, in order to enhance the prediction and select the important variables selection, statisticians typically use phase-out and selection of a subset. Although, it is useful in practice, these selection procedures ignore random errors inherited in the variable selection stages. Here, it is difficult to understand the theoretical properties to some extent. Moreover, the selection of the best subset has several disadvantages, the most serious of which is its instability as analyzed (Fan& Li 2001).

These methods use certain criteria, most of which are based on the sum square errors (SSE), which are defined according to the following formula:

$$\mathrm{SSE} = \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 = \|y - \hat{y}\|^2 \quad , t = 1,2, \ldots, T \qquad (2-13)$$

Where

T: is the sample size.

$\hat{y}_t$: represents the estimated value of the dependent variable of order t. One of these criteria is mean square errors (MSE), which takes the following formula:

$$\mathrm{MSE} = \frac{\mathrm{SSE}}{\mathrm{T}} \qquad (2-14)$$

In (1973), Akiki proposed the Akiki information criterion (AIC), which takes the following formula: (Kutner el al 2005)

$$\mathrm{AIC} = \mathrm{T}\ln(\hat{\sigma}^2) + 2\mathrm{p} \qquad (2-15)$$

where

$$\hat{\sigma}^2 = \mathrm{MSE} \qquad (2-16)$$

Schawarz (1978) proposed another criterion called Bayesian information criterion (BIC) which takes the following formula (Stuart 2011) :

$$\mathrm{BIC} = \mathrm{T}\ln(\hat{\sigma}^2) + ln(T)p \qquad (2-17)$$

2-6 Penalized Least Square Method(PLS)

Penalized least square (PLS) methods is a convenient and common method to deal with high-dimensional data, especially when the number of explanatory variables is greater than the sample size, it is not possible to use the ordinary least square method.

PLS methods are used to overcome computational problems in high-dimensional data as well as improve prediction accuracy (Darwish & Buyuklu 2015).

PLS methods is based on the principle of minimizing the SSE with some limitations on parameters. The estimates of the least square are obtained by reducing the objective function, which consists of two parts, the loss function and the penalty function. which are in accordance with the following formula: (Flexeder 2010)

$$p_{LS}(\lambda, \beta) = (y - X\beta)\acute{}(y - X\beta) + p(\lambda, \beta) \qquad (2-18)$$

where

$p(\cdot)$: penalty function.

$\lambda > 0$: penalty parameter.

Accordingly, the penalized estimator is obtained according to the following formula:

$$\hat{\beta} = \operatorname{argmin}_{\beta}\{p_{LS}(\lambda, \beta)\} \qquad (2-19)$$

PLS methods are the process of estimation and feature selection in the same time (Mylona & Goos 2010).

The good penalty function gives an estimator of three characteristics, including (Fan& Li 2001).

1-Unbiasedness: PLS estimator should be unbiased or almost unbiased when the real anonymous parameter is large.

2-Sparsity: PLS estimator should be has the threshold rule, which sets are estimating with small coefficients to zero.

3-Continuity: PLS estimator should be a continuous function in the data, meaning that little change in the data does not lead to a significant change in the estimates.

Fan & Li also stated that ideally estimator has the characteristics of Oracle Properties, meaning (Fan& Li 2001)

1- Probability Consistency The real model is one when ($T \to \infty$). This property is called Sparsity, i.e.:

$$\lim_{T\to\infty} P_r\left(\hat{\theta}_{An} = \theta\right) = 1 \qquad (2-20)$$

An : refer to that the estimator has asymptotical normal distribution.

2-The estimator has an asymptotical normal distribution as in the case of the Oracle estimator, i.e.:

$$\sqrt{T}\left(\hat{\theta}_{An} - \theta\right) \sim N(0, \Sigma) \qquad (2-21)$$

2-6-1 Least absolute shrinkage and selection operator (lasso)

In (1996), Tibshirani proposed a lasso function, an abbreviation for "least absolute shrinkage and selection operator", to estimate linear model parameters and variable selection together. The main idea of the lasso method is to minimized sum square residuals, plus a constraint representing the absolute sum of the coefficients. For the linear model, the lasso estimator of the ARMA model is obtained according to the following formula: (Tibshirani 1996)

$$\hat{\beta}_{\text{lasso}} = \text{argmin}_{\beta}\left\{\|y_t - X\beta\|^2 + \lambda \sum_{j=0}^{p} \sum_{k=0}^{q_j} \left|\beta_{j,l}\right|\right\} \qquad (2-22)$$

Where

λ: Penalty Parameter or Regularization Parameter

$\lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \left|\beta_{j,l}\right|$: penalty function or regularization function.

lasso is often used in practice because the L_1 penalty bound allows the coefficients to be reduced to exactly zero **(**Konzen & A. Ziegelmann 2013**,** Nardi & Rinaldo,2011**).**

2-6-2 adaptive lasso (alasso)

Obviously studies have found that the lasso estimator may be inefficient and that the results of choosing the real model may be inconsistent. To overcome these problems, Zou (2006) suggested alasso. The alasso method has assigns different penalty bound for each coefficient based on weights. These penalty bound have reflected the size of the parameter for each variable, and alasso is able to determine the correct consistent model and efficient coefficients.

$$\beta_{\mathbf{alasso}} = \arg\min_{\beta} \left\{ ||y_t - X_t\beta||^2 + \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \hat{w}_{j,l} \left|\beta_{jl}\right| \right\} \qquad (2-23)$$

Where, $\hat{w}_{j,l} = \frac{1}{|\hat{\beta}_{j,l}|^{\gamma}}$**,** $\gamma > 0$ and the OLS estimators or the ridge estimators can be used as $\hat{\beta}_{j,l}$. (Zou 2006, Audrino & Camponovo 2013)

2-6-3 The lag weighted lasso (lwlasso)

alasso method can determine the correct form in regression models. However, it can't calculate the lag effect period, which is necessary for a time series model. Thus, it can't reflect the properties of the time series model. To improve the prediction accuracy of the time series model, Park and Sakaori (2013) proposed the Lag weighted lasso (lwlasso) method. lwlasso has imposed different penalty limits for each coefficient on the basis of weights that reflect coefficients size and also the lag effect period such as in equation (1-1). Based on three type of weights which have of two parts, the first reflecting the coefficient size effects and the second the lag effects.

1– Reflecting only the lag effects as in equation (1-2)

2– Reflecting coefficient size and lag effects with γ only affecting the coefficient effects as in equation (1-3)

3– Reflecting coefficient size and lag effect with γ affecting both the coefficient size and lag effects as in equation (1-4).

2-7 Outliers in time series

There are several reasons that lead to the appearance of outliers such as measurement errors, registration errors, blogging and sample errors or external interventions. These reasons can have an effect in determining the outliers form or body to be isolated or patchy, for example we often see economic and commercial time series affected by turbulence, outbreaks of wars, sudden changes in the structure of the market, changes in the modernity of equipment and machinery for some commodities, changes in weather, etc. There are three methods to deal with outliers (Maronna et al 2006)

Method 1: Deleting

Method 2: Adaptation (robust estimation of the model)

Method 3: Detect, model and interpret

Previously, the outliers were deleted, but now many researchers do not recommend this procedure. If we want a proper model, the outlier observations are replaced by some observations. If the identification and estimation of individual outliers or patchy outliers is not necessary for analysis, the goal is to build a model for most of the observations the robust method (Method 2) is appropriated. So the use of robust methods provides protection against the side effects of outliers, because seeing one outliers affect negatively and greatly on OLS estimators.

If the goal is to identify a highly efficient model of prediction and interpretation, the third method is the appropriate method most of researchers prefer the method of integration between the second and third methods, especially in practice

2-8 Masking, swamping, and smearing

The effect of outliers involves three types: Masking, swamping, and smearing (Bruce and Martian 1989).

1-Masking

Masking means that outliers covers or masks the effects of other outliers, therefore the method of statistical estimation or the method of testing outliers fails to detect it. Bruce and Martian (1989) suggested that the diagnosing method the exclusion k of outliers (leave k out) to deal with patchy outliers and the method of repetitive deletion of the Masking effects, but the other problem that emerged was how to determine the k value.

2-Swamping

Swamping is the opposite state of Masking. A group of outliers may be identified as good observations or identify a set of good observations as outliers.

3-Smearing

Smearing is the outliers effect on neighboring observations because of the chain correlation of the time series, so the impact of pollution will depend on the type of outliers.

2-9 Outliers Types in time series

Fox in (1972) classified the outliers in the time series into two categories, the AO and IO. The authors Bruce and Martian (1989) presented a study that showed the effect of AR(1) on the presence of outliers of AO (additive outliers) and IO (innovational outlier).

Tsay in (1988) has classified the abnormal cases that face us in the time series analysis to outliers and structure changes or change of the series and showed that it is a common thing in the applied time series analysis, some existence or neglect time of their impact beins, so it leads to wrong conclusions. Tsay divided outliers in to the AO and IO, and he divided structure changes into the level shift (LS) and the variance change (VC) and the transformation is classified as a permanent level change (PLC) and a transient level change (TLC).

Maronna et al in (2006) has introduced a similar classification of Tsay. They divided the abnormal cases that can be encountered in the time series analysis to isolated outliers which are odd values that appears in the time series, and patchy outliers, which are outliers group that spread in the time series, and LS in mean value. LS isn't mentioned as outliers.

AO is the type of outliers that affects only in one observation after that, the series returns to its normal course as if nothing had happened. IO is the type of outliers that affects subsequent observations from an outlier site, or outliers whose influence is continued to other observations. This effect depends on the strength of the system or model memory. These two types of outliers are common in time series.

2-10 Penalized robust methods for time series

The main objective of penalized robust methods is to provide accurate results rather than penalized least squares methods in the presence of outliers. To reach this goal, deleting the outliers, limiting the influence of outliers by the robust method of reducing the weight of outliers, changing the value of outliers, and penalized robust estimation techniques are used. The general formulation of the penalized robust methods in the time series is as follows: (Fan & Li (2001))

$$\sum_{t=1}^{T} \rho\left(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} \right) + p_\lambda^R(|\beta|), \qquad (2-24)$$

Where k= 1.34, $\rho(.)$ is the Huber loss function (Huber, 1981) defined by:

$$\rho(u) = \begin{cases} u^2 & if\ u \le d \\ 2d|u| - d^2 & if\ u > d, \end{cases} \qquad (2-25)$$

and $p_\lambda^R(|\beta|)$ is the penalty function and d is the tuning constant. Here, R refers to the robust proposed penalty.

The efficiency properties, as well as the breakdown point (BP), are used as a measure to determine the effectiveness of penalized robust methods. The BP is a measure of the resistance of an estimator when the data have a large ratio of contamination. The least squares estimator has BP as low as $1/n$, meaning that even a single outlying observation can turn out an estimator of OLS to be useless. In contrast, there are some estimators that have a high BP of approximately 50%.

2-10-1 M-lag weighted lasso (M-lwlasso) method

The least squares method with penalty function in lwlasso is unprotected because it is affected by the abnormal values. Therefore, in this thesis, we replace the loss function in lwlasso with the M robust loss function (have about a BP of 0.5), then we replace the penalty function in lwlasso with the M robust penalty function, to obtain the M-lwlasso method, which is defined as:

$$\hat{\beta}_{M-lwlasso} = arg\ min_{\beta} \sum_{t=1}^{T} \rho\left(\frac{e_t}{s}\right) + p_{\lambda}^{R}(|\beta|). \qquad (2-26)$$

Equivalently:

$$\hat{\beta}_{M-lwlasso} = arg\ min_{\beta} \sum_{t=1}^{T} \rho\left(\frac{y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t}}{s}\right) + p_{\lambda}^{R}(|\beta|), \qquad (2-27)$$

where s is robustly found, according to the following formula: (Maronna et al 2006)

$$s = \frac{MAD}{0.6745} = \frac{median|e_t - median(e_t)|}{0.6745}, \qquad (2-28)$$

and $p_{\lambda}^{R}(|\beta_{j,l}|)$ is the M-lwlasso penalty function with the following form:

$$p_\lambda^R(|\beta|) = \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \widehat{w}_{j,l}^R \left|\beta_{j,l}\right|. \qquad (2-29)$$

as the three weight types, (1-2), (1-3), and (1-4), reported in Park & Sakaori (2013) are not robust, we propose the following robust weights:

$$\widehat{w}_{j,l}^{R1} = \frac{1}{[\alpha(1-\alpha)^l]^\gamma} \qquad (2-30)$$

$$\widehat{w}_{j,l}^{R2} = \frac{1}{\alpha(1-\alpha)^l \left[\left|\widehat{\beta}_{j,l}^R\right|\right]^\gamma} \qquad (2-31)$$

$$\widehat{w}_{j,l}^{R3} = \frac{1}{\left[\alpha(1-\alpha)^l \left|\widehat{\beta}_{j,l}^R\right|\right]^\gamma}, \qquad (2-32)$$

where $\widehat{\beta}_{j,l}^R$is the $(j,l)^{t\square}$ element in $\widehat{\beta}^R$, which can be estimated as follows:

$$\widehat{\beta}^R = arg\ min_{(\beta)} \left\{ \sum_{t=1}^{n} \rho\left(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} \right) \right\}. \qquad (2-33)$$

To summarize, the proposed M-lwlasso estimators are listed in Table (2-2).

Table (2-2) summaries the proposed M-lwlasso estimators using robust technique

Weight	M-lwlasso estimator
$\boldsymbol{w^{R1}}$	$\hat{\beta}_{\text{M-lwlasso}} = arg\ min_\beta \sum_{t=1}^{T} \rho\left(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} \right) + \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \widehat{w}_{j,l}^{R1} \left|\beta_{j,l}\right|$

$\boldsymbol{w^{R2}}$	$\hat{\beta}_{\text{M-lwlasso}} = arg\ min_{\beta} \sum_{t=1}^{T} \rho\left(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} \right) + \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \widehat{w}_{j,l}^{R2} \lvert \beta_{j,l} \rvert$
$\boldsymbol{w^{R3}}$	$\hat{\beta}_{\text{M-lwlasso}} = arg\ min_{\beta} \sum_{t=1}^{T} \rho\left(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t} \right) + \lambda \sum_{j=0}^{p} \sum_{l=0}^{q_j} \widehat{w}_{j,l}^{R3} \lvert \beta_{j,l} \rvert$

In practice, the choice of tuning parameters is important. K-fold cross validation is used widely in selecting the tuning parameters (Bengio & Grandvalet, 2004; Rodriguez et al., 2010). Here, we find optimal tuning parameters (a, γ, λ) by 10-fold cross validation. Then, we compare the forecast accuracy of each method based on the weight's relative prediction error (RPE^w):

$$RPE^w = E[(\hat{y}_t - y_t^w)^2]/(\hat{\sigma}^w)^2 , \qquad (2-34)$$

$$\hat{\sigma}^w = \frac{\sum_{t=1}^{T}(y_t^w - \bar{y}_t^w)^2}{T-1}, \ y_t^w = y_t * \square(y_t - \sum_{j=0}^{p} \sum_{l=0}^{q_j} \beta_{j,l} L^l x_{j,t}),$$

where $\square(.)$ is the Huber weight function defined by:

$$\square(u) = \begin{cases} 1 & if\ |u| \le d \\ \frac{d}{|u|} & if |u| > d \end{cases},$$

we can summarize the algorithm of proposed methods by the following steps:

Step1: Compute $\hat{\beta}^R$ as in equation (2-33).

Step2: Depending on the $\hat{\beta}^R$ we find the weights $\hat{w}_{j,l}^{R1}$, $\hat{w}_{j,l}^{R2}$ and $\hat{w}_{j,l}^{R3}$.

Step3: Define $x_{j,t-l}^{*} = \frac{x_{j,t-l}}{\hat{w}_{j,l}^{R}}$.

Step4: Solve the problem for all λ as in Equation (2-27).

Step5: Output $\beta_{j,t-l} = \frac{\beta_{j,t-l}^{*}}{\hat{w}_{j,l}^{R}}$

Step6: Compute RPE^{w} as in Equation (2-34).

2-10-2 MM-lwlasso methods

The least squares method with penalty function in lwlasso is unprotected because it's affected by the abnormal values. Therefore, in this thesis, we replace the loss function in lwlasso with the MM robust loss function (have about a BP of 0.5 and supreme efficiency 95% of OLS efficiency under basic assumptions), then we replace the penalty function in lwlasso with the MM robust penalty function, to obtain the MM-lwlasso method. The MM-lwlasso estimation procedure is summarized as follows:

1-Compute the initial estimates of the coefficients $\hat{\beta}_{ini}$ and the corresponding residuals

$$e_t^{(1)} = e(\hat{\beta}_{ini}) = y_t - \acute{x}_t\hat{\beta}_{ini} \qquad (2-35)$$

Where $, t = 1,2, \dots, T$ by employing a high BP estimator such as M-lwlasso estimators with the Huber or bisquare weight function.

2-Calculate the M-lwlasso estimation of the scale of residual $\hat{\sigma}_e$ using the results from step 1.

3-The residuals (in step1) and the scale (in step2) are employed in the first iteration of WLS to find the M-lwlasso estimates of the parameters.

$$\sum_{t=1}^{T} w_t \left(\frac{e_t^{(1)}}{\hat{\sigma}_e} \right) x_t = 0 \qquad (2-36)$$

Where w_t can be chosen as Huber or bisquare weight function.

4-Compute new weights $w_t^{(2)}$ by utilizing the residuals in step3.

5-the $\hat{\sigma}_e$ is kept fixed in step2, then step3 and step4 are reiterated until convergence is involve ($\frac{e_t^{(j)} - e_t^{(j-1)}}{e_t^{(j-1)}}$ as little as possible).

2-11 Selection of penalty parameter

The basic step is the selection of penalty parameter, also called the regularized parameter, symbolized by λ.

In spite of PLS estimator have the oracle properties, the choice of the penalty parameter is very important. It controls the shrinkage and the sub-variables selection included in the final model, (Alfons et al 2013).

When the penalty parameter is $\lambda = 0$, the estimators of PLS are identical to the estimates of OLS. If the value of λ parameter is large, it leads to a situation in which the reduction and sub-sets selection is exaggerated until $\lambda \rightarrow \infty$ all coefficients will be forced to be zero. On the other hand, the small value of λ parameter leads to an opposite matter, since the amount of reduction of the coefficients is very small, and sub-variables selection may include some of the non-useful variables (Sartori 2011).

There are some criteria used in selecting the penalty parameter, including the following:

2-11-1 Cross-validation Criterion(CV)

One of the most commonly used criterion for selecting a penalty parameter is cross-validation (CV) criterion, where the forensic criterion is obtained by estimating the predictive mean squares error (PMSE).

K-fold cross validation is a common method for estimating predictive error and comparing different models. Where the data (x_t, y_t) is divided into k of equal parts. Therefore, for $k = 1,2, \ldots, K$, the k_{th} portion of the data set is removed and the model $f^{-k}(x, \lambda)$ is met. Let C_k represent the indicator observation in the k_{th} field. Therefore, CV criterion for the expected test error is according to the following formula:

$$CV(\lambda) = \frac{1}{T}\sum_{k=1}^{K}\sum_{t \in C_k}\left\|y_t - f^{-k}(x_t, \lambda)\right\|^2 \quad (2-37)$$

The λ parameter, which makes the CV (λ) at its lower end, is selected. (Tibshirani, R. J., & Tibshirani, R. 2009)

2-11-2 Generalized cross-validation criterion(GCV)

The other method to choose the penalty parameter is Generalized Cross-validation Criterion (GCV), and the general formula of GCV is as follows: (Fan& Li 2001)

$$\mathrm{GCV}(\lambda) = \frac{\left\| \mathrm{Y} - \mathrm{X}\hat{\beta}_{\lambda} \right\|^{2}}{\mathrm{T}\left(1 - \frac{\mathrm{df}(\lambda)}{\mathrm{T}}\right)^{2}} \qquad (2-38)$$

Where

$\mathrm{df}(\lambda)$: represent the degrees of freedom and represent the number of estimated non-zero coefficient .

The penalty parameter (λ) that makes GCV (λ) at its lower end, is selected.

2-11-3 Bayesian information criterion(BIC)

This criterion is also called Schwarz criterion widely used in selecting the λ parameter because it requires a little computational effort, and (BIC) is calculated according to the following formula: (Alfons et al 2013)

$$\mathrm{BIC}(\lambda) = \log\left(\hat{\sigma}^{2}\right) + \mathrm{df}(\lambda)\frac{\log(\mathrm{T})}{\mathrm{T}} \qquad (2-39)$$

The λ parameter, which makes BIC (λ) at its lower end, is selected.

Chapter Three

Simulation part

3-1 Introduction

In order to compare the ordinary panelized estimators with the penalized robust estimators of time series in the theoretical side, the simulation technique was adopted. We generate the random error with zero mean and different variance normal distribution and different sample sizes, then contaminating it with different percentages of outlier. The outliers are generated with a zero mean and different variance and distributions. In order to obtain a large number of experiments which are more comprehensive in terms of sample sizes, the PLS methods and proposed methods were tested on the same data generated. The ability of these methods to resist different percentages of outliers (5%, 10%, 15%, 20%), the following sizes of the samples (T = 50,100,200), and RPE^{w} criterion, are used to compare these methods. The method with less value of RPE^{w} is the best. We consider the following ADL (5,3,3) model:

$$y_t = a + \sum_{l=0}^{5} \beta_{0,l} L^l y_t + \sum_{l=0}^{3} \beta_{1,l} L^l x_{1,t} + \sum_{l=0}^{3} \beta_{2,l} L^l x_{2,t} + e_t. \qquad (3-1)$$

The results of the simulation were obtained based on programs written by R in appendix [1] and [2]

3-2 **Simulation steps**

The following steps are used to do the experiments:

1- Generate the time series

Clean and stable data was generated by the model described in equation (3-1) with the following default parameters:

Table (3-1) represents the default parameter values

Cases	Stander deviation	T	Parameters
Sparse case	$\sigma = 1$	٥٠	$\beta = (0.3, -0.2, 0.1, 0, 0, 0.9, 0.7, 0.5, 0, 1, -0.7, 0.5, 0)$
		١٠٠	
		٢٠٠	
	$\sigma = 3$	٥٠	
		١٠٠	
		٢٠٠	
	$\sigma = 5$	٥٠	
		١٠٠	
		٢٠٠	
Very sparse case	$\sigma = 1$	٥٠	$\beta = (0.3, 0, 0, 0, 0, 0.9, 0, 0, 0, 1, 0, 0, 0)$
		١٠٠	
		٢٠٠	
	$\sigma = 3$	٥٠	
		١٠٠	
		٢٠٠	
	$\sigma = 5$	٥٠	
		١٠٠	
		٢٠٠	

With a random error followed by normal distribution with zero mean and a variance $\sigma^2 = 1, 9, 25$ and the sample sizes mentioned in the table above.

2- Generate contamination random errors

Contamination random errors are generated as follows :

1- Contaminating the random error with normal distribution by mean (0) and ($\sigma^2 = 36, 100$) by adding a contamination level (5%, 10%, 15%, 20%) to random error distribution.

2- Contaminating the random error by the distribution of t with degree of freedom (df = 1, 5) by adding a contamination level of 5%, 10%, 15% and 20%.

3-Calculate the dependent variable

The dependent variable was calculated by multiplying the matrix (X) generated in theoretical part by the default parameters vector then add the random error described in theoretical part as in equation (3-1).

4-**Repeat the experiment 1000 times** to stable results.

5-Calculate the RPE^W criterion

The RPE^W criterion is calculate and results were compare.

3-3 The results

Here we compare the PLS methods and the penalized robust methods in order to reach the best estimate for estimating the time series parameters in case of the outliers problem. After implementing the programs, the following results were obtained based on the penalized robust methods.

In this study, we select the optimal set of tuning parameters (a, γ, λ) by 10-foldcross validation. We compute the median (RPE^w) for each method with 1000 replications.

Case 1: sparse case

When generating data by model shown in formula (2-6) with the following default parameters:

Table (3-2) Default parameter values (sparse case)

y_{t-l}					$x_{1,t-l}$				$x_{2,t-l}$			
$\beta_{0,1}$	$\beta_{0,2}$	$\beta_{0,3}$	$\beta_{0,4}$	$\beta_{0,5}$	$\beta_{1,0}$	$\beta_{1,1}$	$\beta_{1,2}$	$\beta_{1,3}$	$\beta_{2,0}$	$\beta_{2,1}$	$\beta_{2,2}$	$\beta_{2,3}$
0.3	-0.2	0.1	0	0	0.9	0.7	0.5	0	1	-0.7	0.5	0

1- **Proposed M-lwlasso methods**

After implementation of the program in appendix [1], the following results were obtained, based on the variance of the random error of the non- contaminated data.

Table (3-3) RPE^{w} for methods when $\sigma = 1$ (sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	Lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		3.811	3.743	3.789	4.147	4.122	3.4395	3.353	3.404	3.841	3.813
	5%	N(0,6)	3.990	3.957	3.960	4.277	4.272	3.631	3.421	3.555	3.880	3.804
		N(0,10)	4.031	3.994	4.003	4.688	4.636	3.333	3.247	3.294	3.891	3.833
		t(1)	3.932	3.853	3.902	4.840	4.842	3.335	3.011	3.267	3.666	3.633
		t(5)	3.917	3.813	4.845	4.202	4.222	3.194	3.039	3.125	3.837	3.700
	10%	N(0,6)	4.131	4.004	4.016	4.94	4.877	3.396	3.001	3.048	3.493	3.426
		N(0,10)	6.280	6.082	6.184	6.929	6.801	3.309	3.269	3.297	3.583	3.470
		t(1)	4.105	4.088	4.133	4.127	4.156	3.406	3.382	3.456	3.418	3.426
		t(5)	4.042	4.007	4.014	4.962	4.872	3.373	3.052	3.150	3.495	3.430
	15%	N(0,٦)	5.601	5.065	5.172	5.856	5.760	3.300	3.153	3.274	3.532	3.401
		N(0,10)	10.122	9.997	10.082	11.107	11.115	3.295	3.078	4.1663	3.618	3.449
		t(1)	5.874	5.314	5.576	6.120	6.019	3.144	3.095	3.121	3.654	3.489
		t(5)	4.576	4.356	4.429	5.149	5.101	3.388	3.049	3.298	3.641	3.612
	20%	N(0,٦)	6.174	6.012	6.106	7.212	6.917	3.156	3.034	3.098	3.451	3.429
		N(0,10)	14.401	14.095	14.219	14.914	15.893	3.119	3.032	4.081	3.521	3.419
		t(1)	5.280	5.297	5.341	5.300	5.365	3.291	3.080	3.176	3.449	3.417
		t(5)	4.798	4.717	4.751	5.184	5.125	3.174	3.033	3.095	3.425	3.414
T=100	٠%		3.999	3.882	3.967	4.028	3.992	3.189	3.137	3.276	3.212	3.211
	5%	N(0,٦)	4.034	3.954	4.001	4.068	4.022	3.243	3.161	3.304	3.268	3.234
		N(0,10)	3.961	3.886	3.959	3.998	3.969	3.163	3.118	3.264	3.187	3.189
		t(1)	3.989	3.871	3.965	4.016	3.997	3.216	3.125	3.280	3.248	3.217
		t(5)	3.969	3.853	3.921	4.000	3.944	3.180	3.096	3.240	3.201	3.193
	10%	N(0,٦)	5.226	5.244	5.308	5.247	5.337	3.247	3.190	3.417	3.271	3.270
		N(0,10)	6.416	6.408	6.476	6.443	6.502	3.224	3.147	3.435	3.244	3.219
		t(1)	8.888	8.905	8.982	8.925	8.994	3.206	3.146	3.537	3.230	3.220
		t(5)	4.047	3.952	4.030	4.078	4.047	3.180	3.160	3.292	3.209	3.219
	15%	N(0,٦)	6.507	6.504	6.598	6.528	6.619	3.268	3.204	3.518	3.289	3.276

		N(0,10)	14.274	14.389	14.372	14.279	14.375	3.259	3.184	3.783	3.276	3.265
		t(1)	5.568	5.610	5.694	5.617	5.747	3.189	3.118	3.487	3.215	3.208
		t(5)	4.085	3.983	4.069	4.117	4.089	3.257	3.182	3.290	3.279	3.256
	20%	N(0,٦)	7.802	7.902	7.937	7.811	7.969	3.303	3.249	3.626	3.328	3.315
		N(0,10)	20.008	20.022	20.042	20.026	20.055	3.246	3.187	3.853	3.270	3.260
		t(1)	8.067	8.172	8.206	8.088	8.216	3.242	3.166	3.704	3.266	3.254
		t(5)	4.126	3.996	4.077	4.156	4.103	3.207	3.131	3.301	3.231	3.210
T=200	٠%		3.843	3.793	3.813	4.175	4.086	3.124	3.021	3.073	3.509	3.492
	٥%	N(0,٦)	5.144	5.075	5.109	6.908	6.603	3.130	3.029	3.095	3.542	3.496
		N(0,10)	8.241	8.023	8.126	10.266	10.077	3.126	3.010	3.103	3.530	3.469
		t(1)	7.379	7.023	7.244	8.304	8.265	3.128	3.032	3.114	3.511	3.507
		t(5)	4.961	4.377	4.576	5.076	5.022	3.139	3.040	3.117	3.655	3.469
	10%	N(0,٦)	5.750	5.736	5.748	5.877	5.810	3.151	3.031	3.125	3.578	3.510
		N(0,10)	11.544	11.440	11.456	11.800	11.593	3.092	3.011	3.038	3.633	3.571
		t(1)	5.674	5.403	5.627	5.777	5.723	3.120	3.036	3.104	3.526	3.504
		t(5)	3.968	3.834	3.897	3.997	3.924	3.130	3.009	3.063	3.584	3.538
	15%	N(0,٦)	7.087	6.997	7.065	7.113	7.086	3.168	3.082	3.135	3.545	3.518
		N(0,10)	16.860	15.913	15.998	17.896	17.009	3.096	3.037	3.073	3.611	3.609
		t(1)	8.094	7.322	8.320	9.410	9.341	3.121	3.033	3.044	3.506	3.328
		t(5)	4.035	3.911	3.979	4.064	4.003	3.148	3.053	3.089	3.718	3.558
	20%	N(0,٦)	8.334	8.240	8.288	8.553	8.464	3.226	3.158	3.185	3.588	3.579
		N(0,10)	21.412	21.184	21.253	21.737	21.687	3.137	3.035	3.082	3.543	3.511
		t(1)	12.647	12.248	12.323	12.766	12.651	3.128	3.035	3.099	3.535	3.511
		t(5)	4.733	3.962	4.197	5.040	5.006	3.158	3.074	3.102	3.545	3.436

Table (3-4) RPE^w for methods when $\sigma = 3$ (sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		3.777	3.602	3.673	4.098	4.011	3.252	3.172	3.232	3.753	3.642
	5%	N(0,6)	4.631	4.525	4.589	4.896	4.825	3.234	3.105	3.142	3.776	3.705
		N(0,10)	4.907	4.805	4.871	4.990	4.911	3.379	3.239	3.264	3.741	3.701
		t(1)	4.631	4.550	4.613	4.816	4.793	3.197	3.113	3.150	3.711	3.671
		t(5)	4.093	3.981	4.004	4.625	4.609	3.389	3.146	3.246	3.720	3.604
	10%	N(0,6)	4.795	4.594	4.667	5.094	5.087	3.344	3.148	3.288	3.724	3.638

		N(0,10)	5.103	5.002	5.089	5.318	5.285	3.322	3.132	3.319	3.813	3.805
		t(1)	4.899	4.613	4.666	4.972	4.914	3.336	3.154	3.245	3.812	3.733
		t(5)	4.604	4.222	4.412	4.880	4.729	3.328	3.138	3.244	3.780	3.710
	15%	N(0,٦)	4.872	4.718	4.780	5.194	5.122	3.322	3.136	3.173	3.741	3.651
		N(0,10)	5.498	5.346	5.386	5.886	5.772	3.298	3.145	3.186	3.831	3.650
		t(1)	4.925	4.756	4.780	5.029	5.019	3.354	3.197	3.325	3.781	3.751
		t(5)	4.783	4.649	4.729	4.906	4.847	3.312	3.211	3.288	3.744	3.723
	20%	N(0,٦)	4.976	4.866	4.904	5.284	5.137	3.333	3.109	3.257	3.727	3.679
		N(0,10)	5.672	5.467	5.596	5.945	5.827	3.264	3.129	3.175	3.690	3.658
		t(1)	5.399	5.145	5.178	5.894	5.829	3.357	3.116	3.249	3.767	3.651
		t(5)	4.975	4.890	4.814	5.004	4.993	3.270	3.124	3.206	3.776	3.744
T=100	٠%		3.921	3.848	3.929	3.949	3.952	3.167	3.091	3.207	3.198	3.154
	5%	N(0,٦)	3.996	3.924	4.014	4.031	4.041	3.221	3.196	3.332	3.255	3.284
		N(0,10)	4.014	3.873	3.948	4.043	3.971	3.242	3.134	3.276	3.266	3.237
		t(1)	3.998	3.916	3.980	4.024	3.998	3.215	3.180	3.315	3.230	3.254
		t(5)	3.982	3.937	4.014	4.011	4.029	3.218	3.149	3.284	3.243	3.231
	10%	N(0,٦)	4.037	3.944	4.036	4.071	4.063	3.221	3.105	3.279	3.245	3.218
		N(0,10)	4.549	4.514	4.593	4.581	4.619	3.279	3.208	3.350	3.305	3.275
		t(1)	4.084	4.010	4.082	4.109	4.102	3.136	3.053	3.248	3.159	3.127
		t(5)	3.930	3.851	3.920	3.962	3.937	3.137	3.086	3.220	3.162	3.147
	15%	N(0,٦)	4.175	4.028	4.104	4.205	4.137	3.268	3.162	3.314	3.292	3.266
		N(0,10)	5.090	5.050	5.134	5.113	5.154	3.309	3.240	3.470	3.324	3.324
		t(1)	4.393	4.353	4.421	4.422	4.449	3.153	3.091	3.325	3.183	3.169
		t(5)	3.899	3.808	3.876	3.929	3.900	3.117	3.035	3.197	3.142	3.127
	20%	N(0,٦)	4.254	4.176	4.261	4.285	4.284	3.285	3.211	3.372	3.317	3.318
		N(0,10)	5.639	5.577	5.671	5.670	5.700	3.358	3.352	3.541	3.373	3.425
		t(1)	4.612	4.580	4.647	4.635	4.675	3.122	3.045	3.323	3.155	3.125
		t(5)	3.837	3.724	3.807	3.873	3.833	3.050	2.957	3.100	3.075	3.053
T=200	٠%		3.884	3.730	3.802	3.913	3.901	3.305	2.173	3.238	3.723	3.704
	٥%	N(0,٦)	3.954	3.819	3.881	4.129	4.078	3.219	3.152	3.185	3.713	3.684
		N(0,10)	4.210	4.100	4.166	4.626	4.601	3.358	3.149	3.205	3.704	3.697
		t(1)	4.095	4.008	4.013	4.499	4.336	3.307	2.177	3.211	3.864	3.808
		t(5)	3.934	3.781	3.851	3.971	3.873	3.244	3.173	3.215	3.534	3.477
	10%	N(0,٦)	4.068	3.956	4.034	4.377	4.338	3.314	3.102	3.238	3.753	3.679
		N(0,10)	4.432	4.328	4.381	4.767	4.713	3.316	3.155	3.265	3.844	3.712
		t(1)	4.269	4.198	4.204	4.893	4.802	3.304	2.203	3.295	3.741	3.709

		t(5)	4.183	4.009	4.077	4.697	4.624	3.342	2.252	3.281	3.695	3.600
	15%	N(0,٦)	4.373	4.233	4.356	4.599	4.523	3.211	3.196	3.201	3.755	3.663
		N(0,10)	5.263	5.163	5.241	5.895	5.614	3.177	3.085	3.131	3.756	3.715
		t(1)	4.794	4.616	4.653	4.952	4.911	3.396	2.138	3.161	3.656	3.601
		t(5)	4.682	4.527	4.594	4.858	4.781	3.300	2.107	3.203	3.731	2.917
	20%	N(0,٦)	5.042	4.963	5.004	5.695	5.526	3.228	3.147	3.203	3.696	3.540
		N(0,10)	5.811	5.710	5.808	5.984	5.928	3.258	3.162	3.217	3.669	3.582
		t(1)	4.980	4.877	4.917	5.043	5.013	3.266	2.116	3.163	3.654	2.607
		t(5)	4.921	4.765	3.867	5.009	4.943	3.278	3.108	2.149	2.712	2.683

Table (3-5) RPE^{w} for methods when $\sigma = 5$ (sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		3.962	3.863	3.918	4.198	4.113	3.402	3.283	3.374	3.782	3.719
	5%	N(0,6)	4.063	4.001	4.007	4.309	4.390	3.364	3.247	3.308	3.895	3.736
		N(0,10)	4.231	4.068	4.132	4.560	4.512	3.340	3.207	3.311	3.800	3.712
		t(1)	4.111	4.044	4.086	4.396	4.319	3.392	3.223	3.252	3.911	3.750
		t(5)	4.015	3.949	4.005	4.301	4.105	3.317	3.253	3.305	3.642	3.566
	10%	N(0,6)	4.486	4.263	4.299	4.713	4.710	3.307	3.203	3.241	3.852	3.764
		N(0,10)	4.648	4.516	4.581	4.943	4.827	3.349	3.210	3.237	3.766	3.717
		t(1)	4.259	4.156	4.186	4.910	4.831	3.338	3.218	3.273	3.827	3.803
		t(5)	4.169	3.068	4.081	4.311	4.304	3.315	3.203	3.275	3.801	3.798
	15%	N(0,٦)	4.745	4.604	4.700	4.874	4.779	3.310	3.280	3.294	3.812	3.773
		N(0,10)	4.944	4.879	4.908	5.012	5.003	3.474	3.338	3.416	3.785	3.738
		t(1)	4.917	4.810	4.865	4.987	4.888	3.322	3.256	3.202	3.809	3.733
		t(5)	4.568	3.481	4.500	4.874	4.804	3.328	3.261	3.297	3.759	3.723
	20%	N(0,٦)	4.923	4.854	4.813	4.977	4.892	3.373	3.113	3.281	3.839	3.822
		N(0,10)	5.174	5.026	5.097	5.424	5.326	3.222	3.126	3.196	3.862	3.748
		t(1)	5.024	4.984	5.006	5.339	5.274	3.371	3.118	3.231	3.908	3.799
		t(5)	3.995	3.899	3.986	4.027	4.003	3.301	3.211	3.293	3.833	3.825
T=100	٠%		3.961	3.913	3.983	3.987	4.005	3.189	3.167	3.294	3.215	3.222
	5%	N(0,٦)	3.941	3.857	3.934	3.967	3.959	3.166	3.083	3.218	3.188	3.182
		N(0,10)	4.023	3.940	4.013	4.046	4.039	3.236	3.184	3.350	3.262	3.257
		t(1)	3.949	3.847	3.910	3.981	3.934	3.218	3.111	3.229	3.240	3.201

		t(5)	3.995	3.921	3.973	4.019	3.985	3.227	3.143	3.279	3.252	3.231
	10%	N(0,٦)	3.966	3.881	3.943	3.994	3.971	3.193	3.119	3.292	3.218	3.199
		N(0,10)	4.199	4.107	4.181	4.229	4.209	3.250	3.209	3.364	3.277	3.296
		t(1)	4.091	3.986	4.059	4.114	4.079	3.156	3.131	3.284	3.179	3.194
		t(5)	3.948	3.812	3.876	3.972	3.903	3.157	3.051	3.195	3.181	3.139
	15%	N(0,٦)	3.984	3.910	3.995	4.021	4.025	3.189	3.138	3.278	3.213	3.208
		N(0,10)	4.323	4.235	4.301	4.348	4.323	3.283	3.205	3.366	3.308	3.287
		t(1)	4.173	4.098	4.164	4.197	4.181	3.098	3.030	3.211	3.118	3.102
		t(5)	3.893	3.808	3.861	3.923	3.882	3.080	2.981	3.126	3.096	3.056
	20%	N(0,٦)	4.019	3.944	4.027	4.045	4.036	3.230	3.147	3.297	3.251	3.237
		N(0,10)	4.507	4.457	4.533	4.527	4.551	3.392	3.341	3.472	3.412	3.406
		t(1)	4.186	4.118	4.181	4.212	4.200	3.062	2.996	3.179	3.082	3.069
		t(5)	3.794	3.715	3.777	3.831	3.804	2.988	2.901	3.046	3.008	2.977
T=200	٠%		3.884	3.739	3.800	3.921	3.828	3.264	2.148	3.197	3.811	3.761
	٥%	N(0,٦)	3.978	3.777	3.834	3.996	3.985	3.217	3.122	3.160	3.806	3.726
		N(0,10)	4.010	3.886	3.959	4.405	4.340	3.315	3.128	3.186	3.803	3.756
		t(1)	3.985	3.854	3.921	4.019	4.009	3.284	3.200	3.251	3.816	3.766
		t(5)	3.880	3.707	3.954	3.918	3.909	3.316	2.181	3.217	3.894	3.838
	10%	N(0,٦)	4.197	4.079	4.109	4.489	3.408	3.289	2.146	3.205	3.861	3.773
		N(0,10)	4.389	4.209	4.315	4.558	4.452	3.225	3.108	3.187	3.853	3.635
		t(1)	4.108	4.045	4.097	4.480	4.652	3.277	2.132	3.166	3.710	3.702
		t(5)	3.928	3.876	3.889	3.911	3.898	3.326	2.155	3.190	3.796	3.775
	15%	N(0,٦)	4.293	4.179	4.219	4.489	4.408	3.289	2.199	3.205	3.821	3.773
		N(0,10)	4.489	4.411	4.455	4.558	4.452	3.225	3.095	3.187	3.853	3.633
		t(1)	4.419	4.372	4.409	4.873	4.652	3.307	2.132	3.232	3.810	3.757
		t(5)	4.187	4.076	4.139	3.612	3.583	3.261	2.155	3.201	3.765	3.707
	20%	N(0,٦)	4.467	4.375	4.416	4.977	4.897	3.216	3.151	3.1885	3.883	3.805
		N(0,10)	4.869	4.719	4.736	5.104	5.035	3.248	3.171	3.226	3.733	3.712
		t(1)	4.650	4.506	4.582	4.822	4.800	3.296	2.168	3.206	3.990	3.933
		t(5)	4.482	4.336	4.416	4.780	4.675	3.387	3.181	3.192	3.900	3.821

Tables(3-3), (3-4), and (3-5) show that the results of the ordinary methods are very close to our proposed robust methods when σ = 1, 3, 5. However, when contaminating the data at different rates and distributions (normal and t), our proposed methods are more stable than the ordinary methods, based on the values of RPE^{w} across deferent error distributions.

2-Proposed MM-lwlasso methods

After implementation of the program in appendix [2], the following results were obtained, based on the random error of the non- contaminated data

Table (3-6) RPE^{w} for methods when $\sigma = 1$ (sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
T=50	0%		4.075	3.959	4.036	4.864	4.576	3.589	3.530	3.533	3.961	3.957
	5%	N(0,6)	5.894	5.456	5.541	5.862	5.854	3.639	3.561	3.592	3.864	3.775
		N(0,10)	7.611	6.926	7.369	8.628	8.538	3.577	3.528	3.539	3.834	3.742
		t(1)	6.454	6.379	6.406	7.485	7.355	3.571	3.455	3.519	3.845	3.665
		t(5)	4.993	4.433	4.858	5.132	5.108	3.574	3.249	3.459	3.908	3.828
	10%	N(0,6)	8.119	8.005	8.019	8.961	8.956	3.499	3.177	3.181	3.841	3.833
		N(0,10)	14.520	14.460	14.518	14.951	14.909	3.401	3.377	3.392	3.830	3.828
		t(1)	7.445	7.160	7.258	8.973	8.917	3.488	3.401	3.450	3.756	3.671
		t(5)	5.684	5.148	5.304	6.236	6.126	3.445	3.369	3.401	3.866	3.721
	15%	N(0,٦)	9.487	9.025	9.264	9.924	9.831	3.460	3.396	3.437	3.854	4.624
		N(0,10)	18.313	18.070	18.160	18.855	18.806	3.461	3.353	3.413	3.799	3.617
		t(1)	10.034	9.971	10.002	10.748	10.697	3.365	3.124	3.294	3.754	3.719
		t(5)	6.242	6.176	6.233	6.825	6.620	3.601	3.549	3.564	3.962	3.919
	20%	N(0,٦)	10.794	10.612	10.710	11.307	11.160	3.543	3.264	3.390	3.712	3.624
		N(0,10)	20.703	20.600	20.673	21.731	21.484	3.474	3.357	3.376	3.772	3.684
		t(1)	14.130	14.029	14.078	14.737	14.687	3.367	3.267	3.312	3.706	3.694
		t(5)	10.461	10.235	10.266	11.005	10.838	3.429	3.3913	3.243	3.791	3.766
T=100	٠%		3.959	3.898	3.938	4.117	4.001	3.417	3.309	3.359	3.704	3.684
	5%	N(0,٦)	5.383	5.114	5.224	5.970	5.768	3.509	3.440	3.474	3.853	3.737
		N(0,10)	8.299	8.187	8.233	8.860	8.610	3.464	3.148	3.194	3.792	3.728
		t(1)	4.499	4.347	4.408	4.883	4.833	3.410	3.181	3.347	3.952	3.611
		t(5)	4.056	3.977	4.033	4.597	4.361	3.427	3.243	3.385	3.959	3.834
	10%	N(0,٦)	7.728	7.413	7.515	7.987	7.801	3.441	3.214	3.384	3.764	3.641

		N(0,10)	14.26	13.603	14.096	14.839	14.664	3.611	3.523	3.587	3.822	3.662
		t(1)	5.622	5.256	5.477	5.986	5.775	3.548	3.462	3.480	3.809	3.518
		t(5)	4.201	4.001	4.022	4.513	4.338	3.431	3.307	3.415	3.838	3.755
	15%	N(0,٦)	9.589	9.082	9.169	9.847	9.716	3.356	3.255	3.275	3.754	3.701
		N(0,10)	19.387	19.029	19.267	19.902	19.676	3.359	3.143	3.198	3.742	3.721
		t(1)	7.504	7.476	7.496	7.753	7.673	3.378	3.185	3.211	3.708	3.624
		t(5)	4.742	3.552	4.637	4.851	4.741	3.502	3.305	3.447	3.953	3.834
	20%	N(0,٦)	11.662	11.111	11.476	11.904	11.859	3.451	3.269	3.280	3.841	3.807
		N(0,10)	24.330	24.034	24.003	24.868	24.769	3.348	3.251	3.315	3.851	4.801
		t(1)	9.418	8.899	9.260	10.045	9.709	3.487	3.204	3.268	3.874	3.819
		t(5)	4.902	4.874	4.900	5.107	5.007	3.507	3.418	3.421	3.935	3.708
T=200	٠%		3.903	3.769	3.830	3.973	3.857	3.415	3.279	3.378	3.782	3.744
	٥%	N(0,٦)	5.627	5.4753	5.570	5.902	5.799	3.441	3.303	3.355	3.753	3.641
		N(0,10)	8.419	8.029	8.196	8.879	8.855	3.443	3.309	3.385	3.933	3.906
		t(1)	4.189	4.033	4.069	4.848	4.808	3.416	3.334	3.359	3.951	3.901
		t(5)	3.953	3.806	3.884	3.980	3.901	3.466	3.323	3.421	3.823	3.745
	10%	N(0,٦)	7.493	7.187	7.243	7.533	7.524	3.543	3.408	3.499	3.830	3.809
		N(0,10)	14.306	14.232	14.279	14.803	14.740	3.486	3.374	3.437	4.743	4.592
		t(1)	6.748	6.466	6.592	6.906	6.741	3.417	3.187	3.397	4.791	4.709
		t(5)	4.213	4.008	4.196	4.681	4.502	3.436	3.231	3.361	3.715	3.661
	15%	N(0,٦)	9.478	9.317	9.454	9.843	9.788	3.307	3.212	3.294	3.964	3.905
		N(0,10)	19.294	19.167	19.222	19.741	19.713	3.443	3.155	4.304	3.636	3.633
		t(1)	10.067	9.808	10.024	10.791	10.581	3.443	3.185	3.249	3.852	3.798
		t(5)	4.937	3.865	3.902	4.054	3.965	3.470	3.313	3.407	3.961	3.869
	20%	N(0,٦)	11.375	11.124	11.240	11.898	11.749	3.389	3.172	3.342	3.784	3.713
		N(0,10)	24.411	24.183	24.233	24.752	24.678	3.275	3.168	3.203	3.776	3.745
		t(1)	16.462	16.276	16.378	16.914	16.849	3.4335	3.014	3.034	3.693	3.521
		t(5)	5.0757	4.942	5.005	5.464	5.435	3.402	3.034	3.098	3.885	3.745

Table (3-7) RPE^{w} for methods when $\sigma = 3$ (sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
T=50	0%		3.981	3.842	4.875	4.160	4.124	3.547	3.317	3.361	3.883	3.813
	5%	N(0,6)	4.236	4.126	4.214	4.818	4.641	3.608	3.308	3.495	3.851	3.720

		N(0,10)	4.585	4.456	4.539	4.885	4.647	3.270	3.163	3.221	3.822	3.716
		t(1)	4.271	4.109	4.161	4.786	4.431	3.575	3.205	3.360	3.710	3.684
		t(5)	4.077	3.905	4.001	4.271	4.146	3.507	3.225	3.138	3.654	3.600
	10%	N(0,6)	4.501	4.444	4.453	4.697	4.605	3.311	3.185	3.285	3.807	3.7277
		N(0,10)	5.266	5.011	5.133	5.738	5.638	3.258	3.107	3.149	3.792	3.585
		t(1)	5.169	4.664	5.083	5.447	5.147	3.522	3.438	3.474	3.879	3.705
		t(5)	4.306	4.215	4.254	4.807	4.737	3.530	3.211	3.349	3.697	3.582
	15%	N(0,٦)	5.129	5.016	5.107	5.621	5.607	3.552	3.486	3.544	3.893	3.804
		N(0,10)	5.898	5.830	5.876	5.998	5.909	3.501	3.439	3.497	3.789	4.704
		t(1)	5.480	5.297	5.336	5.603	5.575	3.562	3.404	3.523	3.811	3.773
		t(5)	4.850	4.435	4.797	4.979	4.888	3.424	3.175	3.257	3.845	3.655
	20%	N(0,٦)	5.766	5.596	5.723	5.955	5.821	3.4459	3.325	3.400	3.880	3.818
		N(0,10)	6.515	6.338	6.377	6.867	6.591	3.242	3.157	3.165	3.7532	3.703
		t(1)	5.756	5.682	5.738	5.964	5.876	3.514	3.370	3.259	3.882	3.819
		t(5)	5.081	5.002	5.019	5.862	5.601	3.553	3.216	3.308	3.834	3.829
T=100	٠%		3.966	3.836	3.915	3.994	3.937	3.493	3.364	3.454	3.722	3.717
	5%	N(0,٦)	4.148	4.039	4.093	4.173	4.122	3.558	3.403	3.456	3.867	3.629
		N(0,10)	4.539	4.421	4.522	4.569	4.543	3.530	3.387	3.405	3.874	3.777
		t(1)	4.548	4.470	4.511	4.755	4.648	3.523	3.445	3.475	3.754	3.625
		t(5)	4.119	3.897	4.043	4.601	4.514	3.205	3.023	3.080	3.785	3.695
	10%	N(0,٦)	4.337	4.260	4.339	4.372	4.358	3.064	3.053	3.061	3.671	3.631
		N(0,10)	5.249	5.021	5.100	5.611	5.523	3.259	3.054	3.145	3.699	3.625
		t(1)	4.834	4.800	4.806	4.921	4.884	3.465	3.306	3.385	3.808	3.651
		t(5)	4.656	4.110	4.186	4.894	4.806	3.474	3.202	3.253	3.833	3.803
	15%	N(0,٦)	4.958	4.710	4.814	5.006	5.001	3.442	3.258	3.337	3.706	3.682
		N(0,10)	5.789	5.657	5.728	5.976	5.821	3.445	3.169	3.286	3.790	3.765
		t(1)	4.947	4.892	4.874	5.064	5.002	3.373	3.180	3.306	3.769	3.713
		t(5)	4.810	4.727	4.798	4.838	4.814	3.265	3.117	3.145	3.946	3.521
	20%	N(0,٦)	5.207	5.124	5.166	5.453	5.353	3.375	3.264	3.323	3.762	3.727
		N(0,10)	6.154	6.015	6.093	6.594	6.517	3.472	3.300	3.326	3.871	3.709
		t(1)	5.393	5.079	5.106	5.790	5.700	3.346	3.213	3.242	3.944	3.518
		t(5)	5.026	4.976	5.008	5.878	5.824	3.224	3.135	3.207	3.985	3.974
T=200	٠%		3.939	3.841	3.909	3.991	3.963	3.471	3.369	3.428	3.844	3.815
	٥%	N(0,٦)	4.112	3.977	4.047	4.382	4.292	3.441	3.385	3.406	3.757	3.727
		N(0,10)	4.507	4.409	4.484	4.537	4.507	3.506	3.389	3.423	3.956	3.923
		t(1)	4.146	4.050	4.111	4.417	4.356	3.431	3.297	3.342	3.977	3.862

		t(5)	4.088	4.019	4.088	4.224	4.212	3.371	3.239	3.339	3.951	3.809
	10%	N(0,٦)	4.231	4.136	4.199	4.610	4.582	3.451	3.397	3.417	3.711	3.646
		N(0,10)	5.004	4.905	4.977	5.309	5.234	3.562	3.481	3.524	3.888	3.855
		t(1)	4.338	4.233	4.300	4.661	4.632	3.382	3.236	3.320	3.891	3.833
		t(5)	4.284	4.155	4.177	4.873	4.799	3.367	3.178	3.208	3.766	3.721
	15%	N(0,٦)	4.403	4.293	4.376	4.843	4.701	3.564	3.382	3.463	3.945	3.716
		N(0,10)	5.467	5.404	5.430	5.785	5.643	3.559	3.407	3.517	3.768	3.732
		t(1)	4.964	4.725	4.841	5.287	5.008	3.313	3.197	3.265	3.766	3.716
		t(5)	4.843	4.707	4.806	4.987	3.922	3.272	3.136	3.257	3.980	3.878
	20%	N(0,٦)	4.862	4.684	4.723	5.590	5.545	3.505	3.184	3.226	3.862	3.835
		N(0,10)	6.100	6.004	6.052	6.305	6.170	3.291	3.161	3.247	3.721	3.706
		t(1)	5.216	5.094	5.185	5.543	5.308	3.302	3.177	3.230	3.932	3.922
		t(5)	5.148	5.061	5.106	5.788	5.708	3.410	3.229	3.137	3.843	3.824

Table (3-8) RPE^{w} for methods when $\sigma = 5$ (sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
T=50	0%		4.071	4.006	4.017	4.189	4.148	3.589	3.194	3.451	3.924	3.912
	5%	N(0,6)	4.183	4.113	4.169	4.208	4.199	3.537	3.197	3.425	3.832	3.802
		N(0,10)	4.249	4.216	4.238	4.433	4.382	3.512	3.333	3.442	3.885	3.700
		t(1)	4.132	4.083	4.106	4.195	4.181	3.552	3.418	3.406	3.826	3.788
		t(5)	4.077	4.016	4.053	4.194	4.187	3.565	3.271	3.404	3.727	3.615
	10%	N(0,6)	4.388	4.171	4.210	4.420	4.401	3.633	3.502	3.616	3.957	3.874
		N(0,10)	4.483	4.411	4.419	4.513	4.503	3.435	3.215	3.343	3.856	3.810
		t(1)	4.223	4.197	4.215	4.568	4.351	3.509	3.489	3.406	3.773	3.705
		t(5)	4.152	4.103	4.122	4.441	4.425	3.407	3.246	3.365	3.825	3.823
	15%	N(0,6)	4.664	4.551	4.588	4.876	4.718	3.498	3.400	3.406	3.762	3.748
		N(0,10)	4.715	4.642	4.700	4.741	4.725	3.428	3.257	3.324	3.837	3.774
		t(1)	4.489	4.423	4.463	4.636	4.608	3.408	3.334	3.376	3.842	3.813
		t(5)	4.318	4.287	4.300	4.419	4.839	3.508	3.387	3.425	3.613	3.546
	20%	N(0,6)	4.991	4.677	4.732	5.031	5.008	3.479	3.347	3.305	3.744	3.723
		N(0,10)	5.073	4.943	5.008	5.683	5.570	3.468	3.394	3.405	3.898	3.807
		t(1)	4.833	4.671	4.732	4.849	4.651	3.378	3.195	3.241	3.775	3.657
		t(5)	4.664	4.608	4.647	4.940	4.741	3.612	2.203	3.368	3.985	3.933

T=100	٠%		3.923	3.632	4.813	4.076	4.014	3.506	3.194	3.451	3.924	3.791
	5%	N(0,6)	4.030	3.938	4.015	4.306	4.440	3.540	3.255	3.332	3.564	3.501
		N(0,10)	4.298	4.197	4.232	4.323	4.311	3.372	3.150	3.164	3.728	3.598
		t(1)	4.178	4.098	4.142	4.233	4.209	3.412	3.139	3.161	3.987	3.841
		t(5)	4.085	4.021	4.068	4.129	4.109	3.381	3.123	3.333	3.984	3.893
	10%	N(0,6)	4.263	4.154	4.206	4.889	4.863	3.467	3.246	3.415	3.975	3.871
		N(0,10)	4.665	4.584	4.626	4.924	4.797	3.522	3.314	3.416	3.726	3.632
		t(1)	4.463	4.347	4.432	4.816	4.661	3.300	3.203	3.220	3.732	3.719
		t(5)	4.277	4.235	4.252	4.811	4.802	3.291	3.121	3.188	3.818	3.795
	15%	N(0,6)	4.492	4.324	4.408	4.550	4.516	3.566	3.133	3.275	3.918	3.790
		N(0,10)	4.988	4.931	4.953	5.185	5.024	3.344	3.178	3.218	3.864	3.805
		t(1)	4.775	4.609	4.652	4.754	4.603	3.561	3.263	3.326	3.840	3.613
		t(5)	4.693	4.573	4.650	4.717	4.667	3.607	3.207	3.255	3.732	3.559
	20%	N(0,6)	4.814	4.691	4.713	4.966	4.915	3.501	3.277	3.472	3.656	3.612
		N(0,10)	5.370	5.272	5.327	5.819	5.718	3.423	3.186	3.256	3.841	3.809
		t(1)	4.995	4.888	5.931	5.997	5.992	3.475	2.127	3.229	3.979	2.716
		t(5)	4.895	4.743	4.791	4.875	4.832	3.179	3.068	3.112	3.811	3.744
T=200	٠%		3.792	3.632	3.781	4.197	4.145	3.506	3.351	3.451	3.924	3.791
	5%	N(0,6)	4.082	3.908	4.043	4.393	4.328	3.523	3.333	3.418	3.804	3.695
		N(0,10)	4.467	4.339	4.391	4.578	4.407	3.480	3.308	3.413	3.759	3.699
		t(1)	4.245	4.116	4.194	4.275	4.217	3.532	3.413	3.525	3.856	3.782
		t(5)	4.117	4.003	4.085	4.255	4.201	3.541	3.183	3.378	3.787	3.719
	10%	N(0,6)	4.3464	4.007	4.195	4.668	4.501	3.485	3.396	3.417	3.854	3.742
		N(0,10)	4.585	4.463	4.530	4.612	4.548	3.525	3.124	3.369	3.761	3.685
		t(1)	4.420	4.341	4.411	4.514	4.496	3.554	3.315	3.412	3.908	3.824
		t(5)	4.371	4.114	4.267	4.748	4.627	3.394	3.112	3.176	3.855	3.718
	15%	N(0,6)	4.616	3.126	4.408	4.906	4.764	3.538	3.324	3.339	3.856	3.788
		N(0,10)	4.849	4.796	4.822	4.910	4.878	3.509	3.215	3.328	3.896	3.770
		t(1)	4.858	4.679	4.786	4.901	4.894	3.469	3.394	3.402	3.923	3.683
		t(5)	4.647	4.474	4.532	4.8705	4.868	3.522	3.172	3.287	3.909	3.818
	20%	N(0,6)	4.913	4.750	4.628	4.963	4.906	3.512	3.462	3.506	3.860	3.802
		N(0,10)	5.608	5.231	5.486	5.722	5.718	3.364	3.205	3.282	3.929	3.842
		t(1)	5.201	5.105	5.181	5.335	5.207	3.236	3.160	3.193	3.935	3.894
		t(5)	4.934	4.805	4.873	4.926	4.460	3.331	3.187	3.207	3.754	3.621

The RPE^w values in Table (3-6), (3-7) and (3-8) show that the behavior of the methods was similar to what was explained in the previous tables in the case of clean data, as well as when contaminating the data with different contamination rates and sizes of different samples. The best results were in the MM-lwlasso method with w^{R2} as the amount of RPE^w is the least followed by the proposed method MM-lwlasso proposed with the w^{R3} and w^{R1}.

The second case: very sparse case

When generating data by model shown in formula (2-6) with the following default parameters:

Table (3-9) Default parameter values (very sparse case)

y_{t-l}					$x_{1,t-l}$				$x_{2,t-l}$			
$\beta_{0,1}$	$\beta_{0,2}$	$\beta_{0,3}$	$\beta_{0,4}$	$\beta_{0,5}$	$\beta_{1,0}$	$\beta_{1,1}$	$\beta_{1,2}$	$\beta_{1,3}$	$\beta_{2,0}$	$\beta_{2,1}$	$\beta_{2,2}$	$\beta_{2,3}$
0.3	0	0	0	0	0.9	0	0	0	1	0	0	0

1-Proposed M-lwlasso methods

After implementation of the program in appendix[1], the following results were obtained, based on the random error limit of the non- contaminated data

Table (3-10) RPE^w for methods when $\sigma = 1$ (very sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		0.940	0.924	0.932	1.966	1.933	0.496	0.339	0.410	1.134	1.051
	5%	N(0,6)	1.937	1.710	1.771	2.699	2.780	0.199	0.110	0.189	1.688	1.617
		N(0,10)	3.997	3.265	3.566	4.141	4.060	0.169	0.070	0.110	1.689	1.522
		t(1)	3.859	3.808	3.842	3.985	3.938	0.078	0.055	0.072	1.553	1.547

		t(5)	2.992	2.016	2.085	3.994	3.010	0.300	0.013	0.199	1.499	1.491
	10%	N(0,6)	3.675	3.625	3.658	4.842	4.811	0.159	0.065	0.087	1.663	1.633
		N(0,10)	4.419	4.000	4.375	5.419	5.357	0.714	0.183	0.212	1.780	1.737
		t(1)	3.996	3.915	3.981	4.861	4.666	0.153	0.133	0.147	1.535	1.524
		t(5)	3.333	3.002	3.110	3.936	3.913	0.399	0.128	0.305	1.598	1.556
	15%	N(0,6)	4.316	4.037	4.306	5.299	5.255	0.209	0.092	0.175	1.930	1.912
		N(0,10)	5.183	5.030	5.091	6.175	6.302	0.859	0.186	0.467	1.961	1.912
		t(1)	4.622	4.399	4.412	5.630	5.116	0.285	0.016	0.092	1.586	1.523
		t(5)	3.570	3.057	3.155	4.058	4.017	0.236	0.055	0.155	1.630	1.546
	20%	N(0,6)	5.906	5.033	5.025	7.940	7.021	0.302	0.132	0.167	1.298	1.310
		N(0,10)	7.228	7.269	7.194	8.179	8.163	0.388	0.394	0.592	1.391	1.411
		t(1)	6.711	6.460	6.426	7.340	7.398	0.611	0.096	0.223	1.609	1.631
		t(5)	3.882	3.643	3.850	4.038	4.053	0.523	0.104	0.286	1.522	1.436
T=100	0%		0.683	0.295	0.557	0.865	0.819	0.404	0.264	0.295	0.949	0.806
	5%	N(0,6)	1.322	1.224	1.275	1.960	1.900	0.567	0.265	0.366	1.048	0.972
		N(0,10)	3.478	3.221	3.354	3.964	3.935	0.300	0.263	0.286	0.743	0.607
		t(1)	1.486	1.217	1.356	1.766	1.679	0.399	0.263	0.297	0.793	0.777
		t(5)	0.854	0.521	0.753	0.933	0.878	0.399	0.224	0.265	0.644	0.609
	10%	N(0,6)	1.718	1.672	1.688	1.911	1.875	0.298	0.201	0.269	1.047	0.977
		N(0,10)	4.869	3.881	4.757	4.964	4.900	0.336	0.297	0.301	1.383	1.139
		t(1)	2.587	2.199	2.448	2.784	2.757	0.314	0.236	0.279	1.360	1.117
		t(5)	1.114	1.028	1.043	1.733	1.684	0.305	0.266	0.278	1.349	1.007
	15%	N(0,6)	2.827	2.719	2.779	3.022	2.992	0.332	0.123	0.196	1.048	0.973
		N(0,10)	4.952	4.901	4.916	5.260	5.011	0.375	0.133	0.238	1.122	1.069
		t(1)	3.707	3.574	3.669	4.236	4.004	0.330	0.028	0.291	1.377	1.027
		t(5)	2.000	1.932	1.968	2.980	2.783	0.308	0.069	0.271	1.350	1.011
	20%	N(0,6)	3.887	3.899	3.835	3.784	4.000	0.365	0.360	0.361	1.047	0.968
		N(0,10)	6.983	6.979	6.902	7.899	7.021	0.427	0.374	0.379	0.475	0.516
		t(1)	5.825	5.849	5.698	5.753	7.004	0.341	0.300	0.303	0.387	0.438
		t(5)	2.804	2.632	2.778	3.689	3.282	0.313	0.074	0.275	1.355	1.414
T=200	0%		0.989	0.950	0.977	1.959	1.767	0.226	0.028	0.125	1.525	1.220
	5%	N(0,6)	1.634	1.432	1.517	2.634	2.192	0.205	0.084	0.128	1.523	1.214
		N(0,10)	3.694	3.102	3.431	5.685	5.601	0.520	0.136	0.216	1.222	1.018
		t(1)	2.703	2.150	2.343	3.706	3.325	0.507	0.085	0.294	1.507	1.304
		t(5)	1.912	1.192	1.203	2.912	2.145	0.467	0.115	0.345	1.567	1.400
	10%	N(0,6)	2.413	2.289	2.318	3.418	3.416	0.575	0.157	0.159	1.575	1.510

		N(0,10)	4.896	4.478	4.591	5.888	5.679	0.621	0.212	0.508	1.621	1.221
		t(1)	3.560	3.204	3.448	5.566	5.423	0.522	0.024	0.186	1.522	1.262
		t(5)	2.902	2.226	2.562	2.966	2.948	0.523	0.081	0.159	1.835	1.482
	15%	N(0,6)	3.712	3.258	3.644	4.274	4.078	0.389	0.046	0.220	1.677	1.457
		N(0,10)	5.348	5.102	5.212	7.410	7.215	0.409	0.068	0.376	1.708	1.115
		t(1)	4.438	4.309	4.399	5.452	5.223	0.519	0.264	0.356	1.517	1.272
		t(5)	3.194	2.659	2.915	3.994	3.889	0.492	0.186	0.249	1.491	1.217
	20%	N(0,6)	4.363	4.015	4.282	5.243	5.177	0.467	0.140	0.211	1.844	1.384
		N(0,10)	6.494	6.244	6.353	7.429	7.253	0.817	0.168	0.696	1.820	1.223
		t(1)	5.302	5.002	5.170	6.302	6.058	0.551	0.148	0.320	1.550	1.164
		t(5)	3.991	3.235	3.329	4.992	4.324	0.919	0.046	0.487	1.906	1.486

Table (3-11) RPE^w for methods when $\sigma = 3$ (very sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		0.554	0.452	0.500	0.902	0.737	0.143	0.015	0.108	0.310	0.318
	5%	N(0,6)	0.941	0.825	0.896	1.126	1.027	0.122	0.013	0.106	0.374	0.317
		N(0,10)	1.235	1.011	1.051	1.985	1.889	0.120	0.023	0.101	0.463	0.373
		t(1)	0.937	0.908	0.911	1.088	1.080	0.186	0.056	0.117	0.416	0.380
		t(5)	0.788	0.606	0.610	0.998	0.813	0.135	0.037	0.098	0.500	0.378
	10%	N(0,6)	1.164	1.039	1.125	1.907	1.757	0.105	0.024	0.085	0.445	0.325
		N(0,10)	1.972	1.807	1.873	2.153	2.011	0.122	0.036	0.071	0.433	0.367
		t(1)	1.013	0.982	0.955	1.390	1.303	0.140	0.037	0.114	0.403	0.384
		t(5)	0.811	0.675	0.687	1.169	1.188	0.126	0.067	0.103	0.545	0.379
	15%	N(0,6)	1.993	1.845	1.904	2.012	1.994	0.197	0.078	0.161	0.446	0.400
		N(0,10)	2.190	1.976	1.999	2.419	2.158	0.129	0.059	0.104	0.586	0.391
		t(1)	1.667	1.091	1.169	1.656	1.501	0.300	0.090	0.143	0.430	0.398
		t(5)	0.965	0.844	0.920	1.645	1.425	0.172	0.072	0.127	0.427	0.345
	20%	N(0,6)	2.011	1.964	2.038	2.552	2.387	0.218	0.125	0.200	0.428	0.366
		N(0,10)	2.874	2.104	2.183	2.909	2.605	0.250	0.084	0.143	0.460	0.356
		t(1)	1.835	1.661	1.691	1.936	1.889	0.246	0.047	0.190	0.412	0.371
		t(5)	1.260	1.021	1.128	1.853	1.797	0.105	0.043	0.098	0.575	0.474
T=100	0%		0.483	0.318	0.356	0.365	0.976	0.302	0.264	0.266	0.344	0.407

	5%	N(0,6)	0.646	0.679	0.600	0.852	0.979	0.308	0.305	0.307	1.048	0.967
		N(0,10)	1.584	1.718	1.754	1.764	1.978	0.298	0.262	0.265	0.344	0.405
		t(1)	0.682	0.620	0.653	0.636	0.979	0.300	0.265	0.268	0.344	0.408
		t(5)	0.480	0.617	0. 540	0. 520	0.981	0.303	0.266	0.269	0.344	0.406
	10%	N(0,6)	1.514	1.648	1.515	1.396	1.983	0.277	0.204	0.275	1.047	0.974
		N(0,10)	1.701	1.789	1.868	1.961	1.983	0.327	0.286	0.291	0.371	0.423
		t(1)	0.502	0.639	0.370	0.388	0.983	0.297	0.261	0.265	0.340	0.405
		t(5)	0.473	0.618	0.468	0.956	0.977	0.292	0.252	0.255	0.337	0.399
	15%	N(0,6)	1.947	1.693	1.741	1.943	2.978	0.297	0.293	0.295	1.047	0.970
		N(0,10)	2.644	2.735	2.953	2.543	2.984	0.354	0.312	0.316	0.401	0.461
		t(1)	0.526	0.655	0.934	0.916	0.984	0.292	0.255	0.258	0.334	0.400
		t(5)	0.465	0.610	0.522	0.643	0.981	0.276	0.243	0.246	0.319	0.384
	20%	N(0,6)	2.656	2.742	2.633	2.548	2.982	0.373	0.353	0.369	1.049	0.971
		N(0,10)	2.899	2.889	2.550	3.475	2.998	0.519	0.229	0.242	0.744	0.701
		t(1)	1.900	1.895	1.526	1.526	0.997	0.475	0.183	0.198	0.728	0.681
		t(5)	1.877	1.884	1.481	1.387	1.992	0.470	0.167	0.182	0.729	0.692
T=200	0%		0.500	0.411	0.478	0.785	0.732	0.118	0.012	0.031	0.477	0.396
	5%	N(0,6)	0.790	0.618	0.691	0.818	0.789	0.154	0.014	0.044	0.540	0.407
		N(0,10)	0.819	0.711	0.792	0.920	0.820	0.158	0.058	0.106	0.585	0.458
		t(1)	0.768	0.709	0.741	0.835	0.818	0.139	0.004	0.024	0.425	0.393
		t(5)	0.719	0.681	0.692	0.820	0.802	0.134	0.033	0.108	0.487	0.398
	10%	N(0,6)	0.827	0.687	0.745	0.861	0.826	0.165	0.015	0.064	0.543	0.446
		N(0,10)	0.879	0.791	0.799	0.982	0.913	0.139	0.073	0.103	0.438	0.382
		t(1)	0.792	0.736	0.773	0.892	0.853	0.169	0.074	0.124	0.428	0.396
		t(5)	0.784	0.719	0.702	0.891	0.820	0.142	0.022	0.107	0.508	0.474
	15%	N(0,6)	0.896	0.731	0.824	0.925	0.896	0.153	0.079	0.103	0.479	0.408
		N(0,10)	0.924	0.810	0.861	1.092	0.963	0.171	0.009	0.030	0.502	0.413
		t(1)	0.801	0.761	0.795	0.940	0.874	0.115	0.024	0.081	0.558	0.476
		t(5)	0.822	0.745	0.761	0.923	0.848	0.094	0.109	0.009	0.095	0.401
	20%	N(0,6)	0.914	0.811	0.871	0.975	0.669	0.147	0.023	0.100	0.445	0.378
		N(0,10)	1.068	0.908	0.986	2.071	1.651	0.181	0.018	0.079	0.410	0.406
		t(1)	0.958	0.890	0.966	0.985	0.899	0.160	0.057	0.128	0.516	0.453
		t(5)	0.895	0.795	0.856	0.961	0.877	0.130	0.008	0.061	0.433	0.351

Table (3-12) RPE^w for methods when $\sigma = 5$ (very sparse case/M-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
T=50	0%		0.692	0.634	0.653	0.881	0.867	0.353	0.144	0.246	0.436	0.414
	5%	N(0,6)	0.806	0.638	0.742	0.994	0.978	0.355	0.154	0.257	0.546	0.545
		N(0,10)	0.925	0.890	0.893	1.259	1.274	0.351	0.151	0.255	0.538	0.537
		t(1)	0.920	0.840	0.916	1.000	0.998	0.351	0.079	0.234	0.511	0.507
		t(5)	0.813	0.806	0.810	0.981	0.981	0.308	0.091	0.166	0.490	0.416
	10%	N(0,6)	0.924	0.839	0.918	0.998	0.990	0.222	0.027	0.156	0.529	0.506
		N(0,10)	0.969	0.916	0.981	1.657	1.561	0.270	0.013	0.176	0.570	2.517
		t(1)	0.946	0.845	0.943	1.005	1.000	0.140	0.018	0.046	0.489	0.407
		t(5)	0.832	0.813	0.823	0.983	0.982	0.235	0.087	0.132	0.582	0.491
	15%	N(0,6)	0.985	0.895	0.941	1.000	0.994	0.275	0.129	0.225	0.468	0.380
		N(0,10)	0.990	0.951	0.997	1.827	1.641	0.285	0.079	0.113	0.570	0.505
		t(1)	0.972	0.877	0.950	1.013	1.002	0.230	0.042	0.145	0.432	0.421
		t(5)	0.892	0.827	0.836	0.994	0.990	0.370	0.037	0.165	0.538	0.472
	20%	N(0,6)	0.996	0.900	0.975	1.056	1.566	0.221	0.075	0.133	0.442	0.407
		N(0,10)	1.015	0.991	1.001	2.106	2.139	0.288	0.099	0.174	0.541	0.501
		t(1)	0.993	0.898	0.980	1.339	1.306	0.351	0.134	0.247	0.534	0.504
		t(5)	0.993	0.851	0.871	0.998	0.993	0.279	0.083	0.128	0.447	0.433
T=100	0%		0.903	0.888	0.494	0.988	1.028	0.500	0.188	0.203	0.752	0.679
	5%	N(0,6)	0.790	0.618	0.784	1.362	0.979	0.263	0.259	0.263	1.047	0.972
		N(0,10)	0.901	0.876	0.951	1.425	1.029	0.494	0.188	0.205	0.742	0.661
		t(1)	0.902	0.882	0.909	1.429	1.034	0.488	0.193	0.207	0.731	0.656
		t(5)	0.906	0.880	0.892	1.420	1.032	0.480	0.188	0.201	0.746	0.663
	10%	N(0,6)	0.875	0.733	0.977	1.368	0.993	0.471	0.200	0.206	0.702	0.697
		N(0,10)	0.991	0.896	0.571	1.425	1.029	0.494	0.188	0.205	0.742	0.661
		t(1)	0.992	0.892	0.585	1.429	1.034	0.488	0.193	0.207	0.731	0.656
		t(5)	0.996	0.900	0.592	1.420	1.032	0.480	0.188	0.201	0.746	0.663
	15%	N(0,6)	0.864	0.762	0.984	1.377	0.990	0.275	0.175	0.272	1.047	0.967
		N(0,10)	0.900	0.889	1.093	1.449	1.025	0.505	0.202	0.218	0.746	0.672
		t(1)	0.912	0.890	1.022	1.452	1.038	0.482	0.184	0.198	0.752	0.666
		t(5)	0.894	0.873	0.485	1.409	1.021	0.475	0.180	0.196	0.727	0.654
	20%	N(0,6)	0.851	0.764	0.852	1.403	0.984	0.217	0.215	0.216	0.747	0.673

		N(0,10)	0.904	0.886	0.953	1.467	1.027	0.501	0.214	0.229	0.751	0.670
		t(1)	0.902	0.879	0.951	1.459	1.030	0.475	0.180	0.196	0.722	0.654
		t(5)	0.896	0.882	0.923	1.414	1.029	0.475	0.172	0.189	0.725	0.664
T=200	0%		0.510	0.404	0.477	0.811	0.803	0.213	0.108	0.178	0.531	0.462
	5%	N(0,6)	0.759	0.621	0.675	0.940	0.905	0.247	0.145	0.200	0.498	0.468
		N(0,10)	0.899	0.711	0.799	0.979	0.922	0.289	0.108	0.176	0.544	0.456
		t(1)	0.728	0.664	0.709	0.906	0.862	0.238	0.102	0.137	0.478	0.413
		t(5)	0.687	0.632	0.655	0.826	0.802	0.110	0.016	0.032	0.531	0.416
	10%	N(0,6)	0.869	0.685	0.714	0.970	0.905	0.258	0.106	0.178	0.526	0.452
		N(0,10)	0.923	0.787	0.852	1.023	0.986	0.214	0.059	0.167	0.513	0.479
		t(1)	0.867	0.736	0.785	0.986	0.935	0.210	0.070	0.102	0.553	0.463
		t(5)	0.712	0.700	0.703	0.897	0.870	0.162	0.087	0.133	0.476	0.350
	15%	N(0,6)	0.907	0.733	0.774	0.998	0.963	0.155	0.046	0.154	0.544	0.465
		N(0,10)	0.984	0.828	0.896	1.459	1.083	0.180	0.082	0.107	0.497	0.379
		t(1)	0.972	0.794	0.819	1.178	0.998	0.117	0.019	0.074	0.512	0.496
		t(5)	0.807	0.761	0.791	0.954	0.926	0.180	0.080	0.103	0.497	0.384
	20%	N(0,6)	0.978	0.806	0.827	1.016	0.974	0.152	0.090	0.116	0.583	0.467
		N(0,10)	1.133	0.908	0.906	1.864	1.304	0.135	0.009	0.081	0.501	0.457
		t(1)	1.019	0.825	0.871	1.521	1.249	0.106	0.001	0.038	0.497	0.416
		t(5)	0.851	0.791	0.812	0.985	0.951	0.176	0.009	0.106	0.374	0.420

Tables (3-10), (3-11), and (3-12) show that the results of the ordinary methods are very close to our proposed robust methods when (σ = 1, 3, 5). However, when contaminating the data with different pollutant rates, the RPE^{w} values for the ordinary methods increase the percentage of contamination and are greatly affected by the increased variance of contaminated data when contaminating by natural distribution or The higher the degree of freedom when contaminating the t distribution, while the penalized robust methods remain resistant to the outlier, the greater the percentage of contamination. The M-lwlasso method with w^{R2} is the best in most experiments, followed by the preference of the M-lwlasso with $\mathrm{w}^{\mathrm{R}3}$ and M-lwlasso with $\mathrm{w}^{\mathrm{R}1}$ rahtin being the lowest value for do yield (RPE^{w}).

2-Proposed MM-lwlasso methods

After implementation of the program in appendix [2], the following results were obtained, based on the random error limit of the non- contaminated data.

Table (3-13) RPE^{w} for methods when $\sigma = 1$ (very sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
T=50	0%		1.495	1.435	1.454	2.588	2.454	1.171	1.116	1.123	2.452	2.433
	5%	N(0,6)	2.646	2.635	2.607	3.672	3.698	1.184	1.104	1.130	2.487	2.477
		N(0,10)	4.683	4.121	4.180	5.966	5.941	1.149	1.117	1.138	2.453	2.427
		t(1)	3.691	3.029	3.114	4.975	4.024	1.140	1.113	1.125	2.399	2.065
		t(5)	2.697	2.197	2.374	3.972	3.224	1.135	1.035	1.114	2.350	2.308
	10%	N(0,6)	3.774	3.412	3.510	4.775	4.529	1.135	1.075	1.106	2.120	2.043
		N(0,10)	5.125	5.043	5.094	6.276	6.055	1.142	1.098	1.109	2.144	2.122
		t(1)	4.379	4.171	4.263	6.528	6.449	1.100	1.049	1.082	2.482	2.419
		t(5)	3.344	3.068	3.145	4.110	4.070	1.343	1.070	1.109	2.339	2.286
	15%	N(0,6)	4.291	4.071	4.159	5.558	5.285	1.140	1.027	1.126	2.137	2.054
		N(0,10)	7.428	7.201	7.288	8.412	8.745	1.131	1.057	1.101	2.157	2.065
		t(1)	5.746	5.170	5.199	6.975	6.842	1.137	1.059	1.106	2.160	2.095
		t(5)	4.078	4.009	4.019	5.078	5.035	1.124	1.008	1.041	2.253	2.130
	20%	N(0,6)	5.702	5.227	5.346	6.503	6.277	1.818	1.906	1.456	2.345	2.084
		N(0,10)	8.329	8.102	8.158	8.881	8.679	1.226	1.086	1.169	2.242	2.188
		t(1)	6.904	6.349	6.454	7.220	7.470	1.165	1.065	1.133	2.165	2.135
		t(5)	5.188	5.126	5.142	6.131	6.141	1.275	1.044	1.184	2.410	2.420
T=100	0%		1.437	1.367	1.446	2.587	2.399	1.261	1.017	1.080	2.261	2.145
	5%	N(0,6)	2.776	2.489	2.545	3.486	3.433	1.161	1.105	1.127	2.181	2.163
		N(0,10)	3.957	3.803	3.845	4.862	4.025	1.294	1.070	1.104	2.133	2.005
		t(1)	4.272	4.083	4.175	4.833	4.889	1.182	1.026	1.135	2.592	2.258

		t(5)	2.650	2.511	2.517	3.167	3.119	1.279	1.069	1.186	2.279	2.046
	10%	N(0,6)	3.291	3.058	3.094	4.289	4.037	1.485	1.480	1.667	2.238	2.146
		N(0,10)	5.194	5.022	5.100	6.700	6.629	1.148	1.056	1.317	2.341	2.193
		t(1)	4.971	4.588	4.762	5.705	5.612	1.128	1.051	1.138	2.287	2.203
		t(5)	3.062	3.015	3.023	4.161	4.063	1.306	1.123	1.236	2.306	2.230
	15%	N(0,6)	4.905	4.179	4.144	5.209	5.146	1.179	1.094	1.133	2.467	2.124
		N(0,10)	6.506	6.159	6.248	7.818	7.774	1.310	1.148	1.267	2.382	2.120
		t(1)	5.873	5.761	5.838	6.973	6.897	1.178	1.089	1.128	2.328	2.293
		t(5)	4.155	3.976	4.103	5.074	5.008	1.149	0.994	1.028	2.272	2.229
	20%	N(0,6)	5.786	5.487	5.639	6.790	6.917	1.132	1.068	1.106	2.134	2.114
		N(0,10)	8.225	8.054	8.116	8.847	8.852	1.146	0.848	1.028	2.199	2.009
		t(1)	6.433	6.433	6.444	7.436	7.396	1.377	0.964	1.075	2.381	2.307
		t(5)	5.202	5.006	5.200	6.201	6.198	1.308	1.007	1.033	2.277	2.046
T=200	0%		1.829	1.521	1.718	2.884	2.876	1.228	1.023	1.130	2.228	2.176
	5%	N(0,6)	2.423	2.423	2.425	3.414	3.196	1.261	1.056	1.128	2.259	2.260
		N(0,10)	3.895	3.000	3.293	4.895	4.364	1.198	1.086	1.128	2.220	2.122
		t(1)	3.159	3.066	3.125	4.257	4.165	1.123	1.074	1.115	2.230	2.224
		t(5)	2.963	2.387	2.473	3.624	3.335	1.255	1.050	1.194	2.256	2.096
	10%	N(0,6)	3.363	3.081	3.206	4.149	4.057	1.135	1.067	1.094	2.352	2.303
		N(0,10)	4.138	4.076	4.099	5.134	5.210	1.144	1.024	1.081	2.288	2.214
		t(1)	3.846	3.726	3.804	4.769	4.671	1.226	1.102	1.165	2.288	2.250
		t(5)	3.203	3.013	3.184	4.020	3.963	1.145	1.014	1.056	2.218	2.116
	15%	N(0,6)	4.365	4.256	4.502	5.564	5.549	1.141	1.013	1.075	2.106	2.029
		N(0,10)	5.222	5.457	5.414	6.208	6.415	1.214	1.086	1.170	2.442	2.269
		t(1)	4.385	4.249	4.306	5.595	5.377	1.216	1.027	1.116	2.265	2.268
		t(5)	3.980	3.906	3.939	4.081	4.000	1.258	1.100	1.179	2.258	2.213
	20%	N(0,6)	5.280	5.375	5.356	7.279	7.134	1.111	1.032	1.066	2.312	2.229
		N(0,10)	7.062	6.899	7.008	8.650	8.056	1.104	0.972	1.057	2.305	2.297
		t(1)	5.122	4.786	5.082	5.867	5.824	1.178	1.038	1.167	2.358	2.133
		t(5)	4.152	4.112	4.121	5.153	5.107	1.262	1.108	1.127	2.262	2.222

Table (3-14) RPE^w for methods when $\sigma = 3$ (very sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
T=50	0%		1.812	1.748	1.769	2.120	2.144	1.156	1.024	1.062	2.200	2.191
	5%	N(0,6)	2.139	2.095	2.100	2.964	2.946	1.137	1.012	1.073	2.046	2.014
		N(0,10)	3.250	3.125	3.163	3.972	3.914	1.143	1.056	2.115	2.169	2.126
		t(1)	2.972	2.716	2.795	3.980	3.980	1.206	1.092	1.133	2.059	1.997
		t(5)	2.688	2.395	2.456	2.981	2.979	1.207	1.023	1.100	2.091	2.023
	10%	N(0,6)	2.249	2.126	2.192	3.031	3.029	1.145	1.075	1.114	2.047	2.018
		N(0,10)	3.429	3.297	3.365	3.992	3.853	1.111	1.039	1.083	2.008	2.002
		t(1)	3.023	3.004	3.013	3.918	3.902	1.117	1.020	1.102	2.122	2.069
		t(5)	2.861	2.518	2.722	2.997	2.984	1.144	1.098	1.116	2.143	2.114
	15%	N(0,6)	3.071	3.015	3.051	3.727	3.687	1.144	1.038	1.085	2.443	2.107
		N(0,10)	3.972	3.755	3.912	4.090	4.056	1.160	1.083	1.108	2.173	2.134
		t(1)	3.276	3.140	3.183	4.187	4.153	1.135	1.034	1.123	2.151	2.135
		t(5)	2.944	2.625	2.859	3.239	3.198	1.159	1.057	1.076	2.059	2.012
	20%	N(0,6)	3.692	3.129	3.188	4.130	4.117	1.165	1.047	1.087	2.065	2.047
		N(0,10)	4.090	3.981	4.034	4.762	4.612	1.155	1.036	1.124	2.139	2.115
		t(1)	3.310	3.218	3.130	3.931	3.825	1.036	1.002	1.008	2.042	2.038
		t(5)	3.057	2.719	2.858	3.471	3.351	1.048	1.007	1.013	1.889	1.772
T=100	0%		1.809	1.710	1.790	2.073	2.029	1.147	1.054	1.118	2.145	2.108
	5%	N(0,6)	2.309	2.123	2.258	2.966	2.922	1.104	1.030	1.098	2.052	2.032
		N(0,10)	3.270	3.199	3.207	3.872	3.711	1.134	1.034	1.114	2.143	2.105
		t(1)	2.988	2.892	2.923	3.099	3.007	1.168	1.063	1.134	2.184	2.169
		t(5)	2.394	2.060	2.163	2.792	2.704	1.147	1.028	2.108	2.148	2.139
	10%	N(0,6)	2.617	2.552	2.579	3.123	3.039	1.288	1.013	1.190	2.292	2.235
		N(0,10)	3.503	3.313	3.483	3.950	3.914	1.139	1.067	1.113	2.084	2.044
		t(1)	3.198	3.106	3.142	3.991	3.863	1.119	1.015	1.044	2.175	2.115
		t(5)	2.697	2.607	2.680	2.969	2.961	1.123	1.021	1.091	2.024	2.015
	15%	N(0,6)	3.153	3.015	3.097	3.963	3.922	1.124	1.004	1.033	2.243	2.133
		N(0,10)	3.921	3.806	3.891	4.216	4.114	1.034	1.008	1.013	2.020	2.010
		t(1)	3.570	3.399	3.428	4.039	4.017	1.101	1.022	1.059	2.099	2.016
		t(5)	2.931	2.823	2.897	2.991	2.984	1.042	1.009	1.013	2.043	2.011
	20%	N(0,6)	3.627	3.454	3.533	4.066	4.051	1.136	1.073	1.118	2.095	2.036

		N(0,10)	4.154	4.111	4.121	4.953	4.754	1.100	1.017	1.077	2.084	2.046
		t(1)	3.958	3.761	3.857	4.052	4.019	1.129	1.018	1.099	2.018	1.997
		t(5)	3.149	3.099	1.124	3.928	3.902	1.178	1.041	1.087	2.116	2.028
T=200	0%		1.805	1.717	1.798	2.033	2.023	1.112	1.048	1.089	2.092	2.004
	5%	N(0,6)	2.190	2.007	2.054	2.754	2.523	1.129	1.073	1.102	2.143	2.020
		N(0,10)	3.221	3.050	3.143	3.801	3.758	1.132	1.067	1.115	2.126	2.113
		t(1)	2.563	2.423	2.471	2.992	2.913	1.130	1.014	1.067	2.159	2.138
		t(5)	2.079	2.001	2.014	2.881	2.729	1.103	1.008	1.062	2.102	2.064
	10%	N(0,6)	2.818	2.742	2.775	2.979	2.927	1.128	1.027	1.072	2.275	2.208
		N(0,10)	3.576	3.207	3.282	3.981	3.848	1.059	1.013	1.042	2.080	2.013
		t(1)	2.762	2.532	2.591	3.300	3.132	1.097	1.005	1.070	2.066	2.057
		t(5)	2.607	2.426	2.505	2.951	2.946	1.128	1.091	1.110	2.036	1.973
	15%	N(0,6)	3.127	3.028	3.082	3.777	3.723	1.121	1.030	1.081	2.298	2.140
		N(0,10)	3.837	3.748	3.782	3.984	3.912	1.177	1.052	1.115	2.183	2.136
		t(1)	3.179	3.014	3.129	3.997	3.813	1.174	1.094	1.114	2.111	2.018
		t(5)	2.876	2.604	2.760	3.003	3.008	1.132	1.028	1.083	2.259	2.129
	20%	N(0,6)	3.801	3.638	3.768	3.994	3.972	1.185	1.033	1.111	2.287	2.152
		N(0,10)	4.189	4.033	4.124	4.614	4.506	1.106	1.023	1.067	2.180	2.013
		t(1)	3.624	3.512	3.585	4.007	4.002	1.143	1.054	1.106	2.189	2.138
		t(5)	2.958	2.821	2.923	3.231	3.136	1.106	1.021	1.072	2.174	2.042

Table (3-15) RPE^{w} for methods when $\sigma = 5$ (very sparse case/MM-lwlasso methods)

Sample size	outlier rate	Dist.	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
	0%		1.510	1.430	1.452	1.807	1.761	1.153	1.040	1.063	1.851	1.726
T=50	5%	N(0,5)	1.857	1.731	1.775	1.955	1.904	1.139	1.005	1.038	1.731	1.651
		N(0,10)	1.981	1.827	1.843	1.985	1.976	1.150	1.039	1.093	1.801	1.749
		t(1)	1.873	1.731	1.786	1.980	1.834	1.107	1.012	1.055	1.831	1.811
		t(5)	1.790	1.678	1.706	1.977	1.892	1.113	1.024	1.060	1.824	1.808
	10%	N(0,6)	1.905	1.761	1.799	1.987	1.960	1.138	1.019	1.080	1.843	1.809
		N(0,10)	2.076	1.964	1.991	2.143	2.107	1.165	1.035	1.110	1.866	1.797
		t(1)	1.935	1.799	1.829	2.106	2.067	1.151	1.022	1.086	1.915	1.890
		t(5)	1.818	1.744	1.815	2.076	1.969	1.101	1.089	1.098	1.777	1.674

	15%	N(0,6)	1.982	1.805	1.823	1.996	1.981	1.144	1.035	1.083	1.938	1.845
		N(0,10)	2.131	2.022	2.089	2.478	2.178	1.130	1.076	1.098	1.845	1.783
		t(1)	1.962	1.823	1.974	2.224	2.127	1.147	1.012	1.119	1.877	1.731
		t(5)	1.873	1.782	1.866	2.153	2.067	1.133	1.034	1.062	1.804	1.741
	20%	N(0,6)	2.020	1.871	1.913	2.319	2.173	1.144	1.038	1.092	1.893	1.849
		N(0,10)	2.690	2.288	2.315	2.866	2.746	1.104	1.048	1.092	1.850	1.808
		t(1)	2.087	1.923	2.035	2.382	2.350	1.146	1.037	1.073	1.834	1.775
		t(5)	1.984	1.819	1.934	2.424	2.349	1.105	1.017	1.089	1.892	1.833
T=100	0%		1.545	1.435	1.464	1.882	1.843	1.108	1.019	1.080	1.872	1.834
	5%	N(0,6)	1.797	1.715	1.764	1.907	1.864	1.129	1.023	1.097	1.942	1.830
		N(0,10)	1.898	1.807	1.832	1.965	1.922	1.135	1.053	1.106	1.854	1.836
		t(1)	1.762	1.677	1.690	1.964	1.916	1.174	1.059	1.112	1.775	1.699
		t(5)	1.779	1.615	1.682	1.817	1.805	1.166	1.058	1.071	1.801	1.764
	10%	N(0,6)	1.831	1.780	1.813	1.949	1.901	1.132	1.016	1.068	1.832	1.817
		N(0,10)	1.922	1.855	1.894	1.986	1.956	1.144	1.024	1.063	2.779	1.743
		t(1)	1.813	1.745	1.784	1.981	1.946	1.118	1.031	1.059	1.795	1.684
		t(5)	1.807	1.670	1.697	1.932	1.984	1.090	2.002	1.067	1.811	1.801
	15%	N(0,6)	1.879	1.791	1.868	1.988	1.951	1.136	1.058	1.104	1.965	1.778
		N(0,10)	1.985	1.908	1.957	1.989	1.971	1.149	1.059	1.125	1.897	1.760
		t(1)	1.862	1.710	1.735	1.997	1.982	1.119	1.022	21.064	1.886	1.871
		t(5)	1.860	1.702	1.718	1.960	1.996	1.109	1.006	1.089	1.839	1.811
	20%	N(0,6)	1.908	1.834	1.872	1.999	1.984	1.136	1.013	1.065	1.756	1.736
		N(0,10)	2.015	1.984	2.008	2.188	2.144	1.128	1.018	1.079	1.877	1.750
		t(1)	1.962	1.772	1.793	2.163	2.118	1.121	1.013	1.099	1.841	1.792
		t(5)	1.964	1.810	1.871	2.082	2.027	1.157	1.077	1.101	1.775	1.691
T=200	0%		1.581	1.507	1.559	1.879	1.810	1.109	1.018	1.052	1.851	1.806
	5%	N(0,6)	1.772	2.736	1.742	1.843	1.842	1.125	1.044	1.103	1.826	1.802
		N(0,10)	1.872	1.815	1.853	1.973	1.953	1.122	1.055	1.105	1.831	1.805
		t(1)	1.743	1.705	1.749	1.868	1.805	1.164	1.027	1.082	1.864	1.816
		t(5)	1.712	1.644	1.694	1.922	1.911	1.112	1.078	1.102	1.842	1.810
	10%	N(0,6)	1.801	1.753	1.795	1.900	1.861	1.129	1.028	1.090	1.852	1.839
		N(0,10)	1.930	1.853	1.881	1.995	1.971	1.140	1.006	1.064	1.782	1.741
		t(1)	1.866	1.769	1.833	1.963	1.929	1.148	1.015	1.085	1.907	1.851
		t(5)	1.796	1.700	1.700	1.971	1.965	1.099	1.004	1.021	1.757	1.724
	15%	N(0,6)	1.877	1.816	1.856	1.999	1.940	1.127	1.026	1.083	1.830	1.801
		N(0,10)	1.986	1.920	1.963	2.098	2.035	1.139	1.014	1.090	1.894	1.806

		t(1)	1.984	1.923	1.953	2.129	2.084	1.168	1.007	1.102	1.868	1.820
		t(5)	1.851	1.766	1.702	1.997	1.987	1.149	1.012	1.114	1.772	1.724
	20%	N(0,6)	1.981	1.875	1.917	2.045	2.021	1.133	1.032	1.106	1.903	1.863
		N(0,10)	2.107	2.055	2.085	2.751	2.732	1.101	1.019	1.084	1.966	1.757
		t(1)	2.034	1.961	2.019	2.582	2.368	1.134	1.023	1.088	1.842	1.828
		t(5)	1.923	1.837	1.782	2.052	2.005	1.119	1.007	1.017	1.784	1.716

In Table (3-13), (3-14) and (3-15) it is observed that there is a convergence in the results between the PLS methods and the penalized robust methods when the data are free from the outlier. However, when contaminating the data with different contamination rates, the RPE^{w}values for the PLS methods increase the percentage of contamination, Polluted by contamination by normal distribution or by increasing the degree of freedom when polluting by t distribution, while penalized robust methods remain resistant to outliers, no matter how much contamination. The MM-lwlasso method with $\widehat{w}^{R2}$ is the best in most experiments, followed by the MM-lwlasso with $\widehat{w}^{R3}$and MM-lwlasso with $\widehat{w}^{R1}$ that give the least value of RPE^{w}.

Chapter Four

Applied part

4-1 Introduction

Lung is the primary organ of the respiratory system responsible for breathing in human lungs located within the thorax of the spine on both sides of the heart. During which the exchange of gases (oxygen absorption and conversion into the bloodstream, and the removal of carbon dioxide from the blood into the atmosphere). The right lung is larger than the left, and the lungs together weigh about 3.1 kg (9.2 lbs).

Lung cancer is characterized by irregular, irregular growth of lung cells. Lung cancer is not only the most common type of cancer in the world, but also the most deadly type of cancer around the world. Primary symptoms of lung cancer may include shortness of breath, respiratory wheezing, and chronic cough, with or without a blood-stained sputum. Lung cancer causes complications such as fluid accumulation in pulmonary cavities and pneumonia. The most serious complication its ability to spread to other parts of the body and cause new cancers. In extreme cases, within one year of the disease, about 3 out of every five people with this type of cancer die. Lung cancer can be divided into several types depending on the nature and location of cancer cells: small cell lung cancer, non-small cell lung cancer, squamous cell lung cancer, and lung cancer with large cancer cells. In addition, lung cancer can be secondary to the origin, meaning that it occurred in the lung as a result of primary cancer originated in other sites of the body and then moved to the lung. It is important to know the classification of lung cancer in any patient to determine the appropriate treatment. Arab and world lung cancer is spreading at very high rates. In 2012, 1.8 million cases of lung cancer were reported, resulting in 1.6 million deaths. This makes it the most common cause of death related to cancer in men and second most common in women after breast cancer. The most common age at diagnosis is 70 years of age. Overall, 17.4% of people in the united states diagnosed with lung cancer survive five years after being diagnosed with cancer, but these results are different in the developing world. In this chapter, real data from Diwaniya Hospital will be used for people suffering from lung cancer. Some of the risk factor that are considered independent variables that lead to lung cancer are the chemical pollutants of water which include 6 chemical pollutants mentioned in item 4.2, Obtain data from the Diwaniyah Water Department. The results of the application were obtained through programs written in (R) by the researcher and are shown in appendixes [3] and [4].

4-2 Describe the variables of the study

In this chapter, the performance methods are illustrated using lung cancer data. These data were collected from an Iraqi medical center in Diwaniyah city, Iraq. They, represent monthly numbers of people with lung cancer in Diwaniyah city, from April 2004 to September 2015. These data consist of one response variable (lung cancer) and 6 chemical water pollutants (Temp, Turb, PH, EC, AlK, TH), as shown in Table (4-1).

Table (4-1) chemical water pollutants

x_1	Temperature(Temp)
x_2	Turbidity (Turb)
x_3	Power of hydrogen (PH)
x_4	Electric conductivity (EC)
x_5	Alkility (Alk)
x_6	Total hardness (TH)

The time series of studied phenomenon is plotted using R program as shown in the figure (4-1):

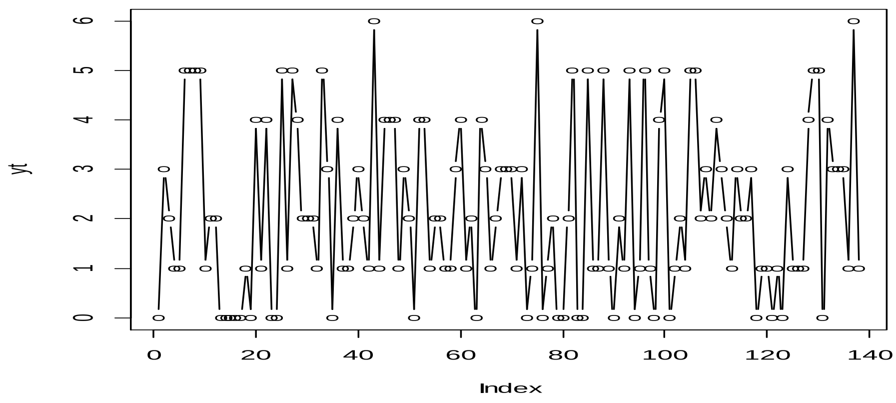

Figure (4-1) represents the data sketch prior to contamination

The data were polluted by 5% of outlier of the AO type by replacing clean observations with outlier ones whose values are high and then we used plot again as in the following figure.

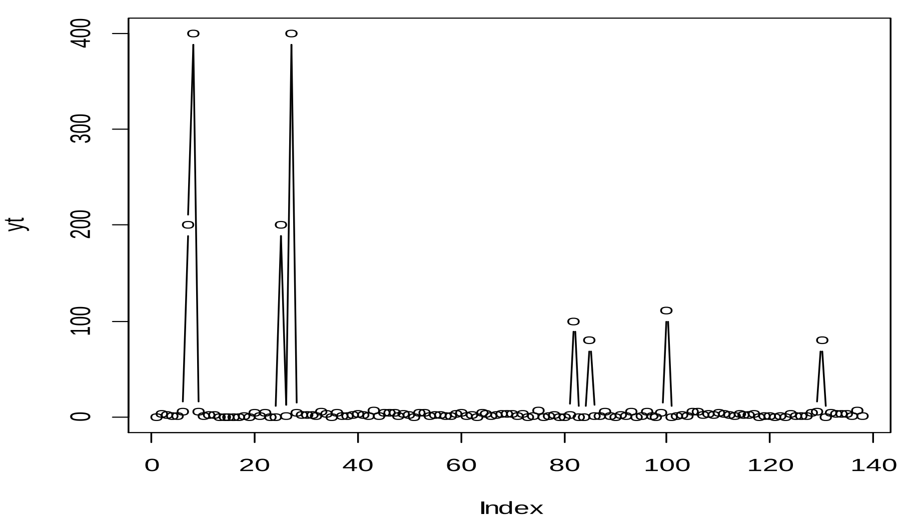

Figure (4-2) represents the data plot after contamination by 5% of the outliers

4-3 The most prominent specifications and behavior of the time series

By plotting the time series of the phenomenon studied

1-In Fig. (4-1) there are no outlier values in the data

2-An outliers is specified in Figure (4-2)

Based on the R program, which is based on the determination of outliers in the time series, it is based on the method proposed by researchers Jong and Penzer in 1998. In this method, outliers can be identified by diagnosing abnormal values by estimating the value at time t_0, depending on the rest of the values of the time series, this estimate is made for the AO values by taking the mean difference between the outlier and the values that precede it and the next and then comparing the estimated value with the real value. If the difference between them is statistically significant, this value is considered as an outlier and the statistic τ^2 is used for the purpose of finding the test. A square of one degree of freedom and a certain level of significance is calculated by the following formula:

$$\tau^2 = \frac{\delta^2}{k\sigma^2} \qquad (4-1)$$

Where δ is the difference estimation, k is constant and is calculated according to the method described by Jong and Penzer in 1998, where its value here is k = 0.27, and the standard deviation is σ = 53.01241.

3-There are outliers in the studied time series of AO type, which are the sequences (7,8,25,27,82,85,100,130).

4-The studied time series doesn't contain IO outliers.

5-There is no level shift (LS) of the time series.

4-4 Data analysis

First: The results of the analysis before adding outliers

In order to know the coefficients affecting lung cancer، the best penalized robust estimators have been applied in the experimental side as follows:

1- M-lwlasso methods

The estimations were obtained after application of the program in appendix [3] and after the introduction of real data prior to contamination as in Table (4-2).

Table (4-2) represents the estimated values of the parameters using the proposed M-lwlasso methods

B for M-lwlasso with w^{R1}	B for M-lwlasso with w^{R2}.	B.w3.r B for M-lwlasso with w^{R3}	B for M-lasso [1]	B for M-alasso
0.000	-0.031	-0.048	-0.049	-0.021
-0.070	-0.134	-0.165	-0.151	-0.117
0.004	0.031	0.038	0.038	0.051
0.000	-0.043	-0.052	-0.065	-0.096
0.000	0.000	0.000	0.000	0.028
0.000	-0.025	-0.062	-0.034	-0.069
0.039	0.107	0.068	0.080	0.100
-0.138	-0.224	-0.198	-0.256	-0.200
-0.057	-0.109	-0.092	-0.097	-0.052
-0.013	-0.154	-0.134	-0.148	-0.167
0.000	0.000	0.000	0.000	0.027
-0.084	-0.334	-0.209	-0.255	-0.210
0.000	0.000	0.029	0.029	0.058
0.000	0.000	0.000	0.000	0.041
0.000	-0.005	-0.015	-0.015	-0.046
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.004	0.004	0.011
0.000	0.000	0.023	0.012	0.124
0.000	-0.036	-0.057	-0.067	-0.095
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.021

0.000	0.027	0.019	0.021	0.050
0.000	0.000	0.000	0.000	-0.110
0.000	0.002	0.014	0.017	0.085
0.000	-0.076	-0.092	-0.082	-0.144
0.021	0.018	0.013	0.013	0.010
0.100	0.124	0.148	0.136	0.140
0.000	0.016	0.021	0.021	0.044
0.040	0.044	0.045	0.056	0.063
0.000	-0.013	-0.023	-0.028	-0.018
0.000	0.025	0.008	0.005	0.000
0.000	0.038	0.037	0.043	0.037
0.000	0.000	0.000	0.000	-0.002
0.000	-0.042	-0.029	-0.031	-0.046
0.000	0.000	0.000	0.000	0.025
0.000	0.051	0.020	0.017	0.000
0.111	0.289	0.183	0.223	0.174
-0.032	-0.155	-0.086	-0.077	-0.081
-0.084	-0.340	-0.660	-0.673	-0.961
0.000	0.119	0.261	0.240	0.355
0.000	0.054	0.007	0.007	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	-0.001
0.295	1.809	1.841	1.006	2.026
0.000	0.000	0.000	0.000	-0.514
0.000	-0.281	-0.164	-0.211	-0.198
0.000	-0.518	-0.466	-0.491	-0.588
0.000	0.000	0.000	0.000	0.253
-0.324	-1.940	-1.275	-1.099	-1.080
0.000	0.410	0.292	0.356	0.547
0.000	0.194	0.093	0.083	-0.043
0.000	0.000	0.000	0.000	-0.001
0.000	0.000	0.000	0.000	-0.005
0.000	0.000	-0.001	-0.001	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.002
0.000	0.000	0.000	0.000	0.000
0.000	0.000	-0.001	-0.001	-0.001
0.000	0.001	0.001	0.001	0.002
0.000	0.000	0.000	0.000	0.000
0.000	0.001	0.001	0.001	0.001
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.018	-0.028	-0.026	-0.042
0.000	0.013	0.020	0.020	0.028

0.000	0.008	0.014	0.018	0.023
0.000	0.004	0.007	0.009	0.006
0.000	-0.035	-0.046	-0.025	-0.043
0.000	0.023	0.025	0.030	0.028
0.000	0.000	-0.002	-0.002	-0.004
0.000	0.015	0.010	0.011	0.014
0.010	0.024	0.022	0.024	0.028
0.000	-0.004	-0.015	-0.013	-0.033
0.000	0.000	0.005	0.007	0.006
0.000	0.000	0.000	0.000	0.005
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.002	0.002	0.019
0.000	0.000	0.000	0.000	-0.003
0.000	0.000	-0.001	-0.001	-0.002
0.000	0.000	0.000	0.000	-0.007
0.000	0.000	0.003	0.002	0.007
0.000	-0.002	-0.001	-0.001	-0.002
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	-0.002
0.000	0.000	0.000	0.000	0.000
0.001	0.006	0.005	0.004	0.006
0.000	-0.001	-0.001	-0.001	0.000
0.000	0.000	-0.001	-0.001	0.000

Note from the table above that parameter values are stable for most of the proposed M-lwlasso methods. The RPE^w criterion was then extracted, as shown in Table (4-3), was then compared to the PLS methods and the penalized robust methods and the identification of the best methods.

Table (4-3) represents the RPE^w values of the PLS methods and the penalized robust methods (proposed M-lwlasso methods) before adding outliers

RPE^w	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	Lasso	Alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	M_lasso	M_alasso
	0.958	0.783	0.710	1.047	1.010	0.678	0.563	0.401	0.847	0.758

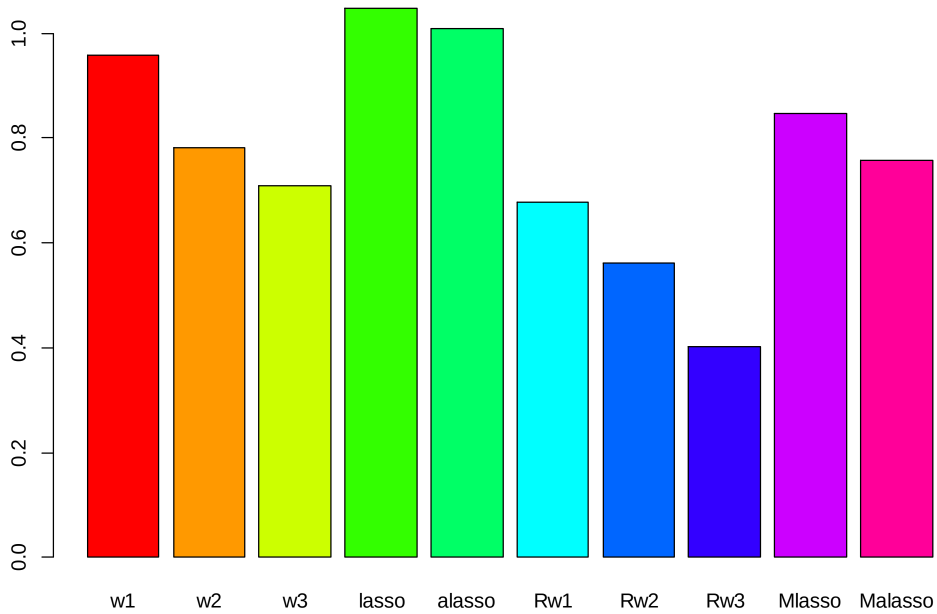

Figure (4-3) represents RPE^{w} of the PLS methods and the penalized robust methods (proposed M-lwlasso methods) before adding outliers

In the table and figure above, the results were similar to those between the PLS methods and the proposed M-lwlasso methods. However, the M-lwlasso method gave $\widehat{w}^{R3}$ less RPE^{w} with a value of 0.401 followed by the M-lwlasso method with $\widehat{w}^{R2}$ at 0.563 and then the M-lwlasso method with $\widehat{w}^{R1}$ at 0.678.

2-Proposed MM-lwlasso methods

Estimators were obtained after application of the program in appendix[4] and after the introduction of real data prior to contamination as in Table (4-4).

Table (4-4) represents the estimated values of the parameters using the proposed MM-lwlasso methods before polluting the data

B for MM-lwlasso with w^{R1}	B for MM-lwlasso with w^{R2}	B for MM-lwlasso with w^{R3}	B for MM-lasso	B for MM-alasso
-0.136	-0.053	-0.242	-0.058	-0.078
-0.152	-0.045	-0.179	-0.067	-0.064
0.090	0.017	0.022	0.033	0.022
-0.045	-0.021	-0.091	0.000	0.000
0.213	0.062	0.082	0.000	0.000
-0.196	-0.069	-0.129	0.000	0.000
-0.028	-0.016	-0.060	0.000	0.000
-0.332	-0.136	-0.216	0.000	0.000
0.077	0.025	0.000	-0.046	-0.036
-0.480	-0.235	-0.259	0.000	0.000
0.136	0.091	0.152	0.000	0.000
-0.226	-0.113	-0.121	-0.032	-0.025
-0.253	-0.090	-0.257	-0.006	-0.008
0.357	0.095	0.199	0.000	0.000
-0.177	-0.037	-0.074	0.000	0.000
0.142	0.059	0.124	0.000	0.000
0.182	0.057	0.111	0.000	0.000
-0.052	-0.015	0.000	0.000	0.000
-0.396	-0.165	-0.279	0.000	0.000
0.461	0.194	0.294	0.022	0.019
-0.022	-0.004	0.000	0.000	0.000
0.085	0.043	0.082	0.000	0.000
-0.319	-0.195	-0.215	0.000	0.000
0.580	0.293	0.304	0.000	0.000
-0.270	-0.184	-0.111	0.000	0.000
0.039	0.014	0.035	0.013	0.017
0.137	0.040	0.137	0.116	0.107
0.082	0.018	0.054	0.013	0.009
0.058	0.025	0.068	0.030	0.036
0.011	0.001	-0.003	0.000	0.000
0.097	0.037	0.080	0.000	0.000
0.100	0.044	0.090	0.000	0.000
-0.086	-0.033	-0.023	-0.023	-0.021
-0.116	-0.042	-0.030	0.000	0.000
0.130	0.066	0.076	0.000	0.000
-0.170	-0.103	-0.094	0.000	0.000
0.336	0.168	0.161	0.041	0.031
-0.034	-0.033	-0.078	0.000	-0.001
-0.953	-0.339	-0.901	0.000	0.000

0.208	0.052	0.000	0.000	0.000
-1.790	-0.359	-0.709	0.000	0.000
0.171	0.081	0.258	0.000	0.000
-0.241	-0.072	-0.123	0.000	0.000
1.131	0.440	1.011	0.204	0.189
-0.429	-0.163	-0.194	0.000	0.000
1.558	0.659	0.987	0.000	0.000
-1.612	-0.648	-0.932	0.000	0.000
2.141	1.082	1.243	0.000	0.000
-1.795	-1.205	-1.872	-0.210	-0.223
0.228	0.063	-0.122	0.000	0.000
0.629	0.515	0.749	0.128	0.137
-0.001	0.000	0.000	0.000	0.000
-0.012	-0.003	-0.005	0.000	0.000
0.000	0.000	-0.001	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.006	0.002	0.000	0.000	0.000
0.013	0.005	0.005	0.000	0.000
0.000	0.000	-0.001	0.000	0.000
-0.006	-0.002	0.000	0.000	0.000
0.001	0.001	0.001	0.000	0.000
0.000	0.001	0.004	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
-0.024	-0.008	-0.014	0.000	0.000
-0.037	-0.010	-0.025	0.000	0.000
0.030	0.006	0.015	0.000	0.000
0.034	0.014	0.027	0.000	0.000
0.000	0.000	0.000	-0.007	-0.006
0.004	0.000	0.000	0.000	0.000
-0.009	-0.002	-0.001	0.000	0.000
-0.006	-0.004	-0.009	0.000	0.000
0.002	0.002	0.003	0.000	0.000
0.043	0.022	0.026	0.000	0.000
-0.111	-0.068	-0.058	0.000	0.000
0.014	0.008	0.006	0.000	0.000
0.027	0.020	0.021	0.000	0.000
0.003	0.000	0.000	0.000	0.000
0.038	0.010	0.016	0.001	0.001
-0.004	-0.001	0.000	0.000	0.000
-0.006	-0.002	-0.005	0.000	0.000
-0.022	-0.006	-0.002	0.000	0.000
-0.029	-0.010	-0.011	0.000	0.000
-0.015	-0.005	-0.006	0.000	0.000
0.027	0.010	0.006	0.000	0.000

-0.001	0.000	-0.001	0.000	0.000
-0.004	-0.004	-0.014	0.000	0.000
0.016	0.011	0.012	0.003	0.003
0.002	0.001	0.000	0.000	0.000
0.001	0.001	0.003	0.000	0.000

Through the table (4-4) shows that the parameters are stable for MM-lwlasso methods. Table (4-5) shows represents the RPE^{w} values of the PLS methods and the proposed MM-lwlasso methods before adding outliers.

Table (4-5) represents the RPE^{w} values of the PLS methods and the penalized robust methods (proposed MM-lwlasso methods) before adding outliers.

RPE	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	Lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	MM_lasso	MM_alasso
	0.958	0.783	0.710	1.047	1.010	0.831	0.681	0.635	0.871	0.844

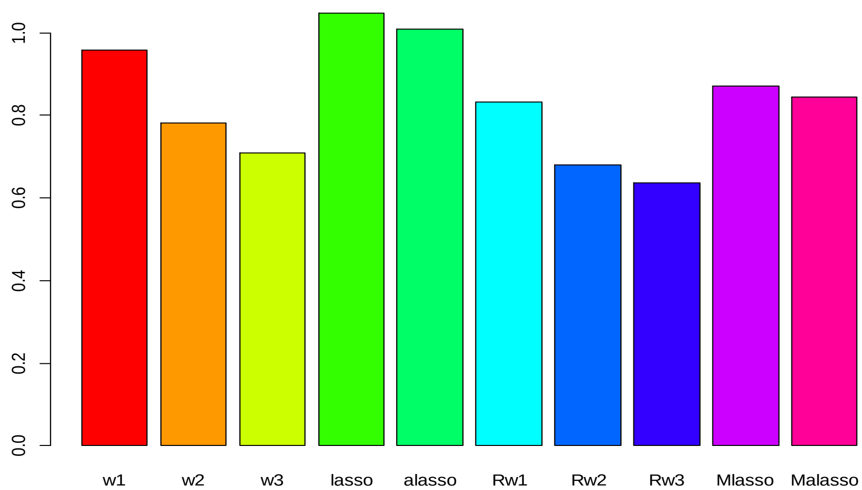

Figure (4-4) represents RPE^{w} of the PLS methods and the penalized robust methods (proposed MM-lwlasso methods) before adding outliers.

Note from the table and figure above, the results between the PLS methods and the proposed MM-lwlasso methods are similar. However, the MM-lwlasso method gave $\widehat{w}^{R3}$ with a lower RPE^{w} of 0.635 followed by MM-lwlasso with $\widehat{w}^{R2}$ at 0.681 and MM-lwlasso with $\widehat{w}^{R1}$ at 0.831.

Second: The results of the analysis after pollution of the data

In the case of clean data, there was a convergence of performance where the comparison criteria were close to each other. When data were contaminated with 5% of the AO type outliers, all PLS methods were affected, but the proposed penalized robust methods remained resistant to these outliers, the proposed methods are described in detail.

1-M-lwlasso methods

After implementation of the program in appendix [3] using contaminated data, the following results were obtained:

Table (4-6) represents the estimated values of the parameters using the proposed M-lwlasso methods after adding 5% of outliers

B for M-lwlasso with w^{R1}	B for M-lwlasso with w^{R2}	B for M-lwlasso with w^{R3}	B for M-lasso	B for M-alasso
-0.005	-0.017	-0.016	-0.025	-0.006
0.074	0.050	0.083	0.102	0.038
0.000	0.000	0.000	0.000	0.000
-0.015	-0.029	-0.026	-0.041	-0.010
0.003	0.010	0.010	0.010	0.003
0.000	0.007	0.000	0.009	0.000
0.000	0.004	0.000	0.003	0.000
0.000	-0.014	-0.001	-0.011	0.000
0.000	-0.007	0.000	-0.002	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.007	0.000	-0.003	0.000
-0.001	-0.017	-0.005	-0.009	-0.001
0.000	0.000	0.000	0.000	0.000
0.043	0.090	0.084	0.188	0.029
0.000	0.000	0.000	0.000	0.000

0.000	-0.049	0.000	-0.071	0.000
0.000	0.030	0.000	0.035	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.386	0.000	-0.239	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.106	0.000	0.052	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.010	0.000
0.000	-0.157	0.000	-0.089	0.000
-0.015	-0.274	-0.057	-0.215	-0.011
0.000	0.000	0.000	0.000	0.000
0.131	0.057	0.147	0.113	0.069
0.062	0.264	0.245	0.273	0.080
-0.412	-0.497	-0.572	-0.707	-0.245
-0.208	-0.446	-0.443	-0.401	-0.179
0.094	0.348	0.221	0.398	0.068
0.024	0.156	0.094	0.091	0.034
0.000	-0.228	-0.052	-0.180	0.000
-0.030	-0.507	-0.202	-0.162	-0.060
0.000	0.000	0.000	0.011	0.000
0.000	0.000	0.000	0.000	0.000
0.060	0.290	0.133	0.153	0.041
0.000	-0.077	-0.002	-0.069	0.000
0.000	0.861	0.142	1.348	0.000
0.000	-0.673	-0.295	-1.359	-0.023
0.000	0.000	0.000	0.000	0.000
0.772	0.975	1.372	1.339	0.510
0.000	0.848	0.020	0.789	0.000
0.447	0.344	0.785	0.306	0.280
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	3.052	0.000	0.959	0.000
-0.601	-0.179	-0.842	-0.112	-0.349
0.000	-0.288	0.000	-0.186	0.000
0.000	0.000	0.000	0.000	0.000
-0.332	-0.189	-0.159	-0.227	-0.094
0.000	0.001	0.000	0.001	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.001	0.000	0.002	0.000	0.001
0.000	0.003	0.000	0.003	0.000
0.000	-0.001	0.000	-0.001	0.000
0.000	-0.005	0.000	-0.003	0.000
0.001	0.005	0.002	0.004	0.001
0.000	0.000	0.000	0.000	0.000
0.001	0.004	0.001	0.003	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.006	0.000	-0.007	0.000
0.000	0.005	0.000	0.012	0.000
0.000	0.018	0.000	0.020	0.000
-0.015	-0.015	-0.026	-0.021	-0.010

0.000	-0.102	-0.009	-0.097	0.000
0.000	0.023	0.000	0.029	0.000
0.000	0.019	0.000	0.014	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.014	0.000	0.005	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.042	0.000	-0.020	0.000
-0.023	-0.124	-0.053	-0.068	-0.016
0.000	0.000	0.000	0.009	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.001	0.006	0.000	0.008	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	-0.003	0.000	-0.003	0.000

After we see the table above we find that the parameter are stable for M-lwlasso methods. As we can see in table (4-7) that the RPE criterion was then extracted.

Table (4-7) represents the RPE^{w} values of the PLS methods and the penalized robust methods (proposed M-lwlasso methods) after adding outliers.

RPE	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	Lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	Mlasso	Malasso
	1.055	0.998	0.872	1.184	1.135	0.673	0.565	0.407	0.843	0.727

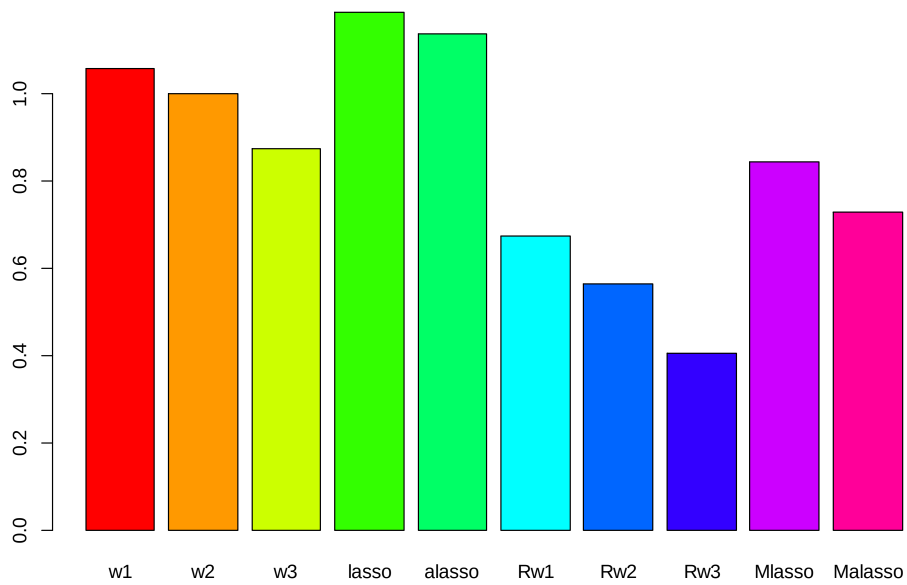

Figure (4-5) represents RPE^w of the PLS methods and the penalized robust methods (proposed M-lwlasso methods) after adding outliers.

It was noted in table (4-7) that the value of the RPE^w criterion for PLS methods increased by 5% in the data contamination. It was also observed that the good results obtained by the penalized robust methods (M-lwlasso methods) Where the method M-lwlasso with $\widehat{w}^{R3}$ less RPE^w at 0.407 followed by the method M-lwlasso with $\widehat{w}^{R2}$ value 0.565 and then the method M-lwlasso with $\widehat{w}^{R1}$ value 0.673. Figure (4-5) graphically shows the values of RPE^w for the methods. We can clearly see that the proposed methods have the smallest values of RPE^w compared with the other methods.

2-MM-lwlasso methods

After implementation of the program in appendix [4] using contaminated data, the following results were obtained:

Table (4-8) represents estimated values of parameters using proposed MM-lwlasso methods after contamination of data by 5%

B for MM-lwlasso with w^{R1}	B for MM-lwlasso with w^{R2}	B for MM-lwlasso with w^{R3}	B for MM-lasso	B for MM-alasso
0.000	0.000	0.000	-0.004	0.000
0.074	0.047	0.063	0.323	0.090
0.000	0.000	0.000	-0.035	0.000
0.000	0.000	0.000	-0.140	0.000
0.000	0.000	0.000	-0.002	0.000
0.000	0.000	0.000	0.131	0.000
0.000	0.000	0.000	-0.036	0.000
0.000	0.000	0.000	-0.055	0.000
0.000	0.000	0.000	-0.024	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.004	0.000
0.000	0.000	0.000	-0.057	0.000
0.000	0.000	0.000	-2.548	0.000
0.000	0.000	0.000	3.070	0.000
0.000	0.000	0.000	-1.186	0.000
0.000	0.000	0.000	-0.632	0.000
0.064	0.046	0.000	1.693	0.022
0.000	0.000	0.000	-0.030	0.000
0.000	0.000	0.000	-1.177	0.000
0.000	0.000	0.000	-1.092	0.000
0.000	0.000	0.000	1.738	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.632	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.540	0.000
0.000	0.000	0.000	-0.265	0.000
5.768	3.719	5.284	8.422	7.351
0.000	0.000	0.000	-0.458	0.000
0.000	0.000	0.000	-2.466	0.000
0.000	0.000	0.000	-0.111	0.000
0.000	0.000	0.000	1.033	0.000
0.000	0.000	0.000	0.395	0.000
0.000	0.000	0.000	-1.841	0.000
0.000	0.000	0.000	-0.617	0.000
0.000	0.000	0.000	0.401	0.000
0.000	0.000	0.000	1.382	0.000
0.000	0.000	0.000	0.286	0.000
0.000	0.000	0.000	0.066	0.000
0.000	0.000	0.000	-8.397	0.000
0.000	0.000	0.000	-1.078	0.000
0.000	0.000	0.000	-2.068	0.000
0.000	0.000	0.000	6.714	0.000
0.000	0.000	0.000	4.447	0.000
0.000	0.000	0.000	0.479	0.000
0.000	0.000	0.000	-7.049	0.000
0.000	0.000	0.000	6.503	0.000

0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.361	0.000
0.000	0.000	0.000	9.053	0.000
0.000	0.000	0.000	9.871	0.000
0.000	0.000	0.000	-6.702	0.000
-0.003	-0.002	-0.001	-0.012	-0.002
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.012	0.000
0.000	0.000	0.000	0.225	0.000
0.000	0.000	0.000	0.010	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.015	0.000
0.000	0.000	0.000	-0.077	0.000
0.000	0.000	0.000	0.008	0.000
0.000	0.000	0.000	-0.001	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.001	0.000
0.000	0.000	0.000	-0.115	0.000
0.000	0.000	0.000	0.493	0.000
0.000	0.000	0.000	0.541	0.000
0.000	0.000	0.000	-1.100	0.000
0.000	0.000	0.000	-0.057	0.000
0.000	0.000	0.000	-0.305	0.000
0.000	0.000	0.000	0.272	0.000
0.000	0.000	0.000	0.000	0.000
0.029	0.034	0.000	0.337	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.416	0.000
0.000	0.000	0.000	-0.445	0.000
0.000	0.000	0.000	0.117	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-0.083	0.000
0.000	0.000	0.000	0.036	0.000
0.000	0.000	0.000	-0.527	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.010	0.000
0.000	0.000	0.000	-0.004	0.000
0.000	0.000	0.000	-0.058	0.000
0.000	0.000	0.000	0.291	0.000
0.000	0.000	0.000	-0.013	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.108	0.000
0.000	0.000	0.000	-0.110	0.000

Noticing from the table that parameter values are stable for the proposed MM-lwlasso methods. Table (4-9) shows then the RPE^w criterion to compared the PLS methods and the penalized robust methods and the identification of the best methods.

Table (4-9) represents the RPE^w values of the PLS methods and the penalized robust methods (proposed MM-lwlasso methods) after adding outliers.

RPE	$\hat{w}^{(1)}$	$\hat{w}^{(2)}$	$\hat{w}^{(3)}$	Lasso	alasso	$\hat{w}^{R1}$	$\hat{w}^{R2}$	$\hat{w}^{R3}$	Mlasso	Malasso
	1.055	0.998	0.872	1.184	1.135	0.720	0.696	0.630	0.833	0.798

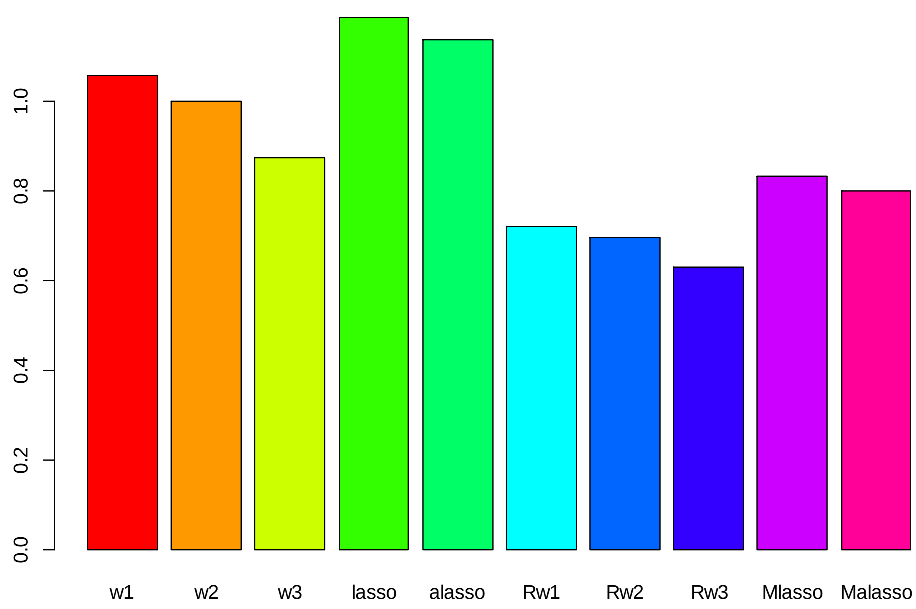

Figure (4-6) represents RPE^w of the PLS methods and the penalized robust methods (proposed MM-lwlasso methods) after adding outliers.

It was noted in Table (4-9) that the efficiency for PLS methods decreased by 5% in the data. The good results of the MM-Lwlasso (MM-lwlasso methods) were observed. The MM-lwlasso method with $\hat{w}^{R3}$ is the best with RPE^w value of 0.630 followed by the MM-lwlasso method with $\hat{w}^{R2}$ at 0.696 and then the MM-lwlasso method with $\hat{w}^{R1}$ at 0.720.

Chapter five

Conclusions

Recommendations

5-1 Conclusions

In this thesis, we stated “M-lag weighted lasso" and “MM-lag weighted lasso" methods to deal with contamination data. The suggested methods were demonstrated using a simulation study and a real data. The results show that the suggested methods are more stable than the other methods in comparison. Consequently, these suggested methods are able to deal with the outlier. In particular, the method " M-lag weighted lasso" and "MM-lag weighted lasso" with w^{R2} and w^{R3} weights gave the best results compared with the others.

5-2 Recommendations

The suggested methods can be extended to other methods such as robust lag-weighted elastic net, robust lag-weighted group lasso, robust lag-weighted fused lasso, robust lag-weighted graphical lasso, and so on. Furthermore, the proposed methods can be used not only for a fixed lag effect but also with varying lag effects across time.

References

References

1-Alfons, A., Croux, C., & Gelper, S. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 226-248.

2-Audrino, F., & Camponovo, L. (2013). Oracle properties and finite sample inference of the adaptive lasso for time series regression models. *arXiv preprint arXiv:1312.1473*.

3-Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, *5*(Sep), 1089-1105.

4-Box, G. E., & Jenkins, G. M. (1976). *Time series analysis: forecasting and control, revised ed*. Holden-Day.

5-Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2008). Forecasting. *Time Series Analysis*, 137-191.

6-Brockwell, P. J., Davis, R. A., & Calder, M. V. (2002). *Introduction to time series and forecasting* (Vol. 2). New York: springer.

7-Bruce, A. G., & Martin, R. D. (1989). Leave-k-out diagnostics for time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 363-424.

8-Chan, K., & Cryer, J. D. (2008). Time series analysis with applications in R.

9-Chang, I.H., Tiao, G.C. and C. Chen (1988). Estimation of Time Series Parameters in the Presence of Outliers. Technometrics, 30, 193-204.

10-Chatfield, C. (1995). *The analysis of time series: an introduction*. CRC press.

11-Chatfield, D. C., Kim, J. G., Harrison, T. P., & Hayya, J. C.(2004). The bullwhip effect—impact of stochastic lead time,

12-Cryer, J. D., & Chan, K. S. (2008). Time series regression models. *Time series analysis: with applications in R*, 249-276.

13-Darwish, K., & Buyuklu, A. H. (2015). Robust Linear Regression Using L1-Penalized MM-Estimation for High Dimensional Data. *American Journal of Theoretical and Applied Statistics*, *4*(3), 78-84.

14-De Jong, P., & Penzer, J. (1998). Diagnosing shocks in time series. *Journal of the American Statistical Association*, *93*(442), 796-806.

15-Fan, J., & Li, R. (2001). Variable selection via non concave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348-1360.

16-Flexeder, C. (2010). *Generalized lasso regularization for regression models* (Doctoral dissertation, Institut für Statistik).

17-Fox, A. J. (1972). Outliers in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 350-363.

18-Georgescu, V. (2015). Time series analysis. ISBN 978-606-11-4954-4.

19-Hesterberg, T., Choi, N. H., Meier, L., & Fraley, C. (2008). Least angle and ℓ1 penalized regression: A review. *Statistics Surveys*, *2*, 61-93.

20-Huber, P. J. Robust statistics. 1981.

information quality, and information sharing: a simulation study. *Production and Operations Management*, *13*(4), 340-353.

21-Jung, K. M. (2011). Weighted least absolute deviation LASSO estimator. *Communications for Statistical Applications and Methods*, *18*(6), 733-739.

22-Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). Applied linear statistical models.

23-Lambert-Lacroix, S., & Zwald, L. (2011).Robust regression through the Huber's criterion and adaptive lasso penalty. *Electronic Journal of Statistics*, *5*, 1015-1053.

24-Liu, Z. Z. (2014). The doubly adaptive LASSO methods for time series analysis.

25-Liu, Z. Z., Kulperger, R., & Yu, H. (2016). The Doubly Adaptive LASSO for Vector Autoregressive Models. In *Advances in Time Series Methods and Applications* (pp. 17-46). Springer, New York, NY.

26-Maronna, R. A. R. D., Martin, R. D., &Yohai, V. (2006). *Robust statistics* (Vol. 1). John Wiley & Sons, Chichester. ISBN.

27-Medeiros, M. C., & Mendes, E. F. (2012). *Estimating High-Dimensional Time Series Models* (No. 602). Texto para discussão.

28-Medeiros, M. C., & Mendes, E. F. (2015). *l1-estimation of high-dimensional time-series models with flexible innovations*. Working Paper.

29-Mylona, K., & Goos, P. (2010). Penalized Generalized Least Squares for Model Selection under Restricted Randomization.

30-Nardi, Y., &Rinaldo, A. (2011).Autoregressive process modeling via the lasso procedure. *Journal of Multivariate Analysis*, *102*(3), 528-549.

31-Park, H., & Sakaori, F. (2011, November). Modeling symbolic candle chart time series (Competition 2). In *Proceedings of the symposium of Japanese Society of Computational Statistics 25* (pp. 21-24). Japanese Society of Computational Statistics.

32-Park, H., Sakaori, F. (2013). Lag weighted lasso for time series model. Comp Stat, 28, 493-504.

33-Pesaran, M. H., & Shin, Y. (1998).An autoregressive distributed-lag modelling approach to cointegration analysis. *Econometric Society Monographs*, *31*, 371-413.

34-Rodriguez, J. D., Perez, A., & Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, *32*(3), 569-575.

35-Sartori, S. (2011). Penalized regression: Bootstrap confidence intervals and variable selection for high-dimensional data sets.

36-Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

37-Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, *7*(1), 1-20.

38-Walker, G. T. (1931). On periodicity in series of related terms. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *131*(818), 518-532.

39-Wang, H., Li, G., & Jiang, G. (2007).Robust regression shrinkage and consistent variable selection through the LAD-Lasso. *Journal of Business & Economic Statistics*, *25*(3), 347-355.

40-Wold, H. (1938). *A study in the analysis of stationary time series* (Doctoral dissertation, Almqvist & Wiksell).

41-Wu, Y., & Liu, Y. (2009).Variable selection in quantile regression. *Statistica Sinica*, 801-817.

42-Yi, C., & Huang, J. (2017). Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics*, *26*(3), 547-557.

43-Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67.

44-Yule, G. U. (1926). Why do we sometimes get nonsense-correlations between Time-Series?--a study in sampling and the nature of time-series. *Journal of the royal statistical society*, *89*(1), 1-63.

45-Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), 1418-1429.

Appendixes

Appendix[1]

```
library(lars)
q0=5
q1=3
q2=3
p=2
m=q0+q1+q2+p
gamma=0.2
sq=5
T<-200
Yt<-rnorm(n,0,sq)
y<-Yt
for(l in 1:q0){
ytl<-Yt[-c(1:l)]
y<-cbind(y,ytl)}
x1t<-rnorm(T,0,sq)
x1<-x1t
for(l in 1:q1){
x1tl<-x1t[-c(1:l)]
x1<-cbind(x1,x1tl)}
x2t<-rnorm(T,0,sq)
x2<-x2t
for(l in 1:q2){
x2tl<-x2t[-c(1:l)]
x2<-cbind(x2,x2tl)}
```

```
rpe.w1.i<-NULL

rpe.w2.i<-NULL

rpe.w3.i<-NULL

rpe.lasso.i<-NULL

rpe.adlasso.i<-NULL

rpe.w1.Robust.i<-NULL

rpe.w2.Robust.i<-NULL

rpe.w3.Robust.i<-NULL

rpe.lasso.Robust.i<-NULL

rpe.adlasso.Robust.i<-NULL

no.of.iteration<-1000

for (it in 1:no.of.iteration){

out<-0.05

BETA<-c(B01=0.3, B02=-0.2, B03=0.1, B04=0, B05=0, B10=0.9, B11=0.7,
B12=0.5, B13=0, B20=1, B21=-0.7, B22=0.5, B23=0)

et1<-rnorm((1-out)*n,0,sq)

#et0<-rnorm(out*T,0,1)

#et0<-rnorm(out*T,0,5)

#et0<-rnorm(out*T,0,10)

#et0<-rt(out*T,1)

et0<-rt(out*T,5)

et<-c(et1,et0)

et<-et[-c((T-max(q0,q1,q2)+1):n)]

data<-cbind(y,x1,x2)

data<-data[-c((T-max(q0,q1,q2)+1):n),]
```

```
yt<-data[,-1]%*%BETA+et
data1<-data[,-1]
#####################Alpha for the lag weight##################
alpha<-0.1
l_alpha<-1-alpha
################ LWLASSO method ###########################
############## bhat(ols) for the  weight######################
bhat<-t(abs(matrix(lm(yt~data[,-1])[[1]])[2:(ncol(data))]))
b.hat<-matrix(numeric(m*m),ncol=m,nrow=m)
diag(b.hat)<-bhat[,]
####################### wj,l type1 ###########################
w1<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
w1[l,l]<-1/((alpha*l_alpha^(l))^gamma)
for (l in 1:(q1+1)){
w1[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)
for (l in 1:(q2+1)){
w1[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}
########################## wj,l type2 ##############
w2<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
w2[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}
for (l in 1:(q1+1)){
w2[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l-1)*(abs(b.hat[l,l]))^gamma)}
for (l in 1:(q2+1)){
```

```
w2[(l+q0+q1+1),(l+q0+q1+1)]<-1/(alpha*l_alpha^(l-1)*(abs(b.hat[l,l]))^gamma)}
####################### wj,l type3 ###############
w3<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
w3[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}
for (l in 1:(q1+1)){
w3[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1)*abs(b.hat[l,l]))^gamma)}
for (l in 1:(q2+1)){
w3[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1)*abs(b.hat[l,l]))^gamma)}
####################### wj adlasso ###############
wad<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
wad[l,l]<-1/(abs(b.hat[l,l])^gamma)}
for (l in 1:(q1+1)){
wad[(l+q0),(l+q0)]<-1/(abs(b.hat[l,l])^gamma)}
for (l in 1:(q2+1)){
wad[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat[l,l])^gamma)}
ww1<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww1[l,l]<-1/w1[l,l]}
x.w1<-data1%*%ww1
##################
ww2<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww2[l,l]<-1/w2[l,l]}
```

```
x.w2<-data1%*%ww2
##################
ww3<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww3[l,l]<-1/w3[l,l]}
x.w3<-data1%*%ww3
##################
wwad<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
wwad[l,l]<-1/wad[l,l]}
x.wad<-data1%*%wwad
yt.bar<-mean(yt)
##################################################
lasso.w1<-(lars(x.w1,yt)$beta)
lambda<-ncol (lars(x.w1,yt)$ lambda)
B.w1<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w1[1:lambda,c]<-(diag(ww1))*(lasso.w1[1:lambda,c])}
error.w1<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w1[,c]<-(yt-t((B.w1%*%t(data1)))[,c])}
rpe.w1<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w1<-apply((error.w1^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))
##################################
lasso.w2<-(lars(x.w2,yt)$beta)
```

```
lambda<- lambda<-ncol (lars(x.w2,yt)$ lambda)
B.w2<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w2[1:lambda,c]<-(diag(ww2))*(lasso.w2[1:lambda,c])}
error.w2<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w2[,c]<-(yt-t((B.w2%*%t(data1)))[,c])}
rpe.w2<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w2<-apply((error.w2^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))
########################################
lasso.w3<-(lars(x.w3,yt)$beta)
lambda<-ncol (lars(x.w3,yt)$ lambda)
B.w3<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w3[1:lambda,c]<-(diag(ww3))*(lasso.w3[1:lambda,c])}
error.w3<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w3[,c]<-(yt-t((B.w3%*%t(data1)))[,c])}
rpe.w3<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w3<-apply((error.w3^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))
############################################
lasso<-(lars(data1,yt)$beta)
lambda<-ncol (lars(data1,yt)$ lambda)
B.lasso<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
```

```
B.lasso[1:lambda,c]<-(lasso[1:lambda,c])}
error.lasso<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.lasso[,c]<-(yt-t((B.lasso%*%t(data1)))[,c])}
rpe.lasso<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.lasso<-apply((error.lasso^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))
###########################################
adlasso<-(lars(x.wad,yt)$beta)
lambda<-ncol (lars(x.wad,yt)$ lambda)
B.wad<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.wad[1:lambda,c]<-(diag(wwad))*(adlasso[1:lambda,c])}
error.wad<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.wad[,c]<-(yt-t((B.wad%*%t(data1)))[,c])}
rpe.wad<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.wad<-apply((error.wad^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))
############################# ROBUST method ######################
# bhat(ols) for the  weight
rlmm<-rlm(yt~data[,-1],method="M")
w.huber<-rlmm$w
bhat.r<-t(abs(matrix(rlmm[[1]])[2:(ncol(data))]))
b.hat.r<-matrix(numeric(m*m),ncol=m,nrow=m)
diag(b.hat.r)<-bhat.r[,]
####################### (wj,l ) type1 ###########################
```

```
w1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w1.r[l,l]<-1/((alpha*l_alpha^(l))^gamma)}

for (l in 1:(q1+1)){

w1.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q2+1)){

w1.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}

############################(wj,l ) type2 ########################

w2.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w2.r[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q1+1)){

w2.r[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l-1)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w2.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(alpha*l_alpha^(l-
1)*(abs(b.hat.r[l,l]))^gamma)}

#########################( wj,l)  type3 ############

w3.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w3.r[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)

for (l in 1:(q1+1)){

w3.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w3.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-
1)*abs(b.hat.r[l,l]))^gamma)}
```

```
####################### wj adlasso ###############
wad.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
wad.r[l,l]<-1/(abs(b.hat.r[l,l])^gamma)}
for (l in 1:(q1+1)){
wad.r[(l+q0),(l+q0)]<-1/(abs(b.hat.r[l,l])^gamma)}
for (l in 1:(q2+1)){
wad.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat.r[l,l])^gamma)}
#############################################
ww1.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww1.r[l,l]<-1/w1.r[l,l]}
x.w1.r<-data1%*%ww1.r
###################
ww2.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww2.r[l,l]<-1/w2.r[l,l]}
x.w2.r<-data1%*%ww2.r
###################
ww3.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww3.r[l,l]<-1/w3.r[l,l]}
x.w3.r<-data1%*%ww3.r
###################
wwad.r<-matrix(numeric(m*m),ncol=m,nrow=m)
```

```
for (l in 1:m){
wwad.r[l,l]<-1/wad.r[l,l]}
x.wad.r<-data1%*%wwad.r
################################################
yt.bar.r<-mean(w.huber*yt)
################################################
rlasso.w1<-(lars(x.w1.r,(w.huber*yt))$beta)
lambda<- lambda<-ncol (lars(x.w1.r,(w.huber*yt)$ lambda)
B.w1.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w1.r[1:lambda,c]<-(diag(ww1.r))*(rlasso.w1[1:lambda,c])}
error.w1.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w1.r[,c]<-((w.huber*yt)-t((B.w1.r%*%t(data1)))[,c])}
rpe.w1.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w1.r<-apply((error.w1.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
#################################
rlasso.w2<-(lars(x.w2.r,(w.huber*yt))$beta)
lambda<- lambda<-ncol (lars(x.w2.r,(w.huber*yt)$ lambda)
B.w2.r<-matrix(numeric(lambda*90),ncol=m,nrow=lambda)
for (c in 1:m){
B.w2.r[1:lambda,c]<-(diag(ww2.r))*(rlasso.w2[1:lambda,c])}
error.w2.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w2.r[,c]<-((w.huber*yt)-t((B.w2.r%*%t(data1)))[,c])}
```

```
rpe.w2.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w2.r<-apply((error.w2.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
########################################
rlasso.w3<-(lars(x.w3.r,(w.huber*yt))$beta)
lambda<-ncol (lars(x.w3.r,(w.huber*yt)$ lambda)
B.w3.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w3.r[1:lambda,c]<-(diag(ww3.r))*(rlasso.w3[1:lambda,c])}
error.w3.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w3.r[,c]<-((w.huber*yt)-t((B.w3.r%*%t(data1)))[,c])}
rpe.w3.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w3.r<-apply((error.w3.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
############################################
rlasso<-(lars(data1,(w.huber*yt))$beta)
lambda<-ncol (lars(data1,(w.huber*yt)$ lambda)
B.lasso.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.lasso.r[1:lambda,c]<-(rlasso[1:lambda,c])}
error.lasso.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.lasso.r[,c]<-((w.huber*yt)-t((B.lasso.r%*%t(data1)))[,c])}
rpe.lasso.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.lasso.r<-apply((error.lasso.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
```

```
############################################
rlasso.wad<-(lars(x.wad.r,(w.huber*yt))$beta)
B.wad.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.wad.r[1:lambda,c]<-(diag(wwad.r))*(rlasso.wad[1:lambda,c])}
error.wad.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.wad.r[,c]<-((w.huber*yt)-t((B.wad.r%*%t(data1)))[,c])}
rpe.wad.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.wad.r<-apply((error.wad.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
###########################################
rpe..w1<-rpe.w1
rpe..w2<-rpe.w2
rpe..w3<-rpe.w3
rpe..lasso<-rpe.lasso
rpe..adlasso<-rpe.wad
rpe.w1.Robust<-rpe.w1.r
rpe.w2.Robust<- rpe.w2.r
rpe.w3.Robust<-rpe.w3.r
rpe.lasso.Robust<-rpe.lasso.r
rpe.adlasso.Robust<-rpe.wad.r
########################################
rpe.w1.i<-cbind(rpe.w1.i,rpe..w1)
rpe.w2.i<-cbind(rpe.w2.i,rpe..w2)
rpe.w3.i<-cbind(rpe.w3.i,rpe..w3)
```

```
rpe.lasso.i<-cbind(rpe.lasso.i,rpe.lasso)

rpe.adlasso.i<-cbind(rpe.adlasso.i,rpe..adlasso)

rpe.w1.Robust.i<-cbind(rpe.w1.Robust.i,rpe.w1.Robust)

rpe.w2.Robust.i<-cbind(rpe.w2.Robust.i,rpe.w2.Robust)

rpe.w3.Robust.i<-cbind(rpe.w3.Robust.i,rpe.w3.Robust)

rpe.lasso.Robust.i<-cbind(rpe.lasso.Robust.i,rpe.lasso.Robust)

rpe.adlasso.Robust.i<-cbind(rpe.adlasso.Robust.i,rpe.adlasso.Robust)

pb <- txtProgressBar(min = 0, max = no.of.iteration, style = 3)

setTxtProgressBar(pb, it)}

rpew1<-median(rpe.w1.i)

rpew2<-median(rpe.w2.i)

rpew3<-median(rpe.w3.i)

rpelasso<-median(rpe.lasso.i)

rpeadlasso<-median(rpe.adlasso.i)

rpew1.Robust<-median(rpe.w1.Robust.i)

rpew2.Robust<-median(rpe.w2.Robust.i)

rpew3.Robust<-median(rpe.w3.Robust.i)

rpelasso.Robust<-median(rpe.lasso.Robust.i)

rpeadlasso.Robust<-median(rpe.adlasso.Robust.i)

rpe<-c(rpew1,rpew2,rpew3,rpelasso,rpeadlasso,rpew1.Robust,

rpew2.Robust,rpew3.Robust,rpelasso.Robust,rpeadlasso.Robust)

rpe<-t(as.matrix(rpe))

colnames(rpe)<-c("w1","w2","w3","lasso","adlasso","Robust w1","Robust
w2","Robust w3","Robust lasso","Robust adlasso")

rpe
```

##################################

Appendix[2]

```
############################ ROBUST method ########################
# bhat(ols) for the  weight
rlmm<-rlm(yt~data[,-1],method="MM")
w.huber<-rlmm$w
bhat.r<-t(abs(matrix(rlmm[[1]])[2:(ncol(data))]))
b.hat.r<-matrix(numeric(m*m),ncol=m,nrow=m)
diag(b.hat.r)<-bhat.r[,]
######################## (wj,l ) type1 ############################
w1.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
w1.r[l,l]<-1/((alpha*l_alpha^(l))^gamma)}
for (l in 1:(q1+1)){
w1.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)}
for (l in 1:(q2+1)){
w1.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}
############################(wj,l ) type2 ########################
w2.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:q0){
w2.r[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}
for (l in 1:(q1+1)){
w2.r[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l-1)*(abs(b.hat.r[l,l]))^gamma)}
for (l in 1:(q2+1)){
```

```
w2.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(alpha*l_alpha^(l-
1)*(abs(b.hat.r[l,l]))^gamma)}

########################( wj,l)  type3 ############

w3.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w3.r[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)

for (l in 1:(q1+1)){

w3.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w3.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-
1)*abs(b.hat.r[l,l]))^gamma)}

###################### wj adlasso ##############

wad.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

wad.r[l,l]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q1+1)){

wad.r[(l+q0),(l+q0)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q2+1)){

wad.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat.r[l,l])^gamma)}

ww1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

ww1.r[l,l]<-1/w1.r[l,l]}

x.w1.r<-data1%*%ww1.r

##################

ww2.r<-matrix(numeric(m*m),ncol=m,nrow=m)
```

```
for (l in 1:m){
ww2.r[l,l]<-1/w2.r[l,l]}
x.w2.r<-data1%*%ww2.r
##################
ww3.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww3.r[l,l]<-1/w3.r[l,l]}
x.w3.r<-data1%*%ww3.r
##################
wwad.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
wwad.r[l,l]<-1/wad.r[l,l]}
x.wad.r<-data1%*%wwad.r
##################################################
yt.bar.r<-mean(w.huber*yt)
##################################################
rlasso.w1<-(lars(x.w1.r,(w.huber*yt))$beta)
lambda<- lambda<-ncol (lars(x.w1.r,(w.huber*yt)$ lambda)
B.w1.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w1.r[1:lambda,c]<-(diag(ww1.r))*(rlasso.w1[1:lambda,c])}
error.w1.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w1.r[,c]<-((w.huber*yt)-t((B.w1.r%*%t(data1)))[,c])}
rpe.w1.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
```

```
rpe.w1.r<-apply((error.w1.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
###################################
rlasso.w2<-(lars(x.w2.r,(w.huber*yt))$beta)
lambda<- lambda<-ncol (lars(x.w2.r,(w.huber*yt)$ lambda)
B.w2.r<-matrix(numeric(lambda*90),ncol=m,nrow=lambda)
for (c in 1:m){
B.w2.r[1:lambda,c]<-(diag(ww2.r))*(rlasso.w2[1:lambda,c])}
error.w2.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w2.r[,c]<-((w.huber*yt)-t((B.w2.r%*%t(data1)))[,c])}
rpe.w2.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w2.r<-apply((error.w2.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
#########################################
rlasso.w3<-(lars(x.w3.r,(w.huber*yt))$beta)
lambda<- lambda<-ncol (lars(x.w3.r,(w.huber*yt)$ lambda)
B.w3.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w3.r[1:lambda,c]<-(diag(ww3.r))*(rlasso.w3[1:lambda,c])}
error.w3.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))
for (c in 1:lambda){
error.w3.r[,c]<-((w.huber*yt)-t((B.w3.r%*%t(data1)))[,c])}
rpe.w3.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))
rpe.w3.r<-apply((error.w3.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
#############################################
rlasso<-(lars(data1,(w.huber*yt))$beta)
```

```
lambda<- lambda<-ncol (lars(data1,(w.huber*yt)$ lambda)

B.lasso.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.lasso.r[1:lambda,c]<-(rlasso[1:lambda,c])}

error.lasso.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))

for (c in 1:lambda){

error.lasso.r[,c]<-((w.huber*yt)-t((B.lasso.r%*%t(data1)))[,c])}

rpe.lasso.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))

rpe.lasso.r<-apply((error.lasso.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-
1))

###########################################

rlasso.wad<-(lars(x.wad.r,(w.huber*yt))$beta)

B.wad.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.wad.r[1:lambda,c]<-(diag(wwad.r))*(rlasso.wad[1:lambda,c])}

error.wad.r<-matrix(numeric(lambda*nrow(yt)),ncol=lambda,nrow=nrow(yt))

for (c in 1:lambda){

error.wad.r[,c]<-((w.huber*yt)-t((B.wad.r%*%t(data1)))[,c])}

rpe.wad.r<-matrix(numeric(1*nrow(yt)),ncol=1,nrow=nrow(yt))

rpe.wad.r<-apply((error.wad.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))

##########################################

rpe..w1<-rpe.w1

rpe..w2<-rpe.w2

rpe..w3<-rpe.w3

rpe..lasso<-rpe.lasso
```

```
rpe..adlasso<-rpe.wad
rpe.w1.Robust<-rpe.w1.r
rpe.w2.Robust<- rpe.w2.r
rpe.w3.Robust<-rpe.w3.r
rpe.lasso.Robust<-rpe.lasso.r
rpe.adlasso.Robust<-rpe.wad.r
########################################
rpe.w1.i<-cbind(rpe.w1.i,rpe..w1)
rpe.w2.i<-cbind(rpe.w2.i,rpe..w2)
rpe.w3.i<-cbind(rpe.w3.i,rpe..w3)
rpe.lasso.i<-cbind(rpe.lasso.i,rpe.lasso)
rpe.adlasso.i<-cbind(rpe.adlasso.i,rpe..adlasso)
rpe.w1.Robust.i<-cbind(rpe.w1.Robust.i,rpe.w1.Robust)
rpe.w2.Robust.i<-cbind(rpe.w2.Robust.i,rpe.w2.Robust)
rpe.w3.Robust.i<-cbind(rpe.w3.Robust.i,rpe.w3.Robust)
rpe.lasso.Robust.i<-cbind(rpe.lasso.Robust.i,rpe.lasso.Robust)
rpe.adlasso.Robust.i<-cbind(rpe.adlasso.Robust.i,rpe.adlasso.Robust)
pb <- txtProgressBar(min = 0, max = no.of.iteration, style = 3)
   setTxtProgressBar(pb, it)}
rpew1<-median(rpe.w1.i)
rpew2<-median(rpe.w2.i)
rpew3<-median(rpe.w3.i)
rpelasso<-median(rpe.lasso.i)
rpeadlasso<-median(rpe.adlasso.i)
rpew1.Robust<-median(rpe.w1.Robust.i)
```

```
rpew2.Robust<-median(rpe.w2.Robust.i)

rpew3.Robust<-median(rpe.w3.Robust.i)

rpelasso.Robust<-median(rpe.lasso.Robust.i)

rpeadlasso.Robust<-median(rpe.adlasso.Robust.i)

rpe<-c(rpew1,rpew2,rpew3,rpelasso,rpeadlasso,rpew1.Robust,

rpew2.Robust,rpew3.Robust,rpelasso.Robust,rpeadlasso.Robust)

rpe<-t(as.matrix(rpe))

colnames(rpe)<-c("w1","w2","w3","lasso","adlasso","Robust w1","Robust
w2","Robust w3","Robust lasso","Robust adlasso")
```

Appendix[3]

```
q0=lagy

q1=lagx1

q2=lagx2

q3=lagx3

q4=lagx4

q5=lagx5

q6=lagx6

p=6

m=q0+q1+q2+q3+q4+q5+q6+p

gamma=0.2

T<-138

yt<-
c(0,3,2,1,1,5,200,400,5,1,2,2,0,0,0,0,0,1,0,4,1,4,0,0,200,1,400,4,2,2,2,1,5,3,0,4,1,1,
2,3,2,1,6,1,4,4,4,1,3,2,0,4,4,1,2,2,1,1,3,4,1,2,0,4,3,1,2,3,3,3,1,3,0,1,6,0,1,2,0,0,2,10
0,0,0,80,1,1,5,1,0,2,1,5,0,1,5,1,0,4,111,0,1,2,1,5,5,2,3,2,4,3,2,1,3,2,2,3,0,1,1,0,1,0,
3,1,1,1,4,5,80,0,4,3,3,3,1,6,1)

y<-yt
```

```
for(l in 1:q0){

ytl<-yt[-c(1:l)]

y<-cbind(y,ytl)}

x1t<-c(20,22,23,29,28,22,20,22,22,12,13,15,18,20,22,28,29,27,23,19,15,13,12,15,

18,25,28,28,30,25,22,18,13,11,12,13,15,20,23,26,35,25,18,15,13,12,14,20,21,21,2
2,28,33,28,22,18,17,13,15,18,17,23,25,27,30,24,18,17,13,15,18,13,18,20,25,28,31,
28,25,18,15,13,14,15,17,19,20,28,31,28,23,21,19,12,13,15,21,23,29,29,30,27,26,2
1,14,12,14,19,24,26,29,32,32,28,27,19,16,13,14,16,21,23,27,29,31,27,22,18,14,11,
12,14,19,22,24,28,32,31)

x1<-x1t

for(l in 1:q1){

x1tl<-x1t[-c(1:l)]

x1<-cbind(x1,x1tl)}

x2t<-c(4,2.9,4.12,2.48,2.39,1,7.47,4.6,40.7,4.7,5,2.6,3.06,4.3,1,2.5,1.2,2.2,4.8,3.9,

5,1,4.3,3.51,4.53,9.72,5,8.68,3.42,6.75,4.8,7.21,1.96,0.6,0.6,1.22,4.1,5,0.3,3.23,1.4
5,7.9,4.81,13.73,0.2,2.1,2.82,1.38,3.2,2.2,5,4.4,5,4.9,4.09,5,2.6,8.5,4.5,2.8,3.42,2.3
4,4.5,3.5,4.1,1,5.32,2.4,7.5,7.2,7.4,1.45,3.7,2.5,2.24,1.7,3,5.22,4.82,1.52,6.31,1.96,
7.67,6.32,3.68,3.48,1.9,3.04,3.85,4.9,2.86,0.13,2.22,2.54,1,3.52,2.71,3.52,2.95,4.3,
1,5.94,2.7,1.82,4.33,5.45,1.07,8.61,6.5,3.77,5.2,1,2.53,6.75,3.4,13,13.24,3.3,2.2,1.
9,1.4,3.2,2,2.53,1.5,1.2,4.1,8.7,1.2,6.44,9.32,2.2,1.49,1.3,14.8,2,4.5,10.9)

x2<-x2t

for(l in 1:q2){

x2tl<-x2t[-c(1:l)]

x2<-cbind(x2,x2tl)}

x3t<-c(8,8,7.9,8,8.1,8.2,8.5,7.8,7.9,8,8,8.3,8.5,8,8.4,8,7.8,8.3,8.2,8.1,8.2,7.6,8.1,

8.2,8.3,8.4,8.4,8.5,8.3,8.4,8.3,8.4,8.2,7.5,7.4,7.1,7.3,7.4,7,7.2,7,7.2,7.1,7.5,7.8,7.4,
7.6,8.2,7.8,7.9,8.1,7.9,7,8,8.5,6.9,7.3,8,8,7.9,8,8.1,8,7.9,8,8,8.1,7.6,8.1,7.9,7.8,8.1,
7.2,8.6,7.6,8.1,8.1,8,7.8,7.9,8,7.4,7.4,7,7.5,7.6,7.7,7.8,8,7.7,7.2,7.7,7.5,7.6,7.4,8.4,
8.2,8.1,8.4,8.3,8.4,8.3,8.4,8.1,8.3,7.3,7.8,7.6,7.6,7.7,7.7,7.9,8.2,8,7.8,7.6,7.6,7.2,7.
2,7.4,7.1,7.2,7.4,7.2,7.4,7,7.3,7.9,8,6.7,7,7.1,7.5,7.4,8.1,7.2,7.1,7.4)
```

```
x3<-x3t

for(l in 1:q3){

x3tl<-x3t[-c(1:l)]

x3<-cbind(x3,x3tl)}

x4t<-c(1007,11m,1220,1200,1328,1339,1321,1102,1095,950,996,978,1123,1107,

1132,1105,1101,1024,1020,1018,1032,1005,987,1069,1133,1120,1048,1044,1091,
1080,1100,1092,1160,1128,1140,1165,1110,1064,1124,1158,1225,1225,2350,232
0,1128,1257,1305,1350,1082,1204,1194,1204,1006,1002,1156,930,1080,965,1002
,1045,1083,1150,1002,1037,1019,1050,1149,1156,1105,1015,1016,1085,1253,103
6,1033,1335,1163,1156,1129,1073,1062,1539,1349,1355,1286,1293,1256,1238,12
58,1250,1236,1233,1463,1823,1823,1105,1053,1060,1140,1149,1375,1380,1073,1
133,1230,1351,1186,1309,1310,1235,1196,1587,1534,1583,1532,1281,1355,1027,
1007,998,983,994,993,976,983,1105,895,981,952,1005,1000,982,986,1058,957,31
032,1008,1042)

x4<-x4t

for(l in 1:q4){

x4tl<-x4t[-c(1:l)]

x4<-cbind(x4,x4tl)}

x5t<-c(124,144,148,150,154,156,132,150,148,120,124,122,138,138,140,140,136,

128,124,125,130,,124,124,134,142,138,132,128,136,132,138,134,138,120,122,116
,108,122,112,118,124,124,168,172,120,146,150,154,136,142,148,150,124,124,142
,118,134,122,124,132,136,144,132,130,128,130,144,146,130,128,124,136,152,130
,130,154,140,138,138,136,132,130,140,138,124,98,122,124,124,128,124,132,124,
156,154,138,130,134,144,142,166,168,136,142,146,122,124,124,124,130,134,
124,128,132,134,138,128,130,124,130,122,124,124,122,124,140,122,124,122,124,
128,124,122,132,150,128,124,130)

x5<-x5t

for(l in 1:q5){

x5tl<-x5t[-c(1:l)]

x5<-cbind(x5,x5tl)}
```

```
x6t<-c(330,3m,399,392,434,439,431,392,383,331,326,322,393,367,373,362,359,

342,338,332,340,332,323,350,371,380,344,354,359,366,361,380,381,340,344,384,
364,340,362,384,406,396,850,832,370,428,428,440,356,391,3m,392,329,327,369,
310,355,320,333,346,359,383,333,339,335,345,379,380,363,336,338,361,415,341,
342,436,385,379,373,352,360,518,460,456,448,455,435,420,423,431,421,425,4m,
592,598,370,343,338,371,365,487,498,353,371,391,465,456,436,430,498,413,530,
513,523,504,437,460,335,346,328,332,360,340,325,353,335,339,323,318,332,326,
328,324,378,313,338,342,358)

x6<-x6t

for(l in 1:q6){

x6tl<-x6t[-c(1:l)]

x6<-cbind(x6,x6tl)}

data<-cbind(y,x1,x2,x3,x4,x5,x6)

data<-data[-c((T-max(q0,q1,q2,q3,q4,q5,q6)+1):T),]

yt<-data[,1]

data1<-data[,-1]

yt.bar<-mean(yt)

########################Alpha for the lag weight ####################

alpha<-0.1

l_alpha<-1-alpha

######################## LWLASSO method ######################

####################### bhat(ols) for the  weight######################

bhat<-t(abs(matrix(lm(yt~data1)[[1]])[2:(ncol(data))]))

b.hat<-matrix(numeric(m*m),ncol=m,nrow=m)

diag(b.hat)<-bhat[,]

####################### (wj,l )type1 ###########################

w1<-matrix(numeric(m*m),ncol=m,nrow=m)
```

```
for (l in 1:q0){

w1[l,l]<-1/((alpha*l_alpha^(l))^gamma)}

for (l in 1:(q1+1)){

w1[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q2+1)){

w1[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q3+1)){

w1[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q4+1)){

w1[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q5+1)){

w1[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-1/((alpha*l_alpha^(l-
1))^gamma)}

for (l in 1:(q6+1)){

w1[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l-1))^gamma)}

######################## (wj,l ) type2 ############################

w2<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w2[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q1+1)){

w2[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q2+1)){

w2[(l+q0+q1+1),(l+q0+q1+1)]<-1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q3+1)){
```

```
w2[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q4+1)){

w2[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q5+1)){

w2[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

for (l in 1:(q6+1)){

w2[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat[l,l]))^gamma)}

########################## (wj,l ) type3 ##############

w3<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w3[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q1+1)){

w3[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q2+1)){

w3[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q3+1)){

w3[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q4+1)){

w3[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q5+1)){
```

```
w3[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

for (l in 1:(q6+1)){

w3[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l)*abs(b.hat[l,l]))^gamma)}

###################### w  adlasso ########

wad<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

wad[l,l]<-1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q1+1)){

wad[(l+q0),(l+q0)]<-1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q2+1)){

wad[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q3+1)){

wad[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q4+1)){

wad[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q5+1)){

wad[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(abs(b.hat[l,l])^gamma)}

for (l in 1:(q6+1)){

wad[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(abs(b.hat[l,l])^gamma)}

###############

ww1<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){
```

```
ww1[l,l]<-1/w1[l,l]
#ifelse(ww1[l,l]==Inf,(ww1[l,l]<-0),(ww1[l,l]<-ww1[l,l]))}
x.w1<-data1%*%ww1
################
ww2<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww2[l,l]<-1/w2[l,l]
#ifelse(ww2[l,l]==Inf,(ww2[l,l]<-0),(ww2[l,l]<-ww2[l,l]))}
x.w2<-data1%*%ww2
###############
ww3<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww3[l,l]<-1/w3[l,l]
#ifelse(ww3[l,l]==Inf,(ww3[l,l]<-0),(ww3[l,l]<-ww3[l,l]))}
x.w3<-data1%*%ww3
################
wwad<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
wwad[l,l]<-1/wad[l,l]
#ifelse(wwad[l,l]==Inf,(wwad[l,l]<-0),(wwad[l,l]<-wwad[l,l]))}
x.wad<-data1%*%wwad
################################################
lasso.w1<-(lars(x.w1,yt)$beta)
lambda<- ncol(lars(x.w1,yt)$ lambda)
B.w1<-matrix(numeric(lambda *m),ncol=m,nrow= lambda)
```

```
for (c in 1:m){

B.w1[1:lambda,c]<-(diag(ww1))*(lasso.w1[1: lambda,c])}

error.w1<-matrix(numeric(lambda *(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=
lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1: lambda){

error.w1[,c]<-(yt-(data1%*%t(B.w1)[,c]))}

rpe.w1<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w1<-apply((error.w1^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))

###################################

lasso.w2<-(lars(x.w2,yt)$beta)

B.w2<-matrix(numeric(lambda *m),ncol=m,nrow= lambda)

for (c in 1:m){

B.w2[1: lambda,c]<-(diag(ww2))*(lasso.w2[1: lambda,c])}

error.w2<-matrix(numeric(lambda *(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=
lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1: lambda){

error.w2[,c]<-(yt-(data1%*%t(B.w2)[,c]))}

rpe.w2<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w2<-apply((error.w2^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))

#########################################

lasso.w3<-(lars(x.w3,yt)$beta)

B.w3<-matrix(numeric(lambda *m),ncol=m,nrow= lambda)

for (c in 1:m){

B.w3[1: lambda,c]<-(diag(ww3))*(lasso.w3[1: lambda,c])}
```

```
error.w3<-matrix(numeric(lambda *(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=
lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1: lambda){

error.w3[,c]<-(yt-(data1%*%t(B.w3)[,c]))}

rpe.w3<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w3<-apply((error.w3^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))

############################################

lasso<-(lars(data1,yt)$beta)

B.lasso<-matrix(numeric(lambda *m),ncol=m,nrow= lambda)

for (c in 1:m){

B.lasso[1: lambda,c]<-(lasso[1: lambda,c])}

error.lasso<-matrix(numeric(lambda *(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=
lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1: lambda){

error.lasso[,c]<-(yt-(data1%*%t(B.lasso)[,c]))}

rpe.lasso<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.lasso<-apply((error.lasso^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))

###########################################

adlasso<-(lars(x.wad,yt)$beta)

B.wad<-matrix(numeric(lambda *m),ncol=m,nrow= lambda)

for (c in 1:m){

B.wad[1:lambda,c]<-(diag(wwad))*(adlasso[1: lambda,c])}

error.wad<-matrix(numeric(lambda *(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=
lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1: lambda){
```

```
error.wad[,c]<-(yt-(data1%*%t(B.wad)[,c]))}

rpe.wad<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.wad<-apply((error.wad^2),2,mean)/((sum((yt-yt.bar)^2))/(T-1))

##################################################################

############################ ROBUST method #################

# bhat(ols) for the  weight

rlmm<-rlm(yt~data[,-1],method="M")

w.huber<-rlmm$w

bhat.r<-t(abs(matrix(rlmm[[1]])[2:(ncol(data))]))

b.hat.r<-matrix(numeric(m*m),ncol=m,nrow=m)

diag(b.hat.r)<-bhat.r[,]

####################### (wj,l ) type1 ########################

w1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w1.r[l,l]<-1/((alpha*l_alpha^(l))^gamma)}

for (l in 1:(q1+1)){

w1.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q2+1)){

w1.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q3+1)){

w1.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q4+1)){

w1.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/((alpha*l_alpha^(l-
1))^gamma)}
```

```
for (l in 1:(q5+1)){

w1.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-1/((alpha*l_alpha^(l-
1))^gamma)}

for (l in 1:(q6+1)){

w1.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l-1))^gamma)}

####################### (wj,l ) type2 ###########################

w2.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w2.r[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q1+1)){

w2.r[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w2.r[(l+q0+q1+1),(l+q0+q1+1)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q3+1)){

w2.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q4+1)){

w2.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q5+1)){

w2.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q6+1)){

w2.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}
```

```
########################### (wj,l ) type3 #############################

w3.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w3.r[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q1+1)){

w3.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w3.r[(l+q0+q1+1),(l+q0+q1+1)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q3+1)){

w3.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q4+1)){

w3.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q5+1)){

w3.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q6+1)){

w3.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

####################### w adlasso ########

wad.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

wad.r[l,l]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q1+1)){

wad.r[(l+q0),(l+q0)]<-1/(abs(b.hat.r[l,l])^gamma)}
```

```
for (l in 1:(q2+1)){

wad.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q3+1)){

wad.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q4+1)){

wad.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q5+1)){

wad.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q6+1)){

wad.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(abs(b.hat.r[l,l])^gamma)}

##########################################3

ww1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

ww1.r[l,l]<-1/w1.r[l,l]

#ifelse(ww1.r[l,l]==Inf,(ww1.r[l,l]<-0),(ww1.r[l,l]<-ww1.r[l,l]))}

x.w1.r<-data1%*%ww1.r

##################

ww2.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

ww2.r[l,l]<-1/w2.r[l,l]

#ifelse(ww2.r[l,l]==Inf,(ww2.r[l,l]<-0),(ww2.r[l,l]<-ww2.r[l,l]))}

x.w2.r<-data1%*%ww2.r

##################
```

```
ww3.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

ww3.r[l,l]<-1/w3.r[l,l]

#ifelse(ww3.r[l,l]==Inf,(ww3.r[l,l]<-0),(ww3.r[l,l]<-ww3.r[l,l]))}

x.w3.r<-data1%*%ww3.r

##################

wwad.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

wwad.r[l,l]<-1/wad.r[l,l]

#ifelse(wwad.r[l,l]==Inf,(wwad.r[l,l]<-0),(wwad.r[l,l]<-wwad.r[l,l]))}

x.wad.r<-data1%*%wwad.r

yt.bar.r<-mean(w.huber*yt)

##################################################

rlasso.w1<-(lars(x.w1.r,(w.huber*yt))$beta)

lambda<-114 lars(x.w1.r,(w.huber*yt))$ lambda

B.w1.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.w1.r[1:lambda,c]<-(diag(ww1.r))*(rlasso.w1[1:lambda,c])}

error.w1.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.w1.r[,c]<-((w.huber*yt)-(data1%*%t(B.w1.r)[,c]))}

rpe.w1.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w1.r<-apply((error.w1.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(n-1))
```

```
#####################################
rlasso.w2<-(lars(x.w2.r,(w.huber*yt))$beta)
B.w2.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w2.r[1:lambda,c]<-(diag(ww2.r))*(rlasso.w2[1:lambda,c])}
error.w2.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))
for (c in 1:lambda){
error.w2.r[,c]<-((w.huber*yt)-(data1%*%t(B.w2.r)[,c]))}
rpe.w2.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))
rpe.w2.r<-apply((error.w2.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
###########################################
rlasso.w3<-(lars(x.w3.r,(w.huber*yt))$beta)
B.w3.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w3.r[1:lambda,c]<-(diag(ww3.r))*(rlasso.w3[1:lambda,c])}
error.w3.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))
for (c in 1:lambda){
error.w3.r[,c]<-((w.huber*yt)-(data1%*%t(B.w3.r)[,c]))}
rpe.w3.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))
rpe.w3.r<-apply((error.w3.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
###############################################
rlasso<-(lars(data1,(w.huber*yt))$beta)
```

```
B.lasso.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.lasso.r[1:lambda,c]<-(rlasso[1:lambda,c])}

error.lasso.r<-matrix(numeric(lambda*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.lasso.r[,c]<-((w.huber*yt)-(data1%*%t(B.lasso.r)[,c]))}

rpe.lasso.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

rpe.lasso.r<-apply((error.lasso.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))

###########################################

rlasso.wad<-(lars(x.wad.r,(w.huber*yt))$beta)

B.wad.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.wad.r[1:lambda,c]<-(diag(wwad.r))*(rlasso.wad[1:lambda,c])}

error.wad.r<-matrix(numeric(lambda*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.wad.r[,c]<-((w.huber*yt)-(data1%*%t(B.wad.r)[,c]))}

rpe.wad.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

rpe.wad.r<-apply((error.wad.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))

#########################################

rpe<-c(min(rpe.w1),min(rpe.w2),min(rpe.w3),min(rpe.lasso),min(rpe.wad),min(rpe.w1.),min(rpe.w2.r),min(rpe.w3.r),min(rpe.lasso.r),min(rpe.wad.r))
```

```
rpe<-t(as.matrix(rpe))

colnames(rpe)<-c("w1","w2","w3","lasso","adlasso","Robust w1","Robust w2","Robust w3","Robust lasso","Robust adlasso")

r<-as.vector(rpe)

##############################

barplot(r,col = rainbow(10),names.arg=c("w1","w2","w3","lasso","alasso","Rw1",

"Rw2","Rw3","Mlasso","Malasso"))

plot(yt,type="b",col=1)
```

Appendix[4]

```
############################## ROBUST method #################

# bhat(ols) for the  weight

set.seed(1)

rlmm<-rlm(yt~data[,-1],method="MM")

w.huber<-rlmm$w

bhat.r<-t(abs(matrix(rlmm[[1]])[2:(ncol(data))]))

b.hat.r<-matrix(numeric(m*m),ncol=m,nrow=m)

diag(b.hat.r)<-bhat.r[,]

####################### (wj,l ) type1 #######################

w1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w1.r[l,l]<-1/((alpha*l_alpha^(l))^gamma)}

for (l in 1:(q1+1)){

w1.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q2+1)){

w1.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/((alpha*l_alpha^(l-1))^gamma)}
```

```
for (l in 1:(q3+1)){

w1.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/((alpha*l_alpha^(l-1))^gamma)}

for (l in 1:(q4+1)){

w1.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/((alpha*l_alpha^(l-
1))^gamma)}

for (l in 1:(q5+1)){

w1.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-1/((alpha*l_alpha^(l-
1))^gamma)}

for (l in 1:(q6+1)){

w1.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l-1))^gamma)}

####################### (wj,l ) type2 ###########################

w2.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w2.r[l,l]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q1+1)){

w2.r[(l+q0),(l+q0)]<-1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w2.r[(l+q0+q1+1),(l+q0+q1+1)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q3+1)){

w2.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q4+1)){

w2.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q5+1)){
```

```
w2.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q6+1)){

w2.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(alpha*l_alpha^(l)*(abs(b.hat.r[l,l]))^gamma)}

########################## (wj,l ) type3 ############################

w3.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

w3.r[l,l]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q1+1)){

w3.r[(l+q0),(l+q0)]<-1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q2+1)){

w3.r[(l+q0+q1+1),(l+q0+q1+1)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q3+1)){

w3.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q4+1)){

w3.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q5+1)){

w3.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

for (l in 1:(q6+1)){

w3.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/((alpha*l_alpha^(l)*abs(b.hat.r[l,l]))^gamma)}

###################### w adlasso ########
```

```
wad.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:q0){

wad.r[l,l]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q1+1)){

wad.r[(l+q0),(l+q0)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q2+1)){

wad.r[(l+q0+q1+1),(l+q0+q1+1)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q3+1)){

wad.r[(l+q0+q1+q2+2),(l+q0+q1+q2+2)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q4+1)){

wad.r[(l+q0+q1+q2+q3+3),(l+q0+q1+q2+q3+3)]<-1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q5+1)){

wad.r[(l+q0+q1+q2+q3+q4+4),(l+q0+q1+q2+q3+q4+4)]<-
1/(abs(b.hat.r[l,l])^gamma)}

for (l in 1:(q6+1)){

wad.r[(l+q0+q1+q2+q3+q4+q5+5),(l+q0+q1+q2+q3+q4+q5+5)]<-
1/(abs(b.hat.r[l,l])^gamma)}

###########################################3

ww1.r<-matrix(numeric(m*m),ncol=m,nrow=m)

for (l in 1:m){

ww1.r[l,l]<-1/w1.r[l,l]

#ifelse(ww1.r[l,l]==Inf,(ww1.r[l,l]<-0),(ww1.r[l,l]<-ww1.r[l,l]))}

x.w1.r<-data1%*%ww1.r

##################

ww2.r<-matrix(numeric(m*m),ncol=m,nrow=m)
```

```
for (l in 1:m){
ww2.r[l,l]<-1/w2.r[l,l]
#ifelse(ww2.r[l,l]==Inf,(ww2.r[l,l]<-0),(ww2.r[l,l]<-ww2.r[l,l]))}
x.w2.r<-data1%*%ww2.r
##################
ww3.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
ww3.r[l,l]<-1/w3.r[l,l]
#ifelse(ww3.r[l,l]==Inf,(ww3.r[l,l]<-0),(ww3.r[l,l]<-ww3.r[l,l]))}
x.w3.r<-data1%*%ww3.r
##################
wwad.r<-matrix(numeric(m*m),ncol=m,nrow=m)
for (l in 1:m){
wwad.r[l,l]<-1/wad.r[l,l]
#ifelse(wwad.r[l,l]==Inf,(wwad.r[l,l]<-0),(wwad.r[l,l]<-wwad.r[l,l]))}
x.wad.r<-data1%*%wwad.r
yt.bar.r<-mean(w.huber*yt)
##################################################
rlasso.w1<-(lars(x.w1.r,(w.huber*yt))$beta)
lambda<-114 lars(x.w1.r,(w.huber*yt))$ lambda
B.w1.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)
for (c in 1:m){
B.w1.r[1:lambda,c]<-(diag(ww1.r))*(rlasso.w1[1:lambda,c])}
error.w1.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))
```

```
for (c in 1:lambda){

error.w1.r[,c]<-((w.huber*yt)-(data1%*%t(B.w1.r)[,c]))}

rpe.w1.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w1.r<-apply((error.w1.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(n-1))

##################################

rlasso.w2<-(lars(x.w2.r,(w.huber*yt))$beta)

B.w2.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.w2.r[1:lambda,c]<-(diag(ww2.r))*(rlasso.w2[1:lambda,c])}

error.w2.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.w2.r[,c]<-((w.huber*yt)-(data1%*%t(B.w2.r)[,c]))}

rpe.w2.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w2.r<-apply((error.w2.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))

########################################

rlasso.w3<-(lars(x.w3.r,(w.huber*yt))$beta)

B.w3.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.w3.r[1:lambda,c]<-(diag(ww3.r))*(rlasso.w3[1:lambda,c])}

error.w3.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.w3.r[,c]<-((w.huber*yt)-(data1%*%t(B.w3.r)[,c]))}
```

```
rpe.w3.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.w3.r<-apply((error.w3.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))

rlasso<-(lars(data1,(w.huber*yt))$beta)

B.lasso.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.lasso.r[1:lambda,c]<-(rlasso[1:lambda,c])}

error.lasso.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.lasso.r[,c]<-((w.huber*yt)-(data1%*%t(B.lasso.r)[,c]))}

rpe.lasso.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.lasso.r<-apply((error.lasso.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-
1))

#############################################

rlasso.wad<-(lars(x.wad.r,(w.huber*yt))$beta)

B.wad.r<-matrix(numeric(lambda*m),ncol=m,nrow=lambda)

for (c in 1:m){

B.wad.r[1:lambda,c]<-(diag(wwad.r))*(rlasso.wad[1:lambda,c])}

error.wad.r<-matrix(numeric(lambda*(T-
max(q0,q1,q2,q3,q4,q5,q6)),ncol=lambda,nrow=(T-max(q0,q1,q2,q3,q4,q5,q6))

for (c in 1:lambda){

error.wad.r[,c]<-((w.huber*yt)-(data1%*%t(B.wad.r)[,c]))}

rpe.wad.r<-matrix(numeric(1*(T-max(q0,q1,q2,q3,q4,q5,q6)),ncol=1,nrow=(T-
max(q0,q1,q2,q3,q4,q5,q6))

rpe.wad.r<-apply((error.wad.r^2),2,mean)/((sum(((w.huber*yt)-yt.bar.r)^2))/(T-1))
```

```
rpe<-
c(min(rpe.w1),min(rpe.w2),min(rpe.w3),min(rpe.lasso),min(rpe.wad),min(rpe.w1.)
,min(rpe.w2.r),min(rpe.w3.r),min(rpe.lasso.r),min(rpe.wad.r))

rpe<-t(as.matrix(rpe))

colnames(rpe)<-c("w1","w2","w3","lasso","adlasso","Robust w1","Robust
w2","Robust w3","Robust lasso","Robust adlasso")

r<-as.vector(rpe)

##############################

barplot(r,col = rainbow(10),names.arg=c("w1","w2","w3","lasso","alasso","Rw1",

"Rw2","Rw3","Mlasso","Malasso"))

plot(yt,type="b",col=1)
```

المستخلص

تعد طرق المربعات الصغرى الجزائية المستخدمة في السلاسل الزمنية طرق ملائمة وشائعة للتعامل مع البيانات عالية الأبعاد ولا سيما التي يكون فيها عدد المتغيرات التوضيحية اكبر من حجم العينة . إن من ضمن مزايا هذه الطرق هي ضمان الحصول على تنبؤ عالي الدقة وكذلك الحصول أيضا على عملية التقدير واختيار المتغيرات في آن واحد ، حيث تقوم بتقليص بعض المعاملات نحو الصفر . إذ إنها تنتج نموذج متبعثر (Model Sparse) أي النموذج الذي يتضمن اقل عدد ممكن من المتغيرات ومن ثم يكون قابل للتفسير بسهوله. بالرغم من تلك المزايا التي تتمتع بها طرق المربعات الصغرى الجزائية إلا إنها تعد طرق غير حصينة بمعنى تأثرها بالقيم الشاذة ، وللتغلب على هذه المشكلة يتم استبدال دالة خسارة المربعات الصغرى الجزائية بدالة خسارة حصينة لنحصل على طرق المربعات الصغرى الجزائية الحصينة ، ويكون المقدر الناتج منها هو مقدر جزائي حصين يستطيع التعامل مع مشكلتي الأبعاد والقيم الشاذة وكذلك يأخذ تأثير فترة الارتداد (lag) بعين الاعتبار. وفي هذه الرسالة تم اقتراح مقدرات جزائية حصينة وهي مقدرات (M-lag weighted lasso) وكذلك مقدرات (MM-lag weighted lasso) لنحصل على مقدرات ذات كفاءة عالية في السلاسل الزمنية .ومن اجل معرفة أفضلية المقدرات تم إجراء المحاكاة بالاعتماد على معيار خطأ التنبؤ النسبي RPE^{w} وتم التوصل إلى أن المقدرات المقترحة حققت كفاءة عالية مقارنة مع باقي المقدرات ولمختلف أحجام العينات .

أما فيما يتعلق بالجانب التطبيقي فقد تم توظيف بيانات طبية حقيقية لأشخاص يعانون من مرض سرطان الرئة إذ جمعت البيانات من مستشفى الديوانية، ومن الأسباب التي تؤدي إلى مرض سرطان الرئة هي وجود الملوثات الكيميائية للماء. وقد تم التوصل إلى إن مقدرات التقلص المقترحة هي الأفضل في تحديد العوامل المؤثرة على مرض سرطان الرئة اعتمادا على اقل قيمة لـ (RPE^{w}).

جامعة القادسية

كلية الإدارة والاقتصاد

قسم الإحصاء

اختيار المتغيرات لنماذج ARMA

مع التطبيق

رسالة ماجستير مقدمة إلى مجلس كلية الإدارة والاقتصاد / جامعة القادسية وهي جزء من متطلبات نيل درجة الماجستير في علوم الإحصاء

من طالبة الماجستير

الاء قاسم ياسين الخفاجي

إشراف

الأستاذ الدكتور

طاهر ريسان دخيل

1440 هـ **٢٠١٩ م**