ROBUST WITHIN GROUP ESTIMATORS WITH ROBUST CENTERING METHODS FOR FIXED EFFECT PANEL DATA MODEL

SIM CHEE YUONG

MASTER OF APPLIED STATISTICS
UNIVERSITI PUTRA MALAYSIA
2019

ROBUST WITHIN GROUP ESTIMATORS WITH ROBUST CENTERING METHODS FOR FIXED EFFECT PANEL DATA MODEL

By

SIM CHEE YUONG

Dissertation Submitted to the Mathematics Department, Universiti Putra Malaysia, in Fulfillment of the Requirements for the Master of Applied Statistics.

July 2019

APPROVAL: CERTIFICATION OF SUPERVISOR

This project paper titled

ROBUST WITHIN GROUP ESTIMATORS

WITH ROBUST CENTERING METHODS

FOR FIXED EFFECT PANEL DATA MODEL

was submitted by

SIM CHEE YUONG

(GS46536)

in partial fulfillment of the requirement for

Master of Applied Statistics

This report was accepted and examined by

as certified by:

PROF. DR HABSHAH MIDI

Project Supervisor

Date: 01/08/2019

DEPARTMENT OF MATHEMATICS

FACULTY OF SCIENCE

UNIVERSITI PUTRA MALAYSIA

SERDANG SELANGOR

DECLARATION

I declare that the thesis is my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously, and is not concurrently, submitted for any other master at Universiti Putra Malaysia or at any other institution.

SIM CHEE YUONG

Date: 01/08/2019

Abstract

ROBUST WITHIN GROUP ESTIMATORS

WITH ROBUST CENTERING METHODS

FOR FIXED EFFECT PANEL DATA MODEL

By

SIM CHEE YUONG

Supervisor : Prof. Dr. Habshah Midi

Department : Department of Mathematics

Faculty : Faculty of Science

The classical Within Group (WG) method by using Least Squares estimator is a Best Linear Unbiased Estimator (BLUE) for parameters estimation of Fixed Effect Panel Data model. However, the classical WG estimator with mean centering is usually affected by the presence of outliers and high leverage points. To rectify this problem, robust estimators which are not much affected by outliers such as Robust Within Group GM (RWGM) and Robust Within Group MM (RWMM) are put forward. Data pretreatment is also performed by considering robust centering methods to eliminate any time-invariant effects. MM centering is proposed as mean centering is greatly affected by outliers and median centering is not able to bring back the linearity of a data. The performances of the WG, RWMM and RWGM methods are evaluated through simulation study and real data analysis. The results of the study indicate that the RWMM based on MM centering outperformed the WG and RWGM estimators.

Abstrak

PENGANGGAR TEGUH DALAM KUMPULAN

DENGAN KAEDAH BERPUSAT TEGUH

BAGI MODEL DATA PANEL KESAN TETAP

Oleh

SIM CHEE YUONG

Penyelia **: Prof. Dr. Habshah Midi**

Jabatan **: Jabatan Matematik**

Fakulti **: Fakulti Sains**

Kaedah klasik Dalam Kumpulan (WG) dengan menggunakan Penganggar Kuasa Dua Terkecil adalah anggaran parameter yang dikenali sebagai penganggar linear tak pincang dan terbaik bagi Model Data Panel Kesan Tetap. Walaubagaimanapun, Penganggar WG Klasik dengan min berpusat biasanya terjejas dengan kehadiran titik terpencil dan titik tuasan tinggi. Untuk mengatasi masalah ini, penganggar teguh yang tidak mudah dipengaruhi oleh titik terpencil seperti Penganggar Tetap Dalam Kumpulan GM (RWGM) dan Penganggar Tetap Dalam Kumpulan MM (RWMM) dicadangkan. Kaedah pra-rawatan data juga dijalankan dengan menggunakan kaedah penganggar teguh berpusat untuk mencegah pengaruh invarian masa terhadap data. Kaedah MM berpusat dicadangkan kerana prestasi min berpusat akan terjejas dengan kehadiran titik terpencil, manakala kaedah median berpusat yang digunakan tidak berjaya mengembalikan data regresi linear. Penilaian prestasi terhadap penganggar

WG, RWMM dan RWGM dijalankan melalui kajian simulasi dan analisis data sebenar. Hasil kajian memaparkan Penganggar Tetap Dalam Kumpulan MM (RWMM) dengan pengunaan MM berpusat adalah lebih baik daripada penganggar WG dan RWGM.

ACKNOWLEDGEMENTS

I would like to thank Universiti Putra Malaysia for offering this great opportunity for me to conduct the research and to deal with the daily life issue. This research has given the awareness to me the importance of an academic research to the society.

Besides that, I would like to express my deep sense of gratitude to my project supervisor, Prof. Dr. Habshah Midi for spending her time to deal with the problems that I faced. The project can be completed successfully is due to the supports in the form of precious advices and comments given by Prof Dr. Habshah Midi based on her wide knowledge in Statistics. In addition, I would like to thank my project supervisor, Prof Dr. Habshah Midi for her encouragement to get the real data and use the real data to perform analysis.

Apart from that, a special thank goes to Department of Environment, Ministry of Energy, Science, Technology, Environment and Climate Change (MESTECC), Malaysia, that has provided the air pollution data to be used in the study.

Last but not least, I would like to thank my friends who always share information with me and my family members who always support me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WG	Within Group Least Squares Estimator
RWGM	Robust Within Group GM estimator
RWMM	Robust Within Group MM-estimator
LS	Least Squares
WLAD	Weighted Least Absolute Deviation
LASSO	Least Absolute Shrinkage and Selection Operator
LAD-LASSO	Least Absolute Deviation with Least Absolute Shrinkage and Selection Operator
WLAD-LASSO	Weighted Least Absolute Deviation with Least Absolute Shrinkage and Selection Operator
SE	Standard Errors
RMSE	Root Mean Square Error
Var	Variance
LTS	Least Trimmed Squares
LMS	Least Median Squares
RD	Robust Distance
MD	Mahalanobis Distance
MVE	Minimum Volume Ellipsoid
MCD	Minimum Covariance Distance
RM	Robustness Measure
API	Air Pollution Index
CO	Carbon Monoxide
SO_2	Sulphur Dioxide
NO_2	Nitrogen Dioxide

O_3	Ozone
PM_{10}	Particular matter of less than 10 microns in size, $\mu g/m^3$
NO	Nitrogen Monoxide
WD	Wind Direction
NO_x	Nitrogen Oxide
Hum	Humidity
Temp	Temperature

CHAPTER 1

INTRODUCTION

1.1 Introduction

Panel data refers to the pooling of cross-sectional data, in which some units of observations such as states, countries, households, firms, industries etc. are measured across time. There are three types of panel data such as balanced panel data, unbalanced panel data and fixed panel data. Panel data is commonly used in economics and finance for research or modelling (Abu Bakar and Habshah, 2015). In regression model, the main approaches to model the panel data are Pooled Regression, Fixed Effects Model and Random Effects Model. In this study, we consider the same set of observation units over time (Greene, 2012), which is defined as fixed panel data with the Fixed Effects model as follow: -

$$y_{it} = \alpha_i + x_{it}'\beta + \varepsilon_{it} \tag{1}$$

where i = 1, ……, n, the i^{th} observation at time series, t = 1, ……, T. y_{it} is the dependent variable, β is K x 1 regression parameters, x_{it} is the i^{th} observation on K independent variables, α_i is the unobservable time-invariant individual effects and considered to be fixed. The ε_{it} denotes the remainder disturbances across time and individual units with an assumption of no endogeneity is applied.

Panel datasets are more oriented towards cross-sectional analysis (Greene, 2012) across time. Nevertheless, panel data has more advantages compare to cross-sectional data, i.e., it considers multiple points of time, but the latter considers a single point of time. For panel data, we also take the same observation units or same individual units over time in the sample, but the latter takes the current proportion of population as sample and hence, the current sample may not be the same as the sample taken for previous cross-sectional analysis. The variables change dynamically and the effects of individual units over time can be identified, in addition, the differences in variables between subjects across periods can also be analyzed with panel datasets. However, cross-sectional data only shows observations that are based on the current population.

1.2 Problem Statement

Similar to multiple linear regression model, the aim of the panel data regression model is to predict the response based on the predictor variables. The parameters of the model in Equation (1), first needs to be estimated. Before parameters estimation, data pretreatment methods are applied and mean centering is often used as a transformation method. But, mean centering is very sensitive to outliers. As an alternative, median centering which is not sensitive to outliers is introduced. Unfortunately, median centering has low efficiency of normal errors. Due to this shortcomings, robust MM centering method is proposed since it has high breakdown point and high efficiency at normal errors.

Once data has been transformed, the commonly used method, i.e. Within Group estimator with OLS method is employed. However, the outlying points can lead to unreliable estimates for the panel data as the outlying points are always not able to be observed through Least Squares fit (Bramati and Croux, 2007). It is noted that outliers can be easily found in panel data due to the presence of errors such as measurement error, typing error, transmission error or natural unusual values (Abu Bakar and Habshah, 2015). Outliers may occur in the points of vertical, horizontal or leverage (Abu Bakar and Habshah, 2015). According to Bramati and Croux (2007), Least Squares estimator suffers from masking effect as OLS method is not capable of detecting outliers. For the past two decades, the practicality of robust estimators in the regression model is well developed, yet, there is still a lack development of robust practice for panel data models (Bramati and Croux, 2007). Robust Within Group GM estimator (RWGM) is proposed to deal with outliers. However, the RWGM estimator tends to swamp some low leverage points as high leverage points and its efficiency tends to decrease as the number of low leverage points increases. Hence, Robust Within Group MM estimator (RWMM) is proposed. The RWMM estimator is expected to perform better than the WG and RWGM.

1.3 Objectives

The aim of this research is to investigate the effect of outliers on the parameter estimation of the fixed effect panel data model. The classical Within Group (WG) estimator is the commonly used method. Nonetheless, it is not resistant to outliers. Therefore, it is crucial to employ robust methods for parameter estimation. This project is mainly focused on the following objectives:

1. To propose using robust MM centering for panel data transformation.
2. To propose RWMM estimator to estimate the parameters of panel data model.
3. To compare the performance of RWMM with the existing methods, WG and RWGM.
4. To employ bootstrapping method to obtain the standard errors for the estimates of panel data model.
5. To apply the LASSO method for variable selection.

1.4 Thesis Structure

In accordance to the objectives of this study, the contents of this research project are outlined in the sequence as below: -

Chapter 2 is a chapter of Literature Review. This chapter covers the existing research papers on centering methods, development of the OLS and classical Robust Within Group methods. The development of Weighted Least Absolute Deviations (WLAD) and Least Absolute Shrinkage and Selection Operator (LASSO) with robust methods for model selections are also presented.

Chapter 3 This chapter briefly discussed the classical within group estimator (WG) and the two robust methods, RWGM and RWMM. The discussions also involved the data pretreatment method by using robust centering methods and weighted LAD-LASSO method for model selection.

Chapter 4 This chapter presents the simulation study in which robustness measure and statistical measures such as bias, standard error (SE) and root mean square error (RMSE) are used to assess the performances of the estimators. Real data analysis is illustrated using environment data. This chapter also includes the conclusion based on the objectives outlined in the research.

Chapter 5 concludes the findings of this study. In this chapter, recommendation for future studies has been presented.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses about a comprehensive review of the literature related to the classical Robust Within Group methods and centering methods for fixed effects panel data model. In addition, model selection methods are also included into the review of the literature, i.e. Least Absolute Shrinkage and Selection Operator (LASSO) method with weighted Least Absolute Deviation (WLAD).

2.2 Comprehensive Reviews

Bramati and Croux (2007) stated that the main capable sources of bias are the contaminated data with vertical outliers and leverage points, and hence, they introduced two robust methods for estimation, such as Robust Within Generalized M (RWGM) estimator and Robust Within Group MS estimator. Based on their simulation and empirical findings, the two robust methods performed equally well with good and stable results over all considered sampling with small standard errors of the estimates. However, according to Yohai (1987), RWGM shows less consistency as high leverage observations may still affect the efficiency of RWGM estimator.

Bramati and Croux (2007) also proposed centering method for data transformation. The centering method is performed to each time series for both the dependent and independent variables. The variables are centered by using median instead of using mean because of the mean is largely affected by the outliers. After data transformation with median centering is done, the robust estimators are applied to deal with any anomalous observations. This research paper is helpful for understanding the application of transformation with centering method and the classical Robust Within Group methods. Even though median centering performs better than mean centering, median centering cannot bring back the linearity of the data (Abu Bakar and Habshah, 2015).

Besides that, it is important to understand the literature of the intermediate processes for the robust methods. Rousseeuw (1984) proposed least median squares (LMS) and least trimmed squares (LTS) estimators. Least median square is defined as the minimization of the median of squares residuals, while least trimmed square is to minimize the trimmed mean of squares residuals. LMS and LTS have high breakdown point as much as 50% but low efficiency. Low efficiency of LMS and LTS are affected by the high leverage points. Hence, LMS and LTS are only suitable to be used as an initial estimator or to act as an intermediate process for other estimators to improve efficiency.

Rousseeuw and Yohai (1984) introduce S estimator with a constant value, c equals to 1.547, which yields high breakdown point. S estimator provides a good approximation to the data as the estimator ignores all the outliers. However, S estimator is insufficient for estimation due to its low efficiency. Hence, S estimator is good to become an initial estimate for M-estimator (Rousseeuw and Yohai, 1984).

MM estimator is recommended by Yohai (1987). MM estimator has the high breakdown point and high efficiency when the errors are normally distributed. There are three stages for MM estimate as the initial scale of S-estimate is computed in first stage and based on the initial scale of S-estimate, the scale of M-estimate is calculated in second stage. Nevertheless, the influence curve is not bounded for MM estimator which means that the robustness of MM estimate is exposed to leverage points. According to Abu Bakar and Habshah (2015), MM centering is recommended for data transformation due to its efficiency and stability compared to other two centering methods. RWGM and RWMM under MM centering provide stable and consistent results. However, RWGM estimator is still sensitive to outliers and the efficiency of RWGM estimator may be affected by high leverage points due to the lack of influential property.

For model selection, Hubert and Rousseeuw (1995) proposed Robust Distances and L1 regression as the method is more efficient than Least Squares estimator to detect outliers. Olcay Arslan (2012) stated that the weighted Least Absolute Deviation (WLAD) based on Least Absolute Shrinkage and Selected Operator (LASSO) method downweighs the leverage points and correctly detects the outliers. WLAD-LASSO method provides an accurate model selection when compared to Least Squares and LAD-LASSO methods. LAD is not an efficient initial estimator as it is very sensitive to the high leverage observations and hence, a more efficient initial estimator should be introduced.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The aims of this section are to present the methodology of variable selection to choose the number of variables to be included in the final model. Transformation by using centering methods are needed to offset the time-invariant effect. The parameters of the Fixed Effect Panel Data is estimated by using classical WG, RWGM and RWMM estimators and their performance are assessed through simulation study and real data.

3.2 Model Selection

Least Absolute Shrinkage and Selection Operator (LASSO) is a L_1-type penalty function. The LASSO method is used to perform model selection and estimation. It is important to note that the LASSO regression estimation is based on least squares (LS) criterion such that the estimator is very sensitive to the outliers or heavy-tailed errors (Olcay Arslan, 2012). So, Least Absolute Deviation (LAD) regression is introduced to detect the outliers. LAD regression is assumed that the random errors, ε_i have median equals to zero (Olcay Arslan, 2012). However, LAD regression is insufficient as it is very sensitive to high leverage points. As an alternative weighted LAD regression is introduced where weight, $(w_i)_{it}$ is obtained by using Robust Distance, RD_{it}. According to Olcay Arslan (2012), LAD criterion is minimized as follows: -

$$\sum_{i=1}^{n} w_i \left| y_i - x_i^T \beta \right| , \quad i = 1, 2, 3, \ldots \ldots, n$$

where β is a coefficient estimator, X_i is the $n\ x\ p$ matrix and y_i is a response vector.

The weighted LAD regression combined with LASSO method is put forward to robustly select the right variables and accurate model (Olcay Arslan, 2012). Hence, the weighted LAD-LASSO are estimated as follow: -

$$\sum_{i=1}^{n} w_i |y_i - x_i^T \beta| + n \sum_{j=1}^{p} \lambda_j |\beta_j|$$

where $\mathrm{i} = 1, 2, 3, \dots \dots, \mathrm{n}$ and $\mathrm{j} = 1, 2, 3, \dots \dots, \mathrm{p}$.

3.3 Data transformation with Centering methods

Data transformation is a process of converting the extracted data from original structure to another structure. One of a common data transformation method is data centering whereby each data point subtracts a constant value and it is applied to remove the fixed effect, α_i. Mean centering is commonly used method for the data transformation.

Mean centering is derived by subtracting each observation x and y by their sample means, respectively. There are two steps involved to perform mean centering as follows: -

i) Find the sample means of x and y.

ii) Compute the transformed values for x and y.

$$\tilde{x}_{it} = x_{it} - \bar{x}_i$$

$$\tilde{y}_{it} = y_{it} - \bar{y}_i \qquad \text{for } 1 \leq i \leq n \text{ and } 1 \leq t \leq T.$$

However, outliers in the data inflate the sample mean value due to the non-robust property of the mean and this will cause the mean centering to be less sensitive to outliers. Bramati and Croux (2007) and Verardi and Wagner (2011) proposed median centering to replace the mean centering.

Median centering is derived by subtracting each observation x and y by their sample medians, respectively. There are two steps involved to perform median centering as follows: -

i) Find the median of x and y.

ii) Compute the transformed values for x and y.

$$\tilde{x}_{it} = \mathrm{x}_{\mathrm{it}} - \mathrm{median}(x_i)$$

$$\tilde{y}_{it} = \mathrm{y}_{\mathrm{it}} - \mathrm{median}(y_i) \qquad \text{for } 1 \leq i \leq n \text{ and } 1 \leq t \leq T.$$

Median centering has low efficiency of normal errors, and hence, MM-centering is proposed. According to Maronna et al. (2006), the location of M-estimate can be treated as a weighted mean. There are two steps to perform the computation of MM-centering as follows: -

i) Find the weight, $\mathrm{w_i}$ from the final step of MM algorithm.

ii) Compute the transformed values for $\tilde{x}_{it}$ and $\tilde{y}_{it}$.

$$\tilde{x}_{it} = x_{it} - \frac{\sum w_i x_i}{\sum w_i}$$

$$\tilde{y}_{it} = y_{it} - \frac{\sum w_i y_i}{\sum w_i} \qquad \text{for } 1 \leq i \leq n \text{ and } 1 \leq t \leq T.$$

where w_i is obtained from the final step of MM algorithm.

3.4 Statistical Analysis by performing RWGM and RWMM

Robust Within Group Generalized M-Estimator (RWGM) and Robust Within Group MM-Estimator (RWMM) are proposed to estimate the panel data model. Both estimators are used to identify real outliers efficiently. According to Bagheri and Habshah (2009), the combination of GM- and MM- estimators or some other combinations of iteration function at the early stage such as π-weights and φ-functions can identify the outliers efficiently and produce an excellent estimator.

3.4.1 Robust Within Group Generalized M-Estimator (RWGM)

Bramati and Croux (2007) employed the Robust Within Group GM-estimator (RWGM) to estimate parameters of the panel data model. A function with π-weights is introduced by Schweppe to downweigh any high leverage points for the solutions of GM-estimator to normal equation as below (Abu Bakar and Habshah, 2015): -

$$\sum_{i=1}^{n} \pi_i \, \varphi\left(\frac{y_i - x_i'\beta}{s\pi_i}\right) = 0$$

where, s is a robust scale estimate and $\varphi -$ function may be a monotonic.

We consider two types of weighting procedures for GM-estimator to improve the efficiency of the estimates. GM-estimator standard practice is to consider Least Trimmed Square (LTS) method as an initial high breakdown estimator at the highest breakdown point of 50% when h = $\left(\frac{nT}{2}\right) + \left(\frac{p+1}{2}\right)$. We obtain initial estimate by applying the Least Trimmed Square (LTS) to the transformed data, as follows: -

$$\hat{\beta}_{LTS} = \arg \begin{matrix} min \\ \beta \end{matrix} \sum_{i=1}^{h} [(\tilde{y}_{it} - \tilde{x}_{it}'\beta)^2]_{i:nT}$$

where h is the number of residuals. Residuals and variance of LTS, σ^2 are computed once the initial estimate is obtained as follows: -

$$r_{it} = \tilde{y}_{it} + \tilde{x}'_{it}\hat{\beta}_{LTS}$$

The first weighting procedure is to downweigh the outliers. Tukey's Biweight, with constant, c is selected as follow:

$$\rho(x) = \begin{cases} \frac{x^2}{2} - \frac{x^4}{2c^2} + \frac{x^6}{6c^4} & for\ |x| \le c \\ \frac{c^2}{6} & for\ |x| > c \end{cases}$$

The diagonal elements, W_r can be calculated with constant, c = 4.685 to obtain a 95% efficiency in the simplified form as follow: -

$$W_r = \begin{cases} (1 - \left(\frac{r_{it}}{c\hat{\sigma}_{LTS}}\right)^2)^2 & for\ \left|\frac{r_{it}}{\hat{\sigma}_{LTS}}\right| \le c \\ 0 & for\ \left|\frac{r_{it}}{\hat{\sigma}_{LTS}}\right| > c \end{cases}$$

It is important to point out that c reflects a balance between efficiency and outlier robustness (Wagenvoort and Waldmann, 2002).

According to Bagheri and Habshah (2009), the choice of diagnostic measures and weights computation can greatly downweigh high leverage points. Since Mahalanobis Distance suffers from masking effect, Robust Distance, RD_{it} is put forward (Rousseeuw and Zomeren, 1990). Robust Distance is calculated under a second weighting scheme as follows: -

$$RD_{it} = \sqrt{(\tilde{x}_{it} - \hat{\mu})\hat{V}^{-1}(\tilde{x}_{it} - \hat{\mu})'}$$

where t = 1, …, T and i = 1, …, n. The location ($\hat{\mu}$) and scale ($\hat{V}$) estimates are calculated using Minimum Volume Ellipsoid (MVE) or Minimum Covariance Distance (MCD).

In this study, we choose Minimum Covariance Distance (MCD) to calculate the location and scale estimates by using fast computation, FastMCD (Salibian-Barrera and Yohai, 2006). Once the Robust Distance, RD_{it} is calculated, the diagonal elements of the second weighting matrix, W_x will be computed as follows: -

$$(W_x)_{it} = \min(1, \frac{\sqrt{\chi^2_{K,0.975}}}{RD_{it}})$$

Under the two weighting schemes, a high breakdown of GM-estimate, $\widehat{\beta}_{\mathrm{RWGM}}$ is obtained as follows: -

$$\widehat{\beta}_{\mathrm{RWGM}} = (\widetilde{\mathrm{X}}'\mathrm{W_x W_r}\widetilde{\mathrm{X}})^{-1}(\widetilde{\mathrm{X}}'\mathrm{W_x W_r}\widetilde{\mathrm{Y}})$$

3.4.2 Robust Within Group MM-Estimator

Since M-estimate is known to have a low breakdown point, Yohai (1987) proposed MM-estimator to remedy the problem. MM-estimates have high breakdown point, high efficiency and only small loss of efficiency in the absence of outliers (Abu Bakar and Habshah, 2015). According to Yohai (1987), three stages are robustly applied to the transformed panel data and the estimates are determined as follows: -

First stage : Compute the initial scale estimate, $\hat{\sigma}_S$ and initial high breakdown coefficient, $\hat{\beta}_0$ by using S-estimator.

Second stage : Based on the initial scale estimate and the initial high breakdown coefficient, compute residuals, r_{it} as follows: -

$$r_{it} = \tilde{y}_{it} + \tilde{x}'_{it}\hat{\beta}_0$$

Obtain M-estimate of scale, $\hat{\sigma}$ with high breakdown point as a solution of: -

$$\frac{1}{n}\sum_{i=1}^{n} \rho_0\left(\frac{r_{it}}{\hat{\sigma}}\right) = b$$

with $\frac{b}{a} = 0.5$ and a = max ρ_0

Third Stage : To get $\widehat{\beta}_{\mathrm{RWMM}}$, the M-estimate is computed as the solution as below:

$$\sum_{i=1}^{n} \varphi_1 \left(\frac{r_{it}}{\hat{\sigma}}\right) x_{it} = 0$$

where $\varphi_1 = \rho_1'$ to achieve high efficiency.

For any outliers with large residuals, a weight of 0 will be assigned. MM-estimator is not only good estimates for large residuals, but also the efficiency of the estimator almost coincides with OLS estimator when no outlier can be found in the data (Abu Bakar and Habshah, 2015). Fast-S and MM-estimator are available in R and it can be computed using the function of ltsReg and lmrob.

3.5 Bootstrapping and Statistical Measures

Bootstrapping is a sampling distribution for a statistic by resampling repeatedly from the data. Bootstrapping is needed to find standard error of the estimates as the distribution of the RWGM and RWMM are unknown, in other words, no mathematical methods is used to derive the closed form of the standard error of the estimates. A fixed-x resampling or residual resampling method is used for the bootstrapping. The steps for bootstrapping are as follows: -

For R = 1, …, 1000,

1) Fit a model to the transformed data or original sample of observations to get $\hat{\beta}$ and the fitted value, $\hat{y}_{it} = f(x_{it}, \hat{\beta})$.
2) Obtain residuals, $r_{it} = y_{it} - \hat{y}_{it}$.
3) Draw r_{it}^* from r_{it} and attach to $\hat{y}_{it}$ to get a fixed-x bootstrap value of $\hat{y}_{it}^*$ where

$$\hat{y}_{it}^* = f(x_{it}, \hat{\beta}) + r_{it}^*$$

4) Fit LS regression to the bootstrapped values $\hat{y}_{it}^*$ on the fixed x values to obtain new coefficient, β^*. Set $\hat{y}_{it}^* = \hat{\beta}_0 + \hat{\beta}_1 x_{it} + r_{it}^*$.

5) Repeat step (3) and step (4) for R times to get $\beta^{*1}, \beta^{*2}, \dots \dots, \beta^{*R}$.

Bias, standard errors and root mean squares errors are the common statistical metrics. Bias is a measurement of the mean difference of the estimators and a true value, β=1. Bias and the mean estimators are calculated as below: -

$$\text{Average}(\hat{\beta}) = \bar{\hat{\beta}} = \frac{\Sigma \hat{\beta}_i}{R}$$

and

$$Bias = \bar{\hat{\beta}} - \beta$$

Standard error is a measure of accuracy of an estimate and can be estimated by using bootstrapping. Mean square error is a measure of overall variability of residuals while root mean square error (RMSE) is the standard deviation of the residuals. RMSE is calculated with the formula below: -

$$\text{MSE} = Bias^2 + \text{Var, where Var} = \frac{\Sigma(\hat{\beta}_i - \beta)^2}{n}$$

$$\text{RMSE} = \sqrt{\text{MSE}}$$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

Numerical examples and Monte Carlo Simulation are carried out to evaluate the performance of the WG, RWMM and RWGM estimators. The numerical examples consist of two data sets whereby the first data set has been analysed by Abu Bakar and Habshah (2015). The second data set is a real data set taken from Department of Environment, Ministry of Energy, Science, Technology, Environment and Climate Change (MESTECC), Malaysia.

4.2 Monte Carlo Simulation Study

In this section, the performance of WG, RWGM and RWMM methods are investigated by using Monte Carlo simulation study. According to Bramati and Croux (2007), the independent variables are generated from a multivariate standard normal distribution, N (0,1) where 1 is a k x 1 vector of ones and the dependent variable is generated based on the panel data model below: -

$$y_{it} = \alpha_i + x_{it}'\beta + \varepsilon_{it} \;, \quad i = 1, \ldots\ldots, N, \quad t = 1, \ldots\ldots, T$$

with $\varepsilon_{it} \sim N\ (0,1), \alpha_i \sim U\ (0,20)$ and the slope coefficient, β is set to a vector of ones.

Data with contamination case and without contamination case are studied by comparing both cases. Data contamination is performed by generating vertical outliers for the selected error term and leverage points in the selected explanatory variables. There are two types of contamination such as random contamination and concentrated contamination. Random contamination is fulfilled by randomly generating the outliers over all observations. Meanwhile, concentrated contamination is fulfilled when the outliers are concentrated in a few time series, thus, creating several outlying blocks. In these Monte Carlo simulation, four types of contamination are considered such as block concentrated vertical, block concentrated leverage, vertical outliers and leverage at two levels of contamination, 5% and 10% respectively. According to Bramati and Croux (2007), several time series are selected randomly from the panel dataset and perform contamination only up to 50% for the block contamination case.

Vertical outliers are also known as outliers in the y-direction are generated by adding an additional term with $\sim N\ (50,1)$ to the selected y values from a few time series in random. Apart from that, bad influential leverage points are also generated by substituting the particular x values corresponding to the contaminated y values, with data points from a K-variate normal distribution $x_{it} \sim N\ (10,1)$. The panel datasets with T = 5, 10, 15 and 20 with n = 25, 50, 100 and 200 are considered in this simulation. For each case, we generated M = 1000 independent simulated panel datasets with K equals to 1.

Data transformation is performed to the simulated panel datasets. The datasets are transformed by using three methods, such as mean centering, median centering and MM-Centering. The WG, RWGM and RWMM were then applied to the data. The root mean square error (RMSE) is computed by comparing slope coefficient, $\widehat{\beta}^{(j)}$ to the true parameter value, $\beta = 1$, with the formula below: -

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left(\widehat{\beta}^{(j)} - \beta\right)^2}$$

where $\widehat{\beta}^{(j)}$ is the estimated slope in j^{th} replication.

The results for uncontaminated data are exhibited in Tables (4.1 – 4.2). Tables (4.3 – 4.6) give the results of bias, SE and RMSE for vertical outliers and bad leverage points of block contamination respectively while Tables (4.7 – 4.10) give the results of bias, SE and RMSE for vertical outliers and leverage points of random contamination respectively. The performances of the estimators are summarized for uncontaminated and contaminated data based on the results in the tables below.

Table 4.1: Bias, SE and RMSE of Uncontaminated Data.
(n = 25 and n = 50, Level of Contamination = 0%)

Contamination Level	Centering Method (Estimation Method)	T	n = 25			n = 50		
			Bias	SE	RMSE	Bias	SE	RMSE
Uncontaminated (0%)	Mean Centering (Within Group)	5	-0.000550	**0.284854**	**0.100711**	-0.001987	**0.142023**	**0.071011**
		10	-0.000289	**0.138236**	**0.069118**	-0.002242	**0.067572**	**0.047781**
		15	0.001691	**0.084851**	**0.051961**	0.001674	**0.044536**	**0.038570**
		20	-0.001378	**0.063114**	**0.044628**	0.000680	**0.031416**	**0.031416**
	Median Centering (RWGM)	5	-0.116223	0.532149	0.188143	-0.119892	0.314335	0.157168
		10	-0.044049	0.179724	0.089862	-0.047952	0.102767	0.072667
		15	-0.035126	0.112742	0.069040	-0.038080	0.066898	0.057935
		20	-0.026488	0.080647	0.057026	-0.025094	0.043167	0.043167
	Median Centering (RWMM)	5	-0.111589	0.452465	0.159970	-0.112884	0.277942	0.138971
		10	-0.045804	0.170281	0.085140	-0.046922	0.096977	0.068573
		15	-0.035626	0.107477	0.065816	-0.037393	0.063719	0.055182
		20	-0.026215	0.076264	0.053927	-0.025462	0.041356	0.041356
	MM Centering (RWGM)	5	**-0.000357**	0.338520	0.119685	-0.001808	0.162711	0.081356
		10	0.000581	0.152108	0.076054	**-0.001668**	0.075356	0.053285
		15	**0.000965**	0.095508	0.058486	**0.000976**	0.050025	0.043323
		20	**-0.001084**	0.069863	0.049401	**0.000414**	0.034593	0.034593
	MM Centering (RWMM)	5	0.000505	0.318751	0.112696	**-0.001304**	0.153986	0.076993
		10	**-0.000133**	0.142812	0.071406	-0.001748	0.071036	0.050230
		15	0.001196	0.088458	0.054169	0.001625	0.046542	0.040307
		20	-0.001111	0.065487	0.046306	0.000586	0.032321	0.032321

Table 4.2: Bias, SE and RMSE of Uncontaminated Data

(n = 100 and n = 200, Level of Contamination = 0%)

Contamination Level	Centering Method (Estimation Method)	T	n = 100			n = 200		
			Bias	SE	RMSE	Bias	SE	RMSE
Uncontaminated (0%)	Mean Centering (Within Group)	5	-0.001337	**0.068236**	**0.048250**	**-0.000089**	**0.034950**	**0.034950**
		10	**0.000636**	**0.034285**	**0.034285**	-0.000553	**0.017298**	**0.024463**
		15	0.001277	**0.021544**	**0.026386**	0.000327	**0.010961**	**0.018986**
		20	0.000229	**0.016198**	**0.022907**	0.000666	**0.008106**	**0.016212**
	Median Centering (RWGM)	5	-0.117353	0.192924	0.136418	-0.118095	0.127768	0.127768
		10	-0.044527	0.059058	0.059058	-0.045940	0.038124	0.053916
		15	-0.037015	0.039194	0.048003	-0.038005	0.025151	0.043563
		20	-0.025807	0.025728	0.036385	-0.025732	0.015868	0.031736
	Median Centering (RWMM)	5	-0.113594	0.178892	0.126496	-0.112807	0.119860	0.119860
		10	-0.044243	0.056855	0.056855	-0.045934	0.037187	0.052591
		15	-0.037603	0.038529	0.047188	-0.038637	0.025004	0.043308
		20	-0.026004	0.025052	0.035429	-0.025915	0.015520	0.031041
	MM Centering (RWGM)	5	0.000169	0.081980	0.057969	-0.001126	0.041675	0.041675
		10	0.001253	0.038518	0.038518	**-0.000302**	0.019162	0.027099
		15	0.001321	0.024330	0.029799	0.000573	0.012193	0.021118
		20	0.000303	0.017963	0.025403	0.000573	0.008987	0.017974
	MM Centering (RWMM)	5	**0.000164**	0.077634	0.054895	-0.000818	0.039467	0.039467
		10	0.001317	0.036294	0.036294	-0.000655	0.018118	0.025622
		15	**0.001262**	0.022892	0.028036	**0.000243**	0.011407	0.019757
		20	**0.000182**	0.016881	0.023873	**0.000548**	0.008418	0.016836

Table 4.3: Bias, SE and RMSE of Contaminated Data with Block Vertical

(n = 25 and n =50, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 25			n = 50		
			Bias	SE	RMSE	Bias	SE	RMSE
Block Vertical (5%)	Mean Centering (Within Group)	5	-0.075544	3.178566	1.123793	-0.037737	1.589159	0.794580
		10	0.035043	1.485146	0.742573	0.006959	0.740801	0.523826
		15	-0.006852	0.797996	0.488671	0.001532	0.421716	0.365217
		20	0.000016	0.498324	0.352368	-0.003239	0.265299	0.265299
	Median Centering (RWGM)	5	-0.126290	0.554082	0.195897	-0.131419	0.335070	0.167535
		10	-0.048960	0.185746	0.092873	-0.052681	0.108457	**0.010579**
		15	-0.037686	0.116654	0.071436	-0.039540	0.068018	0.058905
		20	-0.027700	0.081950	0.057948	-0.026389	0.043937	0.043937
	Median Centering (RWMM)	5	-0.121516	0.479101	0.169388	-0.124388	0.300228	0.150114
		10	-0.049602	0.175428	0.087714	-0.051220	0.102853	0.072728
		15	-0.037811	0.111592	0.068336	-0.038907	0.065105	0.056383
		20	-0.027408	0.077805	0.055016	-0.026754	0.042231	0.042231
	MM Centering (RWGM)	5	-0.013386	0.389154	0.137587	-0.017103	0.188561	0.094280
		10	**-0.003706**	0.161452	0.080726	**-0.004448**	0.079443	0.056175
		15	-0.002115	0.099594	0.060988	**-0.000274**	0.050453	0.043693
		20	-0.002438	0.071113	0.050285	-0.001105	0.034887	0.034887
	MM Centering (RWMM)	5	**-0.010186**	**0.353673**	**0.125042**	**-0.013662**	**0.172718**	**0.086359**
		10	-0.004364	**0.147863**	**0.073931**	-0.004579	**0.074002**	0.052328
		15	**-0.000729**	**0.092075**	**0.056384**	0.000430	**0.046788**	**0.040519**
		20	**-0.002206**	**0.066426**	**0.046970**	**-0.000704**	**0.032357**	**0.032357**
Block Vertical (10%)	Mean Centering (Within Group)	5	-0.070888	4.404920	1.557374	-0.051453	2.250588	1.125294
		10	-0.002933	1.426960	0.713480	0.002379	0.743020	0.525395
		15	0.002144	0.759145	0.464879	0.014466	0.404440	0.350255
		20	-0.004402	0.514589	0.363869	0.009481	0.248722	0.248722
	Median Centering (RWGM)	5	-0.140875	0.587873	0.207844	-0.144890	0.357356	0.178678
		10	-0.045354	0.186720	0.093360	-0.050736	0.106193	0.075089
		15	-0.039251	0.118977	0.072858	-0.043590	0.070521	0.061073
		20	-0.027113	0.080834	0.057158	-0.027552	0.046084	0.046084
	Median Centering (RWMM)	5	-0.132644	0.505676	0.178783	-0.134621	0.317138	0.158569
		10	-0.045930	0.172324	0.086162	-0.049410	0.098550	0.069685
		15	-0.039711	0.114241	0.069958	-0.043665	0.068503	0.059326
		20	-0.027842	0.076740	0.054263	-0.028009	0.044432	0.044432
	MM Centering (RWGM)	5	-0.020535	0.412375	0.145797	-0.016842	0.194328	0.097164
		10	**-0.000040**	0.157898	0.078949	-0.002179	0.075036	0.053058
		15	**-0.001231**	0.099196	0.060745	-0.004954	0.049657	0.043005
		20	**-0.001937**	0.071158	0.050316	-0.002460	0.036723	0.036723
	MM Centering (RWMM)	5	**-0.017441**	**0.367536**	**0.129944**	**-0.015159**	**0.178621**	**0.089311**
		10	-0.000517	**0.143209**	**0.071604**	**-0.000800**	**0.069676**	**0.049269**
		15	-0.001297	**0.092862**	**0.056866**	**-0.004764**	**0.046950**	**0.040660**
		20	-0.002114	**0.066241**	**0.046839**	**-0.002207**	**0.034038**	**0.034038**

Table 4.4: Bias, SE and RMSE of Contaminated Data with Block Vertical

(n = 100 and n = 200, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 100			n = 200		
			Bias	SE	RMSE	Bias	SE	RMSE
Block Vertical (5%)	Mean Centering (Within Group)	5	-0.013452	0.811895	0.574097	0.014756	0.384266	0.384266
		10	-0.015433	0.371659	0.371659	-0.004216	0.189700	0.268277
		15	0.000614	0.205208	0.251327	-0.001555	0.099237	0.171883
		20	0.012026	0.131797	0.186390	0.002776	0.065497	0.130994
	Median Centering (RWGM)	5	-0.128773	0.207674	0.146847	-0.129344	0.138351	0.138351
		10	-0.049770	0.063156	0.063156	-0.050905	0.041229	0.058307
		15	-0.038885	0.040517	0.049623	-0.039972	0.026251	0.045468
		20	-0.027214	0.026543	0.037537	-0.026763	0.016340	0.032680
	Median Centering (RWMM)	5	-0.124096	0.193543	0.136855	-0.124148	0.130806	0.130806
		10	-0.048856	0.060744	0.060744	-0.050665	0.040261	0.056937
		15	-0.039141	0.039746	0.048679	-0.040402	0.026036	0.045095
		20	-0.027385	0.025880	0.036600	-0.026973	0.016044	0.032088
	MM Centering (RWGM)	5	-0.018487	0.095339	0.067415	-0.016103	0.049470	0.049470
		10	-0.002396	0.039447	0.039447	-0.003677	0.019745	0.027923
		15	-0.000692	0.024736	0.030296	-0.001743	0.012635	0.021884
		20	**-0.001272**	0.018178	0.025708	-0.000484	0.009073	0.018145
	MM Centering (RWMM)	5	**-0.016014**	**0.086954**	**0.061486**	**-0.014497**	**0.045414**	**0.045414**
		10	**-0.002088**	**0.036980**	**0.036980**	**-0.003603**	**0.018632**	**0.026349**
		15	**-0.000505**	**0.022996**	**0.028164**	**-0.001705**	**0.011715**	**0.020291**
		20	-0.001307	**0.016949**	**0.023969**	**-0.000327**	**0.008549**	**0.017098**
Block Vertical (10%)	Mean Centering (Within Group)	5	-0.016739	1.059362	0.749082	0.047131	0.553031	0.553031
		10	-0.024176	0.373544	0.373544	0.017869	0.186286	0.263448
		15	**0.001354**	0.188884	0.231335	-0.004774	0.098877	0.171261
		20	0.002070	0.133113	0.188250	-0.001607	0.064930	0.129859
	Median Centering (RWGM)	5	-0.144090	0.228953	0.161894	-0.142520	0.151803	0.151803
		10	-0.050615	0.063803	0.063803	-0.050198	0.041064	0.058073
		15	-0.041559	0.042558	0.052123	-0.040888	0.026777	0.046379
		20	-0.027832	0.026396	0.037329	-0.026994	0.016346	0.032691
	Median Centering (RWMM)	5	-0.138609	0.213303	0.150828	-0.134551	0.140382	0.140382
		10	-0.050837	0.062322	0.062322	-0.050434	0.040223	0.056884
		15	-0.041442	0.041631	0.050987	-0.040705	0.026336	0.045616
		20	-0.027723	0.025832	0.036532	-0.027356	0.016152	0.032303
	MM Centering (RWGM)	5	-0.017698	0.098054	0.069335	-0.015213	0.049681	0.049681
		10	-0.003932	0.039172	0.039172	-0.003218	0.020319	0.028735
		15	-0.002878	0.025207	0.030872	-0.001820	0.012419	0.021510
		20	-0.002029	0.017986	0.025436	**-0.001199**	0.009327	0.018654
	MM Centering (RWMM)	5	**-0.017068**	**0.092866**	**0.065666**	**-0.014208**	**0.045289**	**0.045289**
		10	**-0.003405**	**0.036693**	**0.036693**	**-0.002755**	**0.018712**	**0.026462**
		15	-0.002561	**0.023828**	**0.029183**	**-0.001103**	**0.011663**	**0.020200**
		20	**-0.001642**	**0.016805**	**0.023766**	-0.001303	**0.008646**	**0.017292**

Table 4.5: Bias, SE and RMSE of Contaminated Data with Block Leverage

(n = 25 and n =50, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 25			n = 50		
			Bias	**SE**	**RMSE**	**Bias**	**SE**	**RMSE**
Block Leverage (5%)	Mean Centering (Within Group)	5	3.314829	9.388380	3.319294	3.313450	6.631127	3.315563
		10	3.304579	6.613141	3.306570	3.328645	4.708688	3.329544
		15	3.043413	4.972360	3.044936	3.078573	3.555736	3.079358
		20	2.819081	3.988634	2.820390	2.854797	2.855476	2.855476
	Median Centering (RWGM)	5	-0.103409	0.502310	0.177594	-0.108145	0.293957	0.146979
		10	-0.040694	0.175962	0.087981	-0.044631	0.100386	0.070984
		15	-0.035377	0.114427	0.070072	-0.036587	0.065522	0.056744
		20	-0.026270	0.080842	0.057164	-0.024847	0.042872	0.042872
	Median Centering (RWMM)	5	-0.100472	0.435051	0.153814	-0.103703	0.264589	0.132295
		10	-0.041944	0.166270	0.083135	-0.043835	0.095418	0.067471
		15	-0.035744	0.110008	0.067366	-0.036232	0.062930	0.054499
		20	-0.026126	0.076818	0.054318	-0.025248	0.041327	0.041327
	MM Centering (RWGM)	5	-0.010842	0.381906	0.135024	-0.014363	0.185266	0.092633
		10	**-0.002892**	0.160018	0.080009	**-0.003695**	0.078879	0.055776
		15	-0.001918	0.099021	0.060638	**-0.000009**	0.050257	0.043524
		20	-0.002311	0.070910	0.050141	-0.001022	0.034800	0.034800
	MM Centering (RWMM)	5	**-0.008084**	**0.349522**	**0.123575**	**-0.010940**	**0.169729**	**0.084864**
		10	-0.003578	**0.147471**	**0.073735**	-0.003970	**0.073796**	**0.052182**
		15	**-0.000516**	**0.091993**	**0.056334**	0.000648	**0.046780**	**0.040513**
		20	**-0.002124**	**0.066379**	**0.046937**	**-0.000637**	**0.032364**	**0.032364**
Block Leverage (10%)	Mean Centering (Within Group)	5	3.619830	10.246053	3.622527	3.633732	7.270051	3.635025
		10	3.313542	6.631320	3.315659	3.327669	4.707324	3.328580
		15	3.040222	4.967185	3.041767	3.070182	3.546065	3.070983
		20	2.821746	3.992456	2.823093	2.857497	2.858138	2.858137
	Median Centering (RWGM)	5	-0.098116	0.503411	0.177983	-0.098571	0.278607	0.139304
		10	-0.037597	0.177163	0.088582	-0.043078	0.098813	0.069871
		15	-0.036493	0.115956	0.071009	-0.041226	0.068666	0.059467
		20	-0.025650	0.079820	0.056442	-0.025933	0.045196	0.045196
	Median Centering (RWMM)	5	-0.093033	0.428552	0.151516	-0.094691	0.251407	0.125703
		10	-0.038387	0.164384	0.082192	-0.041949	0.091072	0.064398
		15	-0.037155	0.111538	0.068303	-0.041333	0.066861	0.057904
		20	-0.026357	0.075829	0.053619	-0.026446	0.043563	0.043563
	MM Centering (RWGM)	5	-0.014333	0.396377	0.140140	-0.011790	0.189117	0.094559
		10	0.000610	0.156326	0.078163	-0.001351	0.074601	0.052751
		15	**-0.000982**	0.098764	0.060480	-0.004615	0.049457	0.042831
		20	**-0.001767**	0.070895	0.050131	-0.002394	0.036606	0.036606
	MM Centering (RWMM)	5	**-0.012132**	**0.360734**	**0.127539**	**-0.010408**	**0.174970**	**0.087485**
		10	**0.000082**	**0.143128**	**0.071564**	**-0.000138**	**0.069649**	**0.049249**
		15	-0.001058	**0.092816**	**0.056838**	**-0.004524**	**0.046890**	**0.040608**
		20	-0.001998	**0.066247**	**0.046844**	**-0.002135**	**0.034007**	**0.034007**

Table 4.6: Bias, SE and RMSE of Contaminated Data with Block Leverage

(n = 100 and n = 200, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 100			n = 200		
			Bias	SE	RMSE	Bias	SE	RMSE
Block Leverage (5%)	Mean Centering (Within Group)	5	3.329853	4.710603	3.330899	3.326024	3.326525	3.326526
		10	3.325970	3.326432	3.326433	3.325354	2.351552	3.325596
		15	3.070492	2.507351	3.070866	3.069453	1.772254	3.069635
		20	2.849940	2.015440	2.850262	2.851752	1.425955	2.851910
	Median Centering (RWGM)	5	-0.105140	0.177687	0.125644	-0.105583	0.116009	0.116009
		10	-0.041663	0.056870	0.056870	-0.042914	0.036283	0.051312
		15	-0.036295	0.038758	0.047468	-0.037429	0.024903	0.043133
		20	-0.025717	0.025810	0.036500	-0.025302	0.015728	0.031455
	Median Centering (RWMM)	5	-0.102262	0.165533	0.117050	-0.102399	0.110367	0.110368
		10	-0.041308	0.054930	0.054930	-0.042982	0.035509	0.050217
		15	-0.036694	0.038044	0.046594	-0.038016	0.024776	0.042913
		20	-0.025992	0.025169	0.035595	-0.025566	0.015460	0.030919
	MM Centering (RWGM)	5	-0.015022	0.093353	0.066010	-0.013261	0.048223	0.048223
		10	-0.001544	0.039222	0.039222	**-0.002773**	0.019602	0.027721
		15	-0.000365	0.024673	0.030218	**-0.001452**	0.012556	0.021748
		20	**-0.001170**	0.018125	0.025633	-0.000375	0.009056	0.018113
	MM Centering (RWMM)	5	**-0.013079**	**0.085089**	**0.060167**	**-0.011931**	**0.044313**	**0.044313**
		10	**-0.001376**	**0.036966**	**0.036966**	-0.002873	**0.018592**	**0.026294**
		15	**-0.000246**	**0.023027**	**0.028202**	-0.001472	**0.011689**	**0.020247**
		20	-0.001238	**0.016944**	**0.023962**	**-0.000241**	**0.008550**	**0.017101**
Block Leverage (10%)	Mean Centering (Within Group)	5	3.631722	5.136966	3.632384	3.628299	3.628588	3.628588
		10	3.327922	3.328369	3.328369	3.326840	2.352600	3.327078
		15	3.071498	2.508195	3.071899	3.069559	1.772322	3.069752
		20	2.851421	2.016502	2.851764	2.849543	1.424848	2.849696
	Median Centering (RWGM)	5	-0.100041	0.172743	0.122148	-0.097423	0.109107	0.109107
		10	-0.042496	0.057217	0.057217	-0.041992	0.035949	0.050840
		15	-0.039000	0.040815	0.049988	-0.038255	0.025431	0.044048
		20	-0.026151	0.025491	0.036050	-0.025482	0.015696	0.031392
	Median Centering (RWMM)	5	-0.098259	0.161735	0.114364	-0.094032	0.102363	0.102363
		10	-0.043163	0.056061	0.056061	-0.042615	0.035382	0.050038
		15	-0.038987	0.039983	0.048969	-0.038218	0.025080	0.043440
		20	-0.026124	0.024980	0.035327	-0.025968	0.015557	0.031113
	MM Centering (RWGM)	5	-0.012329	0.095085	0.067236	-0.010033	0.047690	0.047690
		10	-0.003036	0.038898	0.038898	-0.002264	0.020140	0.028483
		15	-0.002580	0.025080	0.030717	-0.001506	0.012349	0.021390
		20	-0.001898	0.017937	0.025367	**-0.001063**	0.009292	0.018583
	MM Centering (RWMM)	5	**-0.012121**	**0.090100**	**0.063710**	**-0.009114**	**0.043549**	**0.043549**
		10	**-0.002698**	**0.038898**	**0.036605**	**-0.001976**	**0.011656**	**0.026379**
		15	**-0.002324**	**0.023802**	**0.029151**	**-0.000872**	**0.011656**	**0.020189**
		20	**-0.001539**	**0.016800**	**0.023758**	-0.001213	**0.008643**	**0.017286**

Table 4.7: Bias, SE and RMSE of Contaminated Data with Vertical Outliers

(n = 25 and n = 50, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 25			n = 50		
			Bias	SE	RMSE	Bias	SE	RMSE
Vertical (5%)	Mean Centering (Within Group)	5	0.036098	3.132329	1.107446	-0.035030	1.544412	0.772206
		10	-0.003428	1.394876	0.697438	-0.027601	0.734883	0.519641
		15	-0.014299	0.957551	0.586378	0.006085	0.466955	0.404395
		20	-0.028638	0.689572	0.487601	-0.028422	0.351362	0.351362
	Median Centering (RWGM)	5	-0.126129	0.556151	0.196629	-0.128616	0.329979	0.164990
		10	-0.048263	0.186306	0.093153	-0.053673	0.110023	0.077798
		15	-0.038187	0.116404	0.071282	-0.041116	0.070457	0.061018
		20	-0.028276	0.082791	0.058542	-0.027464	0.044991	0.044991
	Median Centering (RWMM)	5	-0.121880	0.478074	0.169025	-0.122455	0.296075	0.148038
		10	-0.049527	0.176509	0.088254	-0.052165	0.104049	0.073574
		15	-0.038445	0.111687	0.068394	-0.040596	0.067350	0.058326
		20	-0.027950	0.078561	0.055551	-0.027663	0.043220	0.043220
	MM Centering (RWGM)	5	-0.014911	0.372993	0.131873	-0.016341	0.181548	0.090774
		10	**-0.003153**	0.165554	0.082777	**-0.003986**	0.080335	0.056805
		15	-0.002333	0.100329	0.061439	-0.001572	0.051977	0.045014
		20	-0.001970	0.072848	0.051511	-0.001569	0.036445	0.036445
	MM Centering (RWMM)	5	**-0.012295**	**0.349385**	**0.123526**	**-0.014058**	**0.168130**	**0.084065**
		10	-0.003233	**0.152485**	**0.076243**	-0.004284	**0.074834**	**0.052916**
		15	**-0.001197**	**0.091079**	**0.055774**	**-0.000706**	**0.047857**	**0.041445**
		20	**-0.001724**	**0.067581**	**0.047787**	**-0.001065**	**0.033527**	**0.033527**
Vertical (10%)	Mean Centering (Within Group)	5	**0.007494**	4.077888	1.441751	0.023846	2.191283	1.095641
		10	-0.000975	2.008717	1.004358	-0.016019	0.961654	0.679992
		15	-0.010892	1.395994	0.854868	**-0.003025**	0.646044	0.559491
		20	**0.000285**	0.950333	0.671987	-0.015702	0.495611	0.495611
	Median Centering (RWGM)	5	-0.135261	0.575942	0.203626	-0.140934	0.346703	0.173352
		10	-0.051318	0.194250	0.097125	-0.056416	0.113843	0.080499
		15	-0.043795	0.125403	0.076794	-0.047467	0.075437	0.065330
		20	-0.032587	0.088140	0.062325	-0.031457	0.049013	0.049013
	Median Centering (RWMM)	5	-0.131264	0.507099	0.179286	-0.133557	0.317977	0.158989
		10	-0.051477	0.180753	0.090376	-0.054866	0.106793	0.075514
		15	-0.043640	0.120693	0.073909	-0.046999	0.073292	0.063473
		20	-0.032695	0.084373	0.059661	-0.031646	0.047681	0.047681
	MM Centering (RWGM)	5	-0.014403	0.396137	0.140056	-0.013014	0.201926	0.100963
		10	-0.001265	0.161453	0.080727	-0.003968	0.080935	0.057229
		15	**-0.000352**	0.105903	0.064852	-0.003948	0.053029	0.045925
		20	-0.002351	0.076912	0.054385	-0.002110	0.038983	0.038983
	MM Centering (RWMM)	5	-0.014886	**0.367639**	**0.129980**	**-0.012651**	**0.184145**	**0.092072**
		10	**-0.000466**	**0.148288**	**0.074144**	**-0.002920**	**0.074378**	**0.052593**
		15	-0.000617	**0.097879**	**0.059938**	-0.004170	**0.049242**	**0.042645**
		20	-0.001601	**0.070710**	**0.049999**	**-0.001989**	**0.036415**	**0.036415**

Table 4.8: Bias, SE and RMSE of Contaminated Data with Vertical Outliers

(n = 100 and n = 200, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 100			n = 200		
			Bias	SE	RMSE	Bias	SE	RMSE
Vertical (5%)	Mean Centering (Within Group)	5	**-0.012768**	0.778662	0.550597	**0.009197**	0.387647	0.387647
		10	-0.028516	0.372271	0.372271	-0.010658	0.181524	0.256714
		15	0.015300	0.241393	0.295645	-0.001208	0.118012	0.204402
		20	0.005954	0.178696	0.252714	-0.003271	0.092676	0.185352
	Median Centering (RWGM)	5	-0.129436	0.208100	0.147149	-0.128196	0.137223	0.137223
		10	-0.050048	0.063707	0.063707	-0.051418	0.041820	0.059142
		15	-0.040000	0.041216	0.050479	-0.041305	0.026921	0.046628
		20	-0.028146	0.027211	0.038482	-0.028161	0.016896	0.033792
	Median Centering (RWMM)	5	-0.124566	0.193463	0.136799	-0.123374	0.130090	0.130090
		10	-0.049751	0.062031	0.062031	-0.051270	0.040822	0.057731
		15	-0.040126	0.040425	0.049510	-0.041661	0.026704	0.046253
		20	-0.028233	0.026574	0.037581	-0.028308	0.016619	0.033238
	MM Centering (RWGM)	5	-0.015902	0.093759	0.066297	-0.016319	0.047387	0.047387
		10	-0.003615	0.041474	0.041474	-0.005016	0.020499	0.028990
		15	-0.000693	0.025538	0.031277	**-0.001618**	0.012722	0.022036
		20	-0.000945	0.018807	0.026598	-0.000619	0.009316	0.018633
	MM Centering (RWMM)	5	-0.014601	**0.086312**	**0.061032**	-0.014537	**0.044480**	**0.044480**
		10	**-0.002747**	**0.038303**	**0.038303**	**-0.004806**	**0.019329**	**0.027335**
		15	**-0.000446**	**0.023937**	**0.029316**	-0.001809	**0.012001**	**0.020787**
		20	**-0.000863**	**0.017514**	**0.024768**	**-0.000540**	**0.008690**	**0.017381**
Vertical (10%)	Mean Centering (Within Group)	5	**0.003703**	1.058397	0.748400	-0.028404	0.516194	0.516195
		10	-0.033024	0.486606	0.486606	**-0.000381**	0.255006	0.360633
		15	-0.002306	0.332850	0.407656	0.008246	0.164591	0.285081
		20	-0.021984	0.234166	0.331160	-0.011809	0.125460	0.250919
	Median Centering (RWGM)	5	-0.141475	0.227380	0.160782	-0.138816	0.148470	0.148470
		10	-0.056766	0.069430	0.069430	-0.055388	0.044272	0.062610
		15	-0.046452	0.046081	0.056437	-0.045412	0.029299	0.050748
		20	-0.031948	0.029007	0.041022	-0.031078	0.018284	0.036567
	Median Centering (RWMM)	5	-0.137610	0.212683	0.150390	-0.133046	0.139471	0.139471
		10	-0.056388	0.067505	0.067505	-0.055497	0.043609	0.061673
		15	-0.046019	0.045160	0.055309	-0.044888	0.028752	0.049800
		20	-0.031491	0.028335	0.040072	-0.031098	0.017926	0.035853
	MM Centering (RWGM)	5	-0.017306	0.101549	0.071806	-0.011373	0.049401	0.049401
		10	-0.004300	0.041641	0.041641	-0.004606	0.021763	0.030778
		15	**-0.002154**	0.026668	0.032661	-0.001460	0.013513	0.023406
		20	-0.002130	0.019260	0.027238	**-0.000972**	0.009935	0.019870
	MM Centering (RWMM)	5	-0.016952	**0.093796**	**0.066324**	**-0.010750**	**0.045591**	**0.045591**
		10	**-0.003622**	**0.039010**	**0.039010**	-0.003744	**0.020152**	**0.028499**
		15	-0.002203	**0.025214**	**0.030881**	**-0.000762**	**0.012492**	**0.021637**
		20	**-0.001814**	**0.019260**	**0.025438**	-0.001214	**0.009182**	**0.018364**

Table 4.9: Bias, SE and RMSE of Contaminated Data with Leverage Outliers

(n = 25 and n = 50, Level of Contamination = 5% and 10%)

Contamination Level	**Centering Method (Estimation Method)**	**T**	**n = 25**			**n = 50**		
			Bias	**SE**	**RMSE**	**Bias**	**SE**	**RMSE**
Leverage (5%)	**Mean Centering (Within Group)**	**5**	3.286380	9.307584	3.290728	3.283522	6.571390	3.285695
		10	3.280908	6.565565	3.282782	3.302800	4.672310	3.303822
		15	3.276536	5.352754	3.277879	3.293394	3.803621	3.294032
		20	3.296599	4.663460	3.297564	3.297308	3.297752	3.297752
	Median Centering (RWGM)	**5**	-0.100735	0.499882	0.176735	-0.104599	0.290214	0.145107
		10	-0.038654	0.176138	0.088069	-0.042174	0.098288	0.069500
		15	-0.032858	0.111045	0.068001	-0.035301	0.065766	0.056955
		20	-0.023916	0.078869	0.055769	-0.023163	0.042295	0.042295
	Median Centering (RWMM)	**5**	-0.098003	0.431370	0.152512	-0.099210	0.257904	0.128952
		10	-0.040012	0.167174	0.083587	-0.041327	0.093132	0.065854
		15	-0.033179	0.106670	0.065322	-0.035113	0.062998	0.054558
		20	-0.023884	0.075308	0.053251	-0.023509	0.040664	0.040664
	MM Centering (RWGM)	**5**	-0.011736	0.367912	0.130077	-0.014036	0.179590	0.089795
		10	**-0.001921**	0.163938	0.081969	**-0.002872**	0.079400	0.056144
		15	-0.001727	0.099265	0.060787	-0.001037	0.051495	0.044596
		20	-0.001585	0.072297	0.051122	-0.001236	0.036073	0.036073
	MM Centering (RWMM)	**5**	**-0.009715**	**0.345331**	**0.122093**	**-0.011894**	**0.166221**	**0.083110**
		10	-0.002168	**0.151876**	**0.075938**	-0.003333	**0.074459**	**0.052650**
		15	**-0.000636**	**0.090825**	**0.055619**	**-0.000218**	**0.047733**	**0.041338**
		20	**-0.001526**	**0.067370**	**0.047638**	**-0.000812**	**0.033413**	**0.033413**
Leverage (10%)	**Mean Centering (Within Group)**	**5**	3.586638	10.152576	3.589478	3.594603	7.192023	3.596012
		10	3.596146	7.194770	3.597385	3.592426	5.081313	3.593030
		15	3.586558	5.858145	3.587367	3.590858	4.146792	3.591227
		20	3.596357	5.086893	3.596976	3.591678	3.591991	3.591991
	Median Centering (RWGM)	**5**	-0.086527	0.478911	0.169321	-0.087067	0.260030	0.130015
		10	-0.027806	0.169795	0.084898	-0.032895	0.091339	0.064586
		15	-0.031464	0.112225	0.068724	-0.034919	0.064524	0.055880
		20	-0.022046	0.079739	0.056384	-0.022077	0.043358	0.043358
	Median Centering (RWMM)	**5**	-0.083097	0.419121	0.148182	-0.083542	0.237140	0.118570
		10	-0.028046	0.157716	0.078858	-0.031760	0.085265	0.060291
		15	-0.031842	0.109412	0.067001	-0.034811	0.062720	0.054317
		20	-0.022206	0.076667	0.054211	-0.022337	0.042311	0.042311
	MM Centering (RWGM)	**5**	-0.011771	0.387203	0.136897	-0.009478	0.196245	0.098122
		10	**0.000601**	0.159055	0.079527	-0.002121	0.079579	0.056271
		15	**0.000093**	0.104369	0.063913	**-0.003138**	0.052244	0.045244
		20	-0.001767	0.075765	0.053574	-0.001636	0.038462	0.038462
	MM Centering (RWMM)	**5**	**-0.009499**	**0.365430**	**0.129199**	**-0.007086**	**0.180818**	**0.090409**
		10	0.001034	**0.147646**	**0.073823**	**-0.001306**	**0.073893**	**0.052250**
		15	0.000102	**0.097798**	**0.059889**	-0.003355	**0.048983**	**0.042420**
		20	**-0.001232**	**0.070462**	**0.049824**	**-0.001602**	**0.036283**	**0.036283**

Table 4.10: Bias, SE and RMSE of Contaminated Data with Leverage Outliers

(n = 50 and n = 100, Level of Contamination = 5% and 10%)

Contamination Level	Centering Method (Estimation Method)	T	n = 100 Bias	n = 100 SE	n = 100 RMSE	n = 200 Bias	n = 200 SE	n = 200 RMSE
Leverage (5%)	Mean Centering (Within Group)	5	3.299541	4.667732	3.300585	3.298875	3.299414	3.299414
		10	3.300589	3.301101	3.301101	3.298128	2.332297	3.298366
		15	3.299867	2.694587	3.300183	3.296588	1.903379	3.296750
		20	3.297088	2.331560	3.297323	3.297882	1.649000	3.298000
	Median Centering (RWGM)	5	-0.103799	0.175077	0.123798	-0.102671	0.112988	0.112988
		10	-0.038965	0.055535	0.055535	-0.040659	0.035159	0.049723
		15	-0.034449	0.037747	0.046230	-0.035809	0.024069	0.041689
		20	-0.023631	0.024830	0.035114	-0.023838	0.015111	0.030222
	Median Centering (RWMM)	5	-0.100045	0.162379	0.114819	-0.099424	0.107562	0.107562
		10	-0.038869	0.053707	0.053707	-0.040744	0.034305	0.048515
		15	-0.034710	0.036973	0.045283	-0.036465	0.023987	0.041547
		20	-0.023882	0.024239	0.034278	-0.024034	0.014812	0.029625
	MM Centering (RWGM)	5	-0.012920	0.092162	0.065169	-0.013402	0.046188	0.046188
		10	-0.002478	0.041193	0.041193	-0.003913	0.020236	0.028618
		15	-0.000160	0.025343	0.031039	**-0.001116**	0.012627	0.021871
		20	-0.000579	0.018628	0.026345	**-0.000251**	0.009260	0.018521
	MM Centering (RWMM)	5	**-0.011669**	**0.085866**	**0.060717**	**-0.012051**	**0.043471**	**0.043471**
		10	**-0.001740**	**0.038184**	**0.038184**	**-0.003808**	**0.019143**	**0.027072**
		15	**0.000021**	**0.023865**	**0.029228**	-0.001333	**0.011947**	**0.020693**
		20	**-0.000576**	**0.017442**	**0.024667**	-0.000260	**0.008685**	**0.017370**
Leverage (10%)	Mean Centering (Within Group)	5	3.595701	5.086014	3.596355	3.590424	3.590766	3.590766
		10	3.593650	3.593978	3.593978	3.590801	2.539191	3.590959
		15	3.592548	2.933472	3.592755	3.590293	2.072914	3.590392
		20	3.588030	2.537234	3.588190	3.590574	1.795326	3.590651
	Median Centering (RWGM)	5	-0.086778	0.158249	0.111899	-0.085090	0.098469	0.098469
		10	-0.032706	0.050765	0.050765	-0.031507	0.030143	0.042628
		15	-0.033672	0.037744	0.046227	-0.032634	0.022811	0.039510
		20	-0.021630	0.023650	0.033446	-0.021416	0.014292	0.028584
	Median Centering (RWMM)	5	-0.086259	0.148539	0.105033	-0.082179	0.091915	0.091915
		10	-0.032469	0.049353	0.049353	-0.031658	0.029321	0.041466
		15	-0.033482	0.037143	0.045490	-0.032372	0.022365	0.038737
		20	-0.021449	0.023263	0.032899	-0.021589	0.013958	0.027916
	MM Centering (RWGM)	5	-0.013012	0.098311	0.069516	-0.006471	0.047823	0.047823
		10	-0.002480	0.040789	0.040789	-0.002755	0.021289	0.030107
		15	**-0.001275**	0.026118	0.031988	-0.000535	0.013299	0.023035
		20	-0.001512	0.018986	0.026850	**-0.000428**	0.009759	0.019517
	MM Centering (RWMM)	5	**-0.008962**	**0.093380**	**0.066030**	**-0.003651**	**0.045020**	**0.045020**
		10	**-0.002135**	**0.038599**	**0.038599**	**-0.002070**	**0.019901**	**0.028144**
		15	-0.001476	**0.024985**	**0.030600**	**-0.000003**	**0.012448**	**0.021561**
		20	**-0.001349**	**0.017901**	**0.025315**	-0.000764	**0.009140**	**0.018280**

Based on the simulated results, in general, the greater the time series, the smaller values of SE and RMSE for all estimators. For uncontaminated data, the WG estimator performs better than the two proposed robust methods, but, the results change dramatically in the presence of outliers especially for high leverage points. The results of WG estimator show a significant increase in SE and RMSE for contaminated data. Meanwhile, the SE and RMSE of the WG estimator have larger values for high leverage points when we compare to the results for vertical outliers in the random and block contamination tables. In addition, the results also show that the bias of WG estimator deviates far from 0 for high leverage points.

The RWGM and RWMM estimators give better results under median and MM centering compare to the WG estimator. The SE and RMSE of both robust estimators are smaller than the WG estimator for contaminated data. Nevertheless, in overall simulated results, it can be observed that the SE and RMSE of RWMM estimator under median and MM centering are consistently small and stable for all simulated data. The RWMM estimator also provides close results of SE and RMSE to the results of the WG estimator for uncontaminated data. On the other hand, the bias of the RWMM estimator is also consistently near to 0 for all simulated data, which indicates that it can be proposed as an unbiased estimator.

Another simulation is performed based on robustness measures to evaluate the performance of WG and the two proposed robust estimators. The Robustness Measure (*RM*) is calculated by taking a ratio of the RMSE of WG from uncontaminated data to the RMSE of the respective estimators. The Robustness Measure is computed as follows: -

$$RM = \frac{\text{MSE}\,(\hat{\beta}_{\text{WG (Uncontaminated)}})}{\text{MSE}\,(\hat{\beta}\,)\ \text{of estimators}} \times 100\%$$

It also indicates as the RMSE of WG-mean centering estimator for uncontaminated data is made as a reference point. According to Abu Bakar and Habshah (2015), the higher percentage of the robustness measures indicate the better performance of the respective estimators. It can also indicate as the closer the robustness measures to 100% shows the better performance of the respective estimator.

Tables (4.11 – 4.15) exhibit the results of robustness measure for WG, RWGM and RWMM estimators under mean centering, median centering and MM centering.

Table 4.11: Robustness Measure (%) of Uncontaminated Panel Datasets

Contamination Level	n	T	Mean Centering	Median Centering		MM Centering	
			WG	RWGM	RWMM	RWGM	RWMM
Uncontaminated (0%)	25	5	-	53.53%	62.96%	84.15%	89.37%
		10	-	76.92%	81.18%	90.88%	96.80%
		15	-	75.26%	78.95%	88.84%	95.92%
		20	-	78.26%	82.76%	90.34%	96.38%
	50	5	-	45.18%	51.10%	87.29%	92.23%
		10	-	65.75%	69.68%	89.67%	95.12%
		15	-	66.57%	69.89%	89.03%	95.69%
		20	-	72.78%	75.97%	90.82%	97.20%
	100	5	-	35.37%	38.14%	83.24%	87.90%
		10	-	58.05%	60.30%	89.01%	94.46%
		15	-	54.97%	55.92%	88.55%	94.11%
		20	-	62.96%	64.66%	90.17%	95.95%
	200	5	-	27.35%	29.16%	83.86%	88.56%
		10	-	45.37%	46.52%	90.27%	95.48%
		15	-	43.58%	43.84%	89.90%	96.10%
		20	-	51.08%	52.23%	90.19%	96.29%

Table 4.12: Robustness Measure (%) of Block Vertical Panel Datasets

Contamination Level	n	T	Mean Centering	Median Centering		MM Centering	
			WG	RWGM	RWMM	RWGM	RWMM
Block Vertical (5%)	25	5	8.96%	51.41%	59.46%	73.20%	80.54%
		10	9.31%	74.42%	78.80%	85.62%	93.49%
		15	10.63%	72.74%	76.04%	85.20%	92.15%
		20	12.67%	77.02%	81.12%	88.75%	95.01%
	50	5	8.94%	42.39%	47.30%	75.32%	82.23%
		10	9.12%	62.30%	65.70%	85.06%	91.31%
		15	10.56%	65.48%	68.41%	88.27%	95.19%
		20	11.84%	71.50%	74.39%	90.05%	97.09%
	100	5	8.40%	32.86%	35.26%	71.57%	78.47%
		10	9.22%	54.29%	56.44%	86.91%	92.71%
		15	10.50%	53.17%	54.20%	87.10%	93.69%
		20	12.29%	61.03%	62.59%	89.11%	95.57%
	200	5	9.10%	25.26%	26.72%	70.65%	76.96%
		10	9.12%	41.96%	42.97%	87.61%	92.84%
		15	11.05%	41.76%	42.10%	86.75%	93.57%
		20	12.38%	49.61%	50.52%	89.35%	94.82%
Block Vertical (10%)	25	5	6.47%	48.46%	56.33%	69.08%	77.50%
		10	9.69%	74.03%	80.22%	87.55%	96.53%
		15	11.18%	71.32%	74.27%	85.54%	91.37%
		20	12.26%	78.08%	82.24%	88.70%	95.28%
	50	5	6.31%	39.74%	44.78%	73.08%	79.51%
		10	9.09%	63.63%	68.57%	90.05%	96.98%
		15	11.01%	63.15%	65.01%	89.69%	94.86%
		20	12.63%	68.17%	70.71%	85.55%	92.30%
	100	5	6.44%	29.80%	31.99%	69.59%	73.48%
		10	9.18%	53.74%	55.01%	87.52%	93.44%
		15	11.41%	50.62%	51.75%	85.47%	90.42%
		20	12.17%	61.37%	62.70%	90.06%	96.39%
	200	5	6.32%	23.02%	24.90%	70.35%	77.17%
		10	9.29%	42.13%	43.01%	85.13%	92.45%
		15	11.09%	40.94%	41.62%	88.27%	93.99%
		20	12.48%	49.59%	50.19%	86.91%	93.75%

Table 4.13: Robustness Measure (%) of Block Leverage Panel Datasets

Contamination Level	n	T	Mean Centering	Median Centering		MM Centering	
			WG	RWGM	RWMM	RWGM	RWMM
Block Leverage (5%)	25	5	3.03%	56.71%	65.48%	74.59%	81.50%
		10	2.09%	78.56%	83.14%	86.39%	93.74%
		15	1.71%	74.15%	77.13%	85.69%	92.24%
		20	1.58%	78.07%	82.16%	89.01%	95.08%
	50	5	2.14%	48.31%	53.68%	76.66%	83.68%
		10	1.44%	67.31%	70.82%	85.67%	91.57%
		15	1.25%	67.97%	70.77%	88.62%	95.20%
		20	1.10%	73.28%	76.02%	90.28%	97.07%
	100	5	1.45%	38.40%	41.22%	73.10%	80.19%
		10	1.03%	60.29%	62.42%	87.41%	92.75%
		15	0.86%	55.59%	56.63%	87.32%	93.56%
		20	0.80%	62.76%	64.36%	89.37%	95.60%
	200	5	1.05%	30.13%	31.67%	72.48%	78.87%
		10	0.74%	47.68%	48.72%	88.25%	93.04%
		15	0.62%	44.02%	44.24%	87.30%	93.77%
		20	0.57%	51.54%	52.43%	89.51%	94.80%
Block Leverage (10%)	25	5	2.78%	56.58%	66.47%	71.86%	78.97%
		10	2.08%	78.03%	84.09%	88.43%	96.58%
		15	1.71%	73.18%	76.07%	85.91%	91.42%
		20	1.58%	79.07%	83.23%	89.02%	95.27%
	50	5	1.95%	50.98%	56.49%	75.10%	81.17%
		10	1.44%	68.38%	74.20%	90.58%	97.02%
		15	1.26%	64.86%	66.61%	90.05%	94.98%
		20	1.10%	69.51%	72.12%	85.82%	92.38%
	100	5	1.33%	39.50%	42.19%	71.76%	75.73%
		10	1.03%	59.92%	61.16%	88.14%	93.66%
		15	0.86%	52.79%	53.88%	85.90%	90.51%
		20	0.80%	63.54%	64.84%	90.30%	96.42%
	200	5	0.96%	32.03%	34.14%	73.29%	80.25%
		10	0.74%	48.12%	48.89%	85.89%	92.74%
		15	0.62%	43.10%	43.71%	88.76%	94.04%
		20	0.57%	51.64%	52.11%	87.24%	93.79%

Table 4.14: Robustness Measure (%) of Vertical Panel Datasets

Contamination Level	n	T	Mean Centering	Median Centering		MM Centering	
			WG	RWGM	RWMM	RWGM	RWMM
Vertical (5%)	25	5	9.09%	51.22%	59.58%	76.37%	81.53%
		10	9.91%	74.20%	78.32%	83.50%	90.66%
		15	8.86%	72.89%	75.97%	84.57%	93.16%
		20	9.15%	76.23%	80.34%	86.64%	93.39%
	50	5	9.20%	43.04%	47.97%	78.23%	84.47%
		10	9.19%	61.42%	64.94%	84.11%	90.30%
		15	9.54%	63.21%	66.13%	85.68%	93.06%
		20	8.94%	69.83%	72.69%	86.20%	93.71%
	100	5	8.76%	32.79%	35.27%	72.78%	79.06%
		10	9.21%	53.82%	55.27%	82.67%	89.51%
		15	8.92%	52.27%	53.29%	84.36%	90.01%
		20	9.06%	59.53%	60.95%	86.13%	92.49%
	200	5	9.02%	25.47%	26.87%	73.76%	78.58%
		10	9.53%	41.36%	42.37%	84.38%	89.49%
		15	9.29%	40.72%	41.05%	86.16%	91.34%
		20	8.75%	47.97%	48.78%	87.01%	93.27%
Vertical (10%)	25	5	6.99%	49.46%	56.17%	71.91%	77.48%
		10	6.88%	71.16%	76.48%	85.62%	93.22%
		15	6.08%	67.66%	70.30%	80.12%	86.69%
		20	6.64%	71.61%	74.80%	82.06%	89.26%
	50	5	6.48%	40.96%	44.66%	70.33%	77.13%
		10	7.03%	59.36%	63.27%	83.49%	90.85%
		15	6.89%	59.04%	60.77%	83.98%	90.44%
		20	6.34%	64.10%	65.89%	80.59%	86.27%
	100	5	6.45%	30.01%	32.08%	67.20%	72.75%
		10	7.05%	49.38%	50.79%	82.33%	87.89%
		15	6.47%	46.75%	47.71%	80.79%	85.45%
		20	6.92%	55.84%	57.16%	84.10%	90.05%
	200	5	6.77%	23.54%	25.06%	70.75%	76.66%
		10	6.78%	39.07%	39.67%	79.48%	85.84%
		15	6.66%	37.41%	38.12%	81.12%	87.75%
		20	6.46%	44.33%	45.22%	81.59%	88.28%

Table 4.15: Robustness Measure (%) of Leverage Panel Datasets

Contamination Level	n	T	Mean Centering	Median Centering		MM Centering	
			WG	RWGM	RWMM	RWGM	RWMM
Leverage (5%)	25	5	3.06%	56.98%	66.03%	77.42%	82.49%
		10	2.11%	78.48%	82.69%	84.32%	91.02%
		15	1.59%	76.41%	79.55%	85.48%	93.42%
		20	1.35%	80.02%	83.81%	87.30%	93.68%
	50	5	2.16%	48.94%	55.07%	79.08%	85.44%
		10	1.45%	68.75%	72.56%	85.10%	90.75%
		15	1.17%	67.72%	70.69%	86.49%	93.30%
		20	0.95%	74.28%	77.26%	87.09%	94.02%
	100	5	1.46%	38.97%	42.02%	74.04%	79.47%
		10	1.04%	61.73%	63.84%	83.23%	89.79%
		15	0.80%	57.08%	58.27%	85.01%	90.28%
		20	0.69%	65.24%	66.83%	86.95%	92.87%
	200	5	1.06%	30.93%	32.49%	75.67%	80.40%
		10	0.74%	49.20%	50.42%	85.48%	90.37%
		15	0.58%	45.54%	45.70%	86.81%	91.75%
		20	0.49%	53.64%	54.72%	87.53%	93.33%
Leverage (10%)	25	5	2.81%	59.48%	67.96%	73.57%	77.95%
		10	1.92%	81.41%	87.65%	86.91%	93.63%
		15	1.45%	75.61%	77.55%	81.30%	86.76%
		20	1.24%	79.15%	82.32%	83.30%	89.57%
	50	5	1.97%	54.62%	59.89%	72.37%	78.54%
		10	1.33%	73.98%	79.25%	84.91%	91.45%
		15	1.07%	69.02%	71.01%	85.25%	90.92%
		20	0.87%	72.46%	74.25%	81.68%	86.59%
	100	5	1.34%	43.12%	45.94%	69.41%	73.07%
		10	0.95%	67.54%	69.47%	84.05%	88.82%
		15	0.73%	57.08%	58.00%	82.49%	86.23%
		20	0.64%	68.49%	69.63%	85.32%	90.49%
	200	5	0.97%	35.49%	38.02%	73.08%	77.63%
		10	0.68%	57.39%	59.00%	81.25%	86.92%
		15	0.53%	48.05%	49.01%	82.42%	88.06%
		20	0.45%	56.72%	58.07%	83.06%	88.69%

From the Table 4.11 to Table 4.15, it can be observed that the WG estimator for contaminated data is greatly affected in the presence of vertical outliers and high leverage points. It also indicates that the classical OLS estimator is easily breakdown by any types of outliers. In addition, the inadequate performances of the two proposed robust estimators under median centering are caused by nonlinearity of the transformed data. Nonetheless, the result in the tables show that the robust estimators under MM centering are outperformed other types of robust centering methods. The linearity is brought back to the transformed data by considering the weight for each data points under MM centering method. The robustness measure for RWMM estimator under MM centering is consistently closer to 100% compared to RWGM estimator. Based on the two simulated results, in general, the RWMM estimator performs better than the other two estimators.

4.3 Profit Data

Our first data is taken from Abu Bakar and Habshah (2015). The objective of this section is to apply the two robust methods to the small sample size of panel dataset with only 30 observations. The performances of two robust methods are evaluated, and the effect of outliers are studied. This panel data considers profit as an independent variable, X and investment as a dependent variable, Y over a period of 10 years for 3 firms. Three outliers are introduced to the panel data by inflating three observations in the X variable. The three outliers are created by replacing three good observations in X variable with values equal 68.95, 77.05 and 83.20 (see Table 4.16) as highlighted in bold (in parentheses). For this example, transformation data are performed by using mean centering, median centering and MM centering. Besides that, fixed-x resampling method are performed to estimate the standard errors of the estimates by generating the bootstraps with M = 1000 replications.

Table 4.16: Panel Data for Investment, Y and Profit, X

T	n = 1		n = 2		n = 3	
	Y	X	Y	X	Y	X
1	13.32	12.85	20.30	22.93	8.85	8.65
2	26.30	25.69	17.47	17.96	19.60	16.55
3	2.62	5.48	9.31	9.16	3.87	1.47
4	14.94	13.79 **(68.95)**	18.01	18.73	24.19	24.91
5	15.80	15.41 **(77.05)**	7.63	11.31	3.99	5.01
6	12.20	12.59	19.84	21.15	5.73	8.34
7	14.93	16.64 **(83.20)**	13.76	16.13	26.68	22.70
8	29.82	26.45	10.00	11.61	11.49	8.36
9	20.32	19.64	19.51	19.55	18.49	15.44
10	4.77	5.43	18.32	17.06	20.84	17.87

(Source: Abu Bakar, N.M. & Habshah, M. (2015). Robust centering in the fixed effect panel data model. *Pakistan Journal of Statistics*. 31. 33-48 **[1]**)

Table 4.17: Results for Classical (WG) and Two Robust Methods

		Mean Centering	Median Centering		MM Centering	
		WG	RWGM	RWMM	RWGM	RWMM
Original Data	B_1	1.1020	1.1015	1.1199	1.1038	1.1115
	SE	0.04748	0.03494	0.04399	0.03621	0.04645
Modified Data	B_1	0.1381	0.9332	0.8154	1.0799	1.1024
	SE	0.07046	0.04498	0.03899	0.04737	0.01579

The results of Table 4.17 indicate that the RWMM is the most efficient estimator followed by the RWGM and WG estimators for contaminated data, evident by having the smallest standard errors. The results for both robust methods show small standard errors. The results for original and modified data for RWMM can be interpreted as the mean investment is expected to increase by 1.1115 with 1 dollar increase in profit, meanwhile, the mean investment is expected to increase by 1.1024 with 1 dollar increase in profit, respectively.

4.4 Real Data Analysis – Air Pollution Data

Air pollution is a critical environmental issue affecting human's health in Malaysia. Based on an article from The Star Online, all 111 schools in and around Pasir Gudang area must close due to the emission of toxic gases from a chemical polluted river, Sungai Kim Kim at Johor. In addition, 2775 victims **[16]** were admitted to hospitals and clinics due to the toxic gases from the chemical polluted river. Hence, air quality monitoring status in terms of Air Pollutant Index (API), provided by Department of Environment (DOE) is very important to reflect the level of air pollution in Malaysia.

According to the Environmental Quality Report 2017, API calculation is mainly based on hourly concentration of Carbon Monoxide (CO), Sulphur Dioxide (SO_2), Nitrogen Dioxide (NO_2), Ozone (O_3) and particular matter of less than 10 microns in size, $\mu g/m^3$ (PM_{10}). Table 4.18 indicates the API status which is divided into five categories, such as Good, Moderate, Unhealthy, Very Unhealthy and Hazardous.

Table 4.18: Air Pollutant Index (API)

API	Air Quality Status
0 – 50	Good
51 – 100	Moderate
101 – 200	Unhealthy
201 – 300	Very Unhealthy
> 300	Hazardous

(Source: Environmental Quality Report 2017, Department of Environment, Malaysia)

Based on Figure 4.1, the average API for the year of 2013 is the highest with a value of 195.20 and it is followed by the year of 2015 with a value of 192.29. However, the air quality was improved in the following two years, 2016 and 2017 as the API were stated, on average, as 100.47 and 80.20 respectively. There was an improved of air quality status from unhealthy to moderate over the last 5 years.

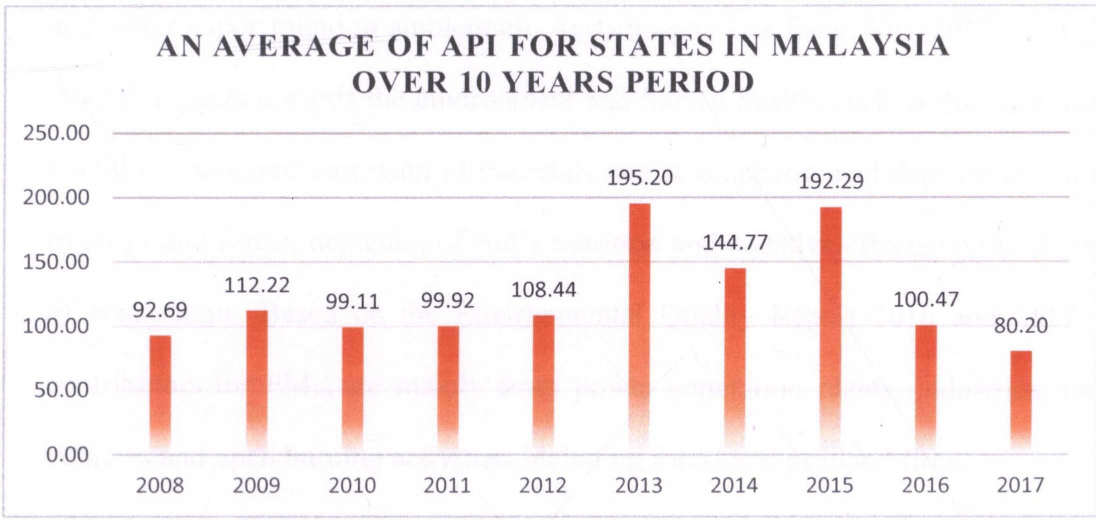

Figure 4.1: Bar Chart shows an Average of API for Selected Stations from All States and Federal Territories in Malaysia for the Year 2008 until Year 2017 (Department of Environment, Malaysia)

All the selected monitoring stations are in urban and industrial areas. Based on Figure 4.2, the air quality status of unhealthy, very unhealthy or hazardous for the states over the past 10 years were Negeri Sembilan and all the Federal Territories (WP), followed by Selangor, Johor, Melaka, Penang, Perak and Kedah. Almost all states in Malaysia showed an unhealthy and very unhealthy air quality status from 2013 to 2015, while the air quality status for Selangor and Melaka were hazardous in 2013 due to illegal forest burning during the dry session.

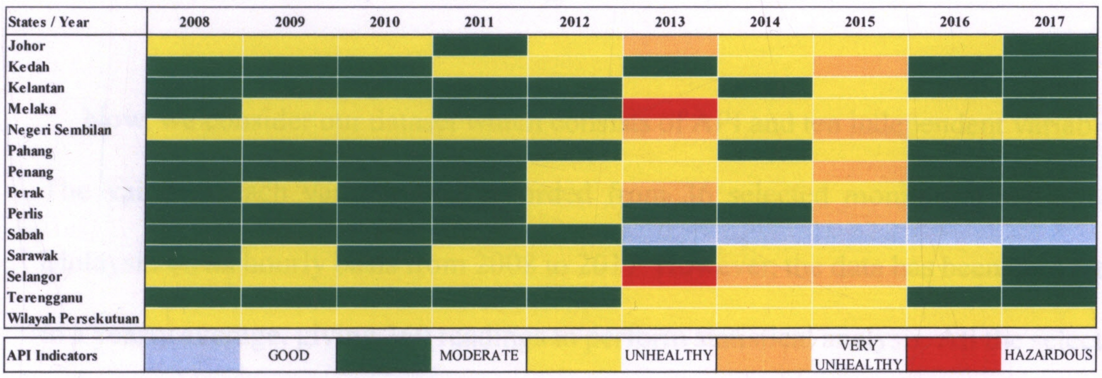

Figure 4.2: The Overall Air Quality for Each States over 10 Years (2008 – 2017)

PM_{10} is one of the major and common pollutant, which consists of solid particles and water vapor found in ambient air. According to Lee Sang Win (2010), PM_{10} has crucial impacts towards the environment and human health, such as the reduction of visibility, increased corrosion of materials and atmospheric acid deposition, damage to crops and forest, depletion of soil's nutrition and negative effects on the diversity of ecosystems. Based on the Environmental Quality Report 2016 and 2017, the contributors for PM_{10} are mainly from power generation plants, industries, motor vehicles and open burning activities, including forest and peatland fires.

In this study, we consider yearly air pollution data which are provided by the Department of Environment (DoE), Malaysia. The data is used to further assess the performance of WG, RWGM and RWMM estimators. Prior to the assessment of the estimators, variable selection methods is performed to select the best model. Variable selection is a process to perform to reduce the number of variables and selecting the most accurate variables to be included into our model. The air pollution data consists the pollutant variables of air pollution index (API), PM_{10} concentration (PM_{10}), Carbon Monoxide (CO), Nitrogen Monoxide (NO), Nitrogen Dioxide (NO_2), Nitrogen Oxide (NOx), Ozone (O_3), Sulphur Dioxide (SO_2), and three meteorological variables are humidity (Hum), temperature (Temp) and wind direction (WD).

Now, we consider our dataset which consists of API and ten independent variables. The value of each variable was recorded from 36 selected monitoring stations in Malaysia on an hourly basis from 2008 to 2017. However, the data has been converted to a yearly average, giving 360 readings to perform statistical analysis. All the selected monitoring stations are in urban and industrial areas of all the states and federal territories in Malaysia. The model selection is performed by using linear regression model. Figure 4.3 and Figure 4.4 show the quantile-quantile (Q-Q) plot to be used as a general residual checking for the models. Since there are points in the Q-Q plot do not fall on the straight line, it indicates that the data is not normal, and it has the chances that both datasets have outliers.

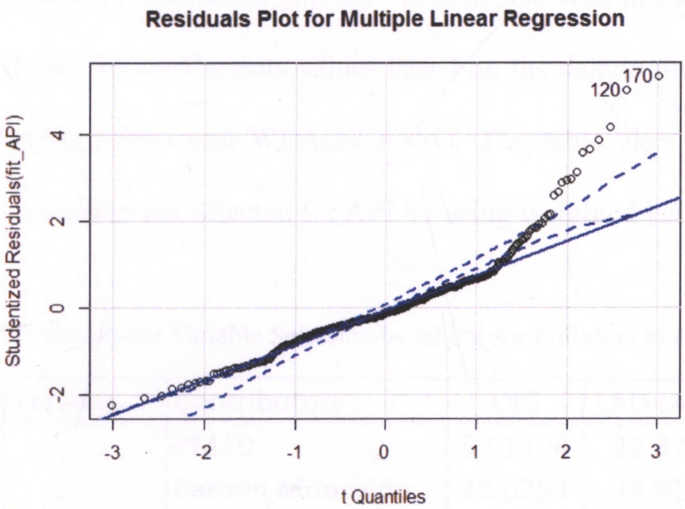

Figure 4.3: Quantile-Quantile Plot (Q-Q plot) of API versus of each component (a total of 10 components) of air quality data.

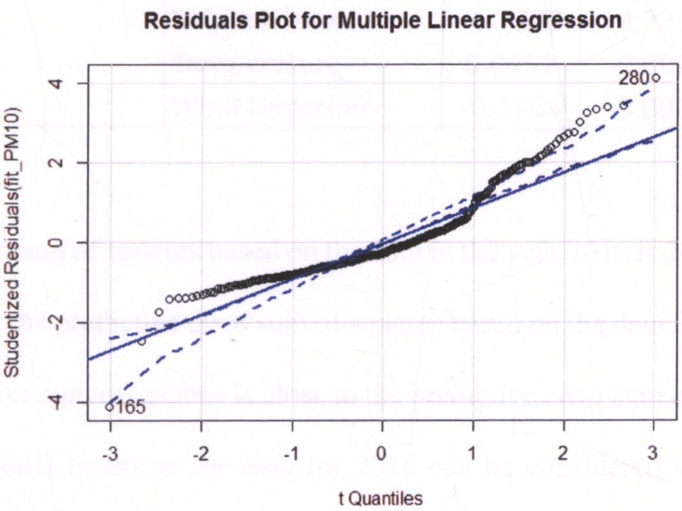

Figure 4.4: Quantile-Quantile Plot (Q-Q plot) of PM_{10} versus of each component (a total of 9 components) of air quality data.

Multiple linear regression with least squares estimator is very sensitive to the outliers and hence, Least Absolute Deviation and Least Absolute Shrinkage and Selection Operator (LAD-LASSO) and its weightage (WLAD-LASSO) are performed for model selection. According to Olcay Arslan (2012), the weighted LAD-LASSO method is less sensitive to leverage points and the outliers and hence, able to improve the robust parameter estimation and the variable selection in the statistical analysis. Table 4.19 shows the coefficients values based on the data in the year 2016 for OLS estimator, LAD-LASSO and WLAD-LASSO. The table also indicates that seven independent variables are selected for API by using weighted LAD-LASSO method.

Table 4.19: Results for Variable Selection by taking Air Pollution as dependent variable.

Dependent variable, Y	Contributors	OLS	LAD-LASSO	WLAD-LASSO
Air Pollution	PM10	57.1204	22.3704	39.8831
	Carbon Monoxide	20.0254	18.9274	-5.34741
	Humidity	0.0181	0.00000	0.00000
	Nitrogen Monoxide	-24.9768	-32.38693	-52.4146
	Nitrogen Dioxide	-2.0187	1.8521	-5.4881
	Nitrogen Oxide	38.9270	35.4458	63.8495
	Ozone	3.2992	0.0000	34.6878
	Sulphur Dioxide	-6.7782	4.2297	1.7140
	Temperature	0.0452	0.0000	0.0000
	Wind Direction	-0.2765	0.0000	0.0000

The error sum of squares based on the data in the year 2016 is 244.1655 for WLAD-LASSO and the predictive error sum of squares based on the data for 2017 is 268.2303. Since the error sum of squares is close to the predictive error sum of squares, the model selection results based on the data for 2016 can be considered valid. Hence, in this study, we consider only seven pollutants variables, namely PM_{10} concentration (PM_{10}), Carbon Monoxide (CO), Nitrogen Monoxide (NO), Nitrogen Dioxide (NO_2), Nitrogen Oxide (NOx), Ozone (O_3) and Sulphur Dioxide (SO_2), to be included in the model.

4.4.1 Panel Data Model and Estimated Results with Real Data Analysis

The yearly data of 25 air quality monitoring stations are selected from 36 stations from the year 2008 to 2017. All selected monitoring station are chosen from all states in Malaysia, which is given 360 readings. The performance of the WG, RWGM and RWMM estimators are compared based on standard errors and the results are shown in Table 4.20. The predicted models for API are shown in Table 4.21.

Table 4.20: Performance SE indicator between WG method and two robust methods.

Centering	Estimator	Air Pollution Contributors						
		PM10	CO	NO	NO_2	NOx	O_3	SO_2
Mean Centering	OLS	0.0146	1.3259	129.3583	97.0349	126.0680	69.4611	287.4241
Median Centering	RWGM	0.0072	1.0412	64.4603	59.3918	60.7455	36.8448	198.2018
	RWMM	0.0063	0.5528	56.2711	42.9967	54.0781	32.1710	128.7654
MM Centering	RWGM	0.0081	1.0374	71.4046	60.4204	67.1021	32.3449	185.9237
	RWMM	**0.0065**	**0.5428**	**57.5832**	**44.0655**	**55.3533**	**26.0803**	**133.6299**

Table 4.21: Model for Predicting API using WG and two robust methods.

Centering	Estimator	Predicted Model
Mean Centering	OLS	$0.4888PM_{10} + 7.6894CO - 193.8529NO - 93.4698NO_2$ $+107.2803NO_x + 359.4715O_3 - 89.3445SO_2$
Median Centering	RWGM	$0.4426PM_{10} - 1.8285CO - 121.9501NO - 161.4275NO_2$ $+144.0175NO_x + 289.6549O_3 - 70.5563SO_2$
	RWMM	$0.4345PM_{10} - 0.4865CO - 128.6128NO - 85.7872NO_2$ $+140.9419NO_x + 250.1740O_3 + 16.0354SO_2$
MM Centering	RWGM	$0.4195PM_{10} - 1.6933CO - 227.7773NO - 99.7374NO_2$ $+218.9190NO_x + 336.2632O_3 - 214.6311SO_2$
	RWMM	$\mathbf{0.4194PM_{10} - 0.4267CO - 198.2932NO}$ $\mathbf{-89.7650NO_2 + 203.1325NO_x + 301.7525O_3}$ $\mathbf{+81.0457SO_2}$

Table 4.20 and Table 4.21 showed the performance of standard error indicator between WG estimator and the two robust methods under mean, median and MM centering. All the result showed that robust estimators perform better than WG estimator. The result showed WG estimator is less suitable to perform prediction for air pollution data with outliers. In overall, the results showed that RWMM estimator under MM centering is still better than RWGM estimator for future API prediction in urban and industrial area, Malaysia. The RWMM estimator under MM centering has small standard errors for all coefficients. Hence, we propose RWMM estimator under MM centering as our final model: -

$$API = 0.4194PM_{10} - 0.4267CO - 198.2932NO - 89.7650NO_2 + 203.1325NO_x + 301.7525O_3 + 81.0457SO_2$$

The results can be improved by taking into consideration the simultaneous problems of multicollinearity and outliers in the data set.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

Fixed Effect Panel Data is not only used by the economists and investors, but it can also be used by any fields, such as the environment field. The proper estimation method should be performed so that we are able to obtain more efficient estimates. Classical WG Least Squares estimator is still the Best Linear Unbiased estimator (BLUE) when there are no outliers in the data. However, the WG estimator is greatly affected by the presence of outliers or leverage points with the biased estimated results.

Our research focuses on the robust estimation methods to overcome the data issue when there are outliers or leverage points. Two proposed robust estimators, RWGM and RWMM are performed to remedy the problem. Centering methods are recommended to offset the time-invariant individual effects. The performances of the WG, RWGM and RWMM are evaluated based on the bias, SE, RMSE and robustness measures. Based on the results, MM centering is the best centering method for contaminated data. In addition, RWMM estimator under MM centering performs better than RWGM for all types of data. Besides that, RWMM estimator has consistent results for all data types and when the data is clean, the estimated results of RWMM are closer to WG estimator. Hence, MM centering is proposed to be selected for data transformation and RWMM estimator is proposed to be used for panel data model estimation.

5.2 Future Studies

In general, the methods of robust estimation are performed at several stages and the combination of several techniques are used to estimate the parameters so that the most desirable properties are obtained. For example, there are three stages to perform RWMM estimator. Hence, the development of improvised Within Group estimator by using a more efficient estimators such as FIMGT of Habshah and Shelan (2018) should be considered in future studies. Future studies may also focus on the establishment of Robust Within Group Estimator to rectify the simultaneous problems of multicollinearity and outliers.

Model selection is very important to select the accurate and the best model to perform estimation. WLAD-LASSO can be an efficient model selection in regression analysis. However, other efficient methods for model selection can still be explored and developed.

REFERENCES

1. Abu Bakar, N.M. & Habshah, M. (2015). Robust centering in the fixed effect panel data model. *Pakistan Journal of Statistics*. 31. 33-48.
2. Arslan, O. (2012). Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression. *Computational Statistics & Data Analysis*, 56(6), 1952–1965.
3. Bagheri, A. and Habshah, M. (2009). Robust Estimations as a Remedy for Multicollinearity Caused by Multiple High Leverage Points. *J. Math. and Statist.*, 5(4), 311-321.
4. Baltagi, B.H. (2005). *Econometric Analysis of Panel Data*. John Wiley & Sons, New York.
5. Bramati, M.C. and Croux, C. (2007). Robust Estimators for the Fixed Effects Panel Data Model. *Econometrics Journal*, 10(3), 521-540.
6. Davison, A.C. and Hinkley, D.V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press.
7. Department of Environment (DoE), (2016), Malaysia, *Malaysia Environmental Quality Report 2016.* Kuala Lumpur: Department of Environment, Ministry of Sciences, Technology and the Environment, Malaysia.
8. Department of Environment (DoE), (2017), Malaysia, *Malaysia Environmental Quality Report 2017.* Kuala Lumpur: Department of Environment, Ministry of Sciences, Technology and the Environment, Malaysia.
9. Greene, W.H. (2012). *Econometric Analysis*, 7th Edition, Prentice Hall.
10. Harrison R. L. (2010). Introduction To Monte Carlo Simulation. *AIP conference proceedings, 1204*, 17–21. doi:10.1063/1.3295638
11. Hassan S. Uraibi, Habshah Midi, and Sohel Rana (2015). Robust Stability Best Subset Selection for Autocorrelated Data Based on Robust Location and Dispersion Estimator. *Journal of Probability and Statistics*, vol. 2015, Article ID 432986, 8 pages, 2015. https://doi.org/10.1155/2015/432986.
12. Hubert, Mia & Rousseeuw, Peter. (1997). Robust regression with both continuous and binary regressors. *Journal of Statistical Planning and Inference*. 57. 153-163.
13. Lee, S. W. (2010). Fine particulate matter measurement and international standardization for air quality and emissions from stationary sources. *Fuel*, 89(4), 874 882.
14. Maronna, R.A., Martin, R.D. and Yohai, V.J. (2006). *Robust Statistics: Theory and Methods*, John Wiley, New York.

15. Mohd Farhaan Shah. (2019, March 18). Chemical dumping: All 111 schools to remained closed for now. *The Star Online*. Retrieved from https://www.thestar.com.my/news/nation/2019/03/18/chemical-dumping-all-111-schools-to-remained-closed-for-now/
16. Mohd Farhaan Shah and Remar Nordin. (2019, March 14). Number rise: 2,775 people affected by Pasir Gudang chemical pollution. *The Star Online*. Retrieved from https://www.thestar.com.my/news/nation/2019/03/14/numbers-rise-2775-people-affected-by-pasir-gudang-chemical-pollution/
17. Rousseeuw, P. (1984). Least Median of Squares Regression. *Journal of the American Statistical Association*, 79(388), 871-880.
18. Rousseeuw, P. (1985). Multivariate Estimation with High Breakdown Point. *Mathematical Statistics and Applications Vol. B*. 283-297.
19. Rousseeuw, P. and Leroy, A.M. (2003). *Robust Regression and Outlier Detection.* John Wiley, New York.
20. Rousseeuw, P & Yohai, V. (1984). *Robust Regression by Means of S-Estimators*. Springer Lecture Notes in Statistics. 26. 256-272.
21. Rousseeuw, P. and Zomeran, B.C.V. (1990). Unmasking Multivariate Outliers and Leverage Points. *J. Amer. Statist. Assoc.*, 85(411), 633-639.
22. Ruppert, D. (1992). Computing S Estimators for Regression and Multivariate Location/Dispersion. *Journal of Computational and Graphical Statistics*, 15(2), 414-427.
23. Salibian-Barrera, M. and Yohai, V.J. (2006). A Fast Algorithm for S-regression Estimates. *Journal of Computational and Graphical Statistics*, 15(2), 414-427.
24. Shelan, S.I. & Habshah, M. (2018). Robust within group estimator for fixed effect panel data. *Pakistan Journal of Statistics*. 34(4). 297-310.
25. Simpson, J.R. and Montgomery, D.C. (1998). The Development and Evaluation of Alternative Generalized M-Estimation Techniques. *Commun. In Statist. – Simul. And Compu.*, 27(4), 999-1018.
26. Susanti, Yuliana & Pratiwi, Hasih & Sulistijowati H., Sri & Liana, Twenty. (2014). M estimation, S estimation, and MM estimation in robust regression. *International Journal of Pure and Applied Mathematics*. 91. 10.12732/ijpam.v91i3.7.
27. Ul-Saufie, Ahmad Zia & Yahaya, Ahmad Shukri & Ramli, Nor & Abdul Hamid, Hazrul. (2012). Robust Regression Models for Predicting PM 10 Concentration in an Industrial Area. *International Journal of Engineering and Technology*, vol.2, no.3, pp.364-370, 2012.
28. Van den Berg, R.A., Hoefsloot, H.C., Westerhuis, J.A. et al (2006). Centering,

scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7(142), 1-15.

29. Verardi, V. and Wagner, J. (2011). Robust Estimation of Linear Fixed Effects Panel Data Models with an Application to the Exporter Productivity Premium. *Journal of Economics and Statistics (Jahrbuecher fuer Nationaloekonomie und Statistik)*. 231(4), 546-557.
30. Wagenvoort, R. and Waldmann, R. (2002). On B-robust Instrumental Variable Estimation of the Linear Model with Panel Data. *Journal of Econometrics*, 106, 297-324.
31. Wang, Hansheng & Li, Guodong & Jiang, Guohua. (2007). Robust Regression Shrinkage and Consistent Variable Selection Through the LAD-Lasso. *Journal of Business & Economic Statistics*. 25. 347-355.
32. Wei Sheng, Tin & Habshah, M. (2018). *The Performance of Robust Within Group GM6 and Robust Within Group MM Estimator for Fixed Effect Panel Data In The Presence of High Leverage Points*. Unpublished Thesis. Universiti Putra Malaysia.
33. Yohai, V.J. (1987). High Breakdown-Point and High Efficiency Estimates for Regression. *The Annals of Statistics*, 15, 642-65.
34. Zaman, A., Reousseeuw, P.J. and Orhan, M. (2001). Econometric Applications of High-Breakdown Robust Regression Techniques. *Economics Letters*, 71(1), 1-8.

APPENDIX

R Programming Code

1 Monte Carlo Simulation

```
library(MASS)
library(robustbase)
rm(list=ls())
set.seed(1)

#Variables
R = 1000
n = 25
t = 5
N = n*t
p = 1
mean = 0
mean.lev = 10
mean.ver = 50
sd = 1
min = 0
max = 20
cpp = 0.05
prob = floor(N*cpp)

##### RWGM #####
GMM_Reg<-function(x,y,iter=50,c=4.685,SEED=TRUE){
 x<- as.matrix(x)
 init_gm<-ltsReg(y~x)
 residuals_gm<-init_gm$residuals
 xx<- cbind(1,x)
 covariance<-cov.rob(x,method="mcd")
 robust<-mahalanobis(x,covariance$center,covariance$cov)
 threshold<-qchisq(0.95,p)
 w.weightage<-threshold/robust
 in_weightage<-c(ifelse(w.weightage<1,w.weightage,1))
 scale_gm<-1.4826*(1+5/(N-p-1))*median(abs(residuals_gm))
```

```
  for(it in 1:iter){
   res_std<-abs(residuals_gm/(scale_gm*in_weightage))
   weighted_gm<-c(ifelse(res_std<=c,(1-(res_std/c)^2)^2,0))
   new_gm<-lsfit(x,y,weighted_gm)
   if(max(abs(new_gm$coef-init_gm$coef))<0.0001)
    break
   init_gm$coef <- new_gm$coef
   residuals_gm <- new_gm$residuals
  }
  residuals_gm <- y - xx%*%new_gm$coef
  if(max(abs(new_gm$coef-init_gm$coef))>=0.0001)
   warning(paste("failed to converged in", iter, "step"))
  list(coef= new_gm$coef, residuals=residuals_gm, w= weighted_gm)
}
```

Simulated Data

```
for(i in 1:R){
 A<-matrix(0,n,1)
 x <- mvrnorm(t, A, diag(n))
 epsilon<- matrix(rnorm(N,mean,sd),t,n)
 time_inv_sim<- runif(n,min=min, max=max)
 time_inv.rep<-function(x,t){
  matrix(rep(time_inv_sim,each=t),nrow=t)
 }
 time_inv_value<-time_inv.rep(1:n,t)
 y = time_inv_value + epsilon + x
```

Matrices

```
#Uncontaminated Beta
beta_OLS<- vector("numeric",R)
beta_Med_RWGM<- vector("numeric",R)
beta_Med_RWMM<- vector("numeric",R)
beta_MM_RWGM<- vector("numeric",R)
beta_MM_RWMM<- vector("numeric",R)

#Block Vertical Beta
beta_OLS.BV<- vector("numeric",R)
beta_Med_RWGM.BV<- vector("numeric",R)
```

```
beta_Med_RWMM.BV<- vector("numeric",R)
beta_MM_RWGM.BV<- vector("numeric",R)
beta_MM_RWMM.BV<- vector("numeric",R)

#Block Leverage Beta
beta_OLS.BL<- vector("numeric",R)
beta_Med_RWGM.BL<- vector("numeric",R)
beta_Med_RWMM.BL<- vector("numeric",R)
beta_MM_RWGM.BL<- vector("numeric",R)
beta_MM_RWMM.BL<- vector("numeric",R)

#Vertical Beta
beta_OLS.Ver<- vector("numeric",R)
beta_Med_RWGM.Ver<- vector("numeric",R)
beta_Med_RWMM.Ver<- vector("numeric",R)
beta_MM_RWGM.Ver<- vector("numeric",R)
beta_MM_RWMM.Ver<- vector("numeric",R)

#Leverage Beta
beta_OLS.Lev<- vector("numeric",R)
beta_Med_RWGM.Lev<- vector("numeric",R)
beta_Med_RWMM.Lev<- vector("numeric",R)
beta_MM_RWGM.Lev<- vector("numeric",R)
beta_MM_RWMM.Lev<- vector("numeric",R)

##### Centering Methods #####
### Uncontaminated Data ###
# Mean Centering
mean_x <- apply(x,2,mean)
mean_x_dm <- t(mean_x*t(matrix(1,t,n)))
mean_centering_x<- x - mean_x_dm
lx <- as.vector(mean_centering_x)

mean_y <- apply(y,2,mean)
mean_y_dm <- t(mean_y*t(matrix(1,t,n)))
mean_centering_y <- y - mean_y_dm
ly <-as.vector(mean_centering_y)
```

```
##### Estimated Coefficients with LS estimator #####
beta_OLS[i] <- coef(summary(lm(ly~lx)))["lx","Estimate"]

# Median Centering
median_x <- apply(x,2,median)
median_x_dm <- t(median_x* t(matrix(1,t,n)))
med_centering_x <- x - median_x_dm
cx <- as.vector(med_centering_x)
median_y <- apply(y,2,median)
median_y_dm <- t(median_y*t(matrix(1,t,n)))
med_centering_y <- y - median_y_dm
cy <- as.vector(med_centering_y)

##### Estimated Coefficients with RWGM and RWMM estimators #####
beta_Med_RWGM[i] <- GMM_Reg(cx,cy)$coef[2]
beta_Med_RWMM[i] <- coef(summary(rlm(cy~cx,method="MM")))[2]

##### Contaminated Data #####
### Block Vertical and Block Leverage ###
## Block Vertical
bv.outliers <- matrix(0,t,n)
row_num <- sample.int(t,1)
bv.rand <- rnorm(prob,mean.ver,sd)

## Block Leverage
oridata.bl <- x
x.bl <- matrix(0,t,n)
bl.rand <- rnorm(prob,mean.lev,sd)

for(m in 1:(0.5*n)){
 bv.outliers[row_num,m] <- NA
 x.bl[row_num,m] <- NA
}
na.bv <- which(is.na(bv.outliers))
na.bl <- which(is.na(x.bl))

for(j in 1:prob){
 bv.outliers[na.bv[j]] <- bv.rand[j]
```

```
  x.bl[na.bl[j]] <- bl.rand[j]
}

bv.outliers[which(is.na(bv.outliers))] <- 0
x.bl[which(is.na(x.bl))] <- 0
x.bl[which(x.bl==0)] <- oridata.bl[which(x.bl==0)]

## Block Vertical ##
y.bv <- y + bv.outliers

##### Block Contamination #####
# Mean Centering
mean_y.bv <- apply(y.bv,2, mean)
mean_y_dm.bv <- t(mean_y.bv*t(matrix(1,t,n)))
mean_centering_y.bv <- y.bv - mean_y_dm.bv
lybv <-as.vector(mean_centering_y.bv)
beta_OLS.BV[i] <- coef(summary(lm(lybv~lx)))["lx","Estimate"]

# Median Centering
median_y.bv <- apply(y.bv,2,median)
median_y_dm.bv <- t(median_y.bv*t(matrix(1,t,n)))
med_centering_y.bv <- y.bv - median_y_dm.bv
cybv <- as.vector(med_centering_y.bv)

beta_Med_RWGM.BV[i] <- GMM_Reg(cx,cybv)$coef[2]
beta_Med_RWMM.BV[i] <- coef(summary(rlm(cybv~cx,method="MM")))[2]

## Block Leverage ##
# Mean Centering
mean_x.bl <- apply(x.bl,2,mean)
mean_x_dm.bl <- t(mean_x.bl*t(matrix(1,t,n)))
mean_centering_x.bl <- x.bl - mean_x_dm.bl
lxbl <- as.vector(mean_centering_x.bl)
beta_OLS.BL[i] <- coef(summary(lm(lybv~lxbl)))["lxbl","Estimate"]

# Median Centering
median_x.bl <- apply(x.bl,2,median)
median_x_dm.bl <- t(median_x.bl*t(matrix(1,t,n)))
```

```
med_centering_x.bl <- x.bl - median_x_dm.bl
cxbl <- as.vector(med_centering_x.bl)
beta_Med_RWGM.BL[i] <- GMM_Reg(cxbl,cybv)$coef[2]
beta_Med_RWMM.BL[i] <- coef(summary(rlm(cybv~cxbl,method="MM")))[2]
```

Random Contamination

```
## Vertical
outlier.ver <- matrix(0,t,n)
toreplace_ver <- sample(x = seq_along(outlier.ver), size = prob, replace = FALSE)
outlier.ver[toreplace_ver] <- rnorm(n = prob, mean=mean.ver, sd = sd)

## Leverage
oridata.lev <-x
x.lev <- matrix(0,t,n)
x.lev[outlier.ver!= 0] <- rnorm(prob,mean.lev,sd)
x.lev[which(x.lev==0)] <- oridata.lev[which(x.lev==0)]

## Vertical ##
y.ver <- y + outlier.ver

# Mean Centering
mean_y.ver <- apply(y.ver,2, mean)
mean_y_dm.ver <- t(mean_y.ver*t(matrix(1,t,n)))
mean_centering_y.ver <- y.ver - mean_y_dm.ver
lyver <-as.vector(mean_centering_y.ver)
beta_OLS.Ver[i] <- coef(summary(lm(lyver~lx)))["lx","Estimate"]   #OLS estimator

# Median Centering
median_y.ver <- apply(y.ver,2,median)
median_y_dm.ver <- t(median_y.ver*t(matrix(1,t,n)))
med_centering_y.ver <- y.ver - median_y_dm.ver
cyver <- as.vector(med_centering_y.ver)

# RWGM and RWMM estimators
beta_Med_RWGM.Ver[i] <- GMM_Reg(cx,cyver)$coef[2]
beta_Med_RWMM.Ver[i] <- coef(summary(rlm(cyver~cx,method="MM")))[2]

## Leverage ##
```

```
# Mean Centering
mean_x.lev <- apply(x.lev,2,mean)
mean_x_dm.lev <- t(mean_x.lev*t(matrix(1,t,n)))
mean_centering_x.lev <- x.lev - mean_x_dm.lev
lxlev <- as.vector(mean_centering_x.lev)
beta_OLS.Lev[i] <- coef(summary(lm(lyver~lxlev)))["lxlev","Estimate"]  #OLS
estimator

# Median Centering
median_x.lev <- apply(x.lev,2,median)
median_x_dm.lev <- t(median_x.lev*t(matrix(1,t,n)))
med_centering_x.lev <- x.lev - median_x_dm.lev
cxlev <- as.vector(med_centering_x.lev)
beta_Med_RWGM.Lev[i] <- GMM_Reg(cxlev,cyver)$coef[2]
beta_Med_RWMM.Lev[i] <- coef(summary(rlm(cyver~cxlev,method="MM")))[2]
```

MM Centering

```
## No Contamination
MM_init<-(rlm(ly~lx,psi= psi.biweight, method="MM"))$w
w.dm <-  matrix(MM_init,ncol=n,nrow=t)

wx.dm <- w.dm*x

mean_x.MM <- apply(wx.dm,2,mean)
mean_x_dm.MM <- t(mean_x.MM*t(matrix(1,t,n)))
mean_cent_x.MM<- x - mean_x_dm.MM
mmx <- as.vector(mean_cent_x.MM)

wy.dm <- w.dm*y

mean_y.MM <- apply(wy.dm,2,mean)
mean_y_dm.MM <- t(mean_y.MM*t(matrix(1,t,n)))
mean_cent_y.MM<- y - mean_y_dm.MM
mmy <- as.vector(mean_cent_y.MM)

combine_mm <- cbind(mmx,mmy)
combine_mm.df<-data.frame(combine_mm)
colnames(combine_mm.df)<-c("X","Y")
```

```
# RWGM and RWMM estimators
beta_MM_RWGM[i] <- GMM_Reg(mmx,mmy)$coef[2]
beta_MM_RWMM[i] <- coef(summary(rlm(Y~X,
data=combine_mm.df,method="MM")))[2]

## Block Contamination ##
#Block Vertical
MM_init.bv <- (rlm(cybv~cx,psi= psi.biweight, method="MM"))$w
wbv.dm <-  matrix(MM_init.bv,ncol=n,nrow=t)

wx_bv.dm <- wbv.dm*x

mean_xbv.MM <- apply(wx_bv.dm,2,mean)
mean_xbv_dm.MM <- t(mean_xbv.MM*t(matrix(1,t,n)))
mean_cent_xbv.MM<- x - mean_xbv_dm.MM
mmxbv <- as.vector(mean_cent_xbv.MM)

wy_bv.dm <- wbv.dm*y.bv

mean_ybv.MM <- apply(wy_bv.dm,2,mean)
mean_ybv_dm.MM <- t(mean_ybv.MM*t(matrix(1,t,n)))
mean_cent_ybv.MM<- y.bv - mean_ybv_dm.MM
mmybv <- as.vector(mean_cent_ybv.MM)

combine_mm_bv <- cbind(mmxbv,mmybv)
combine_mm_bv.df<-data.frame(combine_mm_bv)
colnames(combine_mm_bv.df)<-c("X","Y")

beta_MM_RWGM.BV[i] <- GMM_Reg(mmxbv,mmybv)$coef[2]
beta_MM_RWMM.BV[i] <- coef(summary(rlm(Y~X,
data=combine_mm_bv.df,method="MM")))[2]

# Block Leverage
MM_init.bl <- (rlm(cybv~cxbl,psi= psi.biweight, method="MM"))$w
wbl.dm <-  matrix(MM_init.bl,ncol=n,nrow=t)

wx_bl.dm <- wbl.dm*x.bl
```

```
mean_xbl.MM <- apply(wx_bl.dm,2,mean)
mean_xbl_dm.MM <- t(mean_xbl.MM*t(matrix(1,t,n)))
mean_cent_xbl.MM<- x.bl - mean_xbl_dm.MM
mmxbl <- as.vector(mean_cent_xbl.MM)

wy_bl.dm <- wbl.dm*y.bv

mean_ybl.MM <- apply(wy_bl.dm,2,mean)
mean_ybl_dm.MM <- t(mean_ybl.MM*t(matrix(1,t,n)))
mean_cent_ybl.MM<- y.bv - mean_ybl_dm.MM
mmybl <- as.vector(mean_cent_ybl.MM)

combine_mm_bl <- cbind(mmxbl,mmybl)
combine_mm_bl.df<-data.frame(combine_mm_bl)
colnames(combine_mm_bl.df)<-c("X","Y")

beta_MM_RWGM.BL[i] <- GMM_Reg(mmxbl,mmybl)$coef[2]
beta_MM_RWMM.BL[i] <- coef(summary(rlm(Y~X,
data=combine_mm_bl.df,method="MM")))[2]

## Vertical and Leverage Outliers ##
# Vertical
MM_init.ver <- (rlm(cyver~cx,psi= psi.biweight, method="MM"))$w
wver.dm <-  matrix(MM_init.ver,ncol=n,nrow=t)

wx_ver.dm <- wver.dm*x

mean_xver.MM <- apply(wx_ver.dm,2,mean)
mean_xver_dm.MM <- t(mean_xver.MM*t(matrix(1,t,n)))
mean_cent_xver.MM<- x - mean_xver_dm.MM
mmxver <- as.vector(mean_cent_xver.MM)

wy_ver.dm <- wver.dm*y.ver

mean_yver.MM <- apply(wy_ver.dm,2,mean)
mean_yver_dm.MM <- t(mean_yver.MM*t(matrix(1,t,n)))
mean_cent_yver.MM<- y.ver - mean_yver_dm.MM
```

```
mmyver <- as.vector(mean_cent_yver.MM)

combine_mm_ver <- cbind(mmxver,mmyver)
combine_mm_ver.df<-data.frame(combine_mm_ver)
colnames(combine_mm_ver.df)<-c("X","Y")

beta_MM_RWGM.Ver[i] <- GMM_Reg(mmxver,mmyver)$coef[2]
beta_MM_RWMM.Ver[i] <- coef(summary(rlm(Y~X,
data=combine_mm_ver.df,method="MM")))[2]

# Leverage
MM_init.lev <- (rlm(cyver~cxlev,psi= psi.biweight, method="MM"))$w
wlev.dm <-  matrix(MM_init.lev,ncol=n,nrow=t)

wx_lev.dm <- wlev.dm*x.lev

mean_xlev.MM <- apply(wx_lev.dm,2,mean)
mean_xlev_dm.MM <- t(mean_xlev.MM*t(matrix(1,t,n)))
mean_cent_xlev.MM<- x.lev - mean_xlev_dm.MM
mmxlev <- as.vector(mean_cent_xlev.MM)

wy_lev.dm <- wlev.dm*y.ver

mean_ylev.MM <- apply(wy_lev.dm,2,mean)
mean_ylev_dm.MM <- t(mean_ylev.MM*t(matrix(1,t,n)))
mean_cent_ylev.MM<- y.ver - mean_ylev_dm.MM
mmylev <- as.vector(mean_cent_ylev.MM)

combine_mm_lev <- cbind(mmxlev,mmylev)
combine_mm_lev.df<-data.frame(combine_mm_lev)
colnames(combine_mm_lev.df)<-c("X","Y")

beta_MM_RWGM.Lev[i] <- GMM_Reg(mmxlev,mmylev)$coef[2]
beta_MM_RWMM.Lev[i] <- coef(summary(rlm(Y~X,
data=combine_mm_lev.df,method="MM")))[2]
 }
```

```
# No Contamination
sum_beta_OLS <-sum(beta_OLS)
sum_beta_OLS
avg_beta_OLS <- sum_beta_OLS/R
avg_beta_OLS
sum(beta_Med_RWGM)
sum(beta_Med_RWMM)
sum(beta_MM_RWGM)
sum(beta_MM_RWMM)

sum((beta_OLS-1)^2)
sum((beta_Med_RWGM-1)^2)
sum((beta_Med_RWMM-1)^2)
sum((beta_MM_RWGM-1)^2)
sum((beta_MM_RWMM-1)^2)

rmseols <-sqrt((sum((beta_OLS-1)^2))/R)
rmseols
rmseGM.Med<- sqrt((sum((beta_Med_RWGM-1)^2))/R)
rmseGM.Med
rmserwmm.Med <- sqrt((sum((beta_Med_RWMM-1)^2))/R)
rmserwmm.Med
rmseGM.MM <- sqrt((sum((beta_MM_RWGM-1)^2))/R)
rmseGM.MM
rmserwmm.MM <- sqrt((sum((beta_MM_RWMM-1)^2))/R)
rmserwmm.MM

rmseols/rmseGM.Med
rmseols/rmserwmm.Med
rmseols/rmseGM.MM
rmseols/rmserwmm.MM

# Block Vertical
sum(beta_Med_RWGM.BV)
sum(beta_Med_RWMM.BV)
sum(beta_MM_RWGM.BV)
sum(beta_MM_RWMM.BV)
```

```
sum((beta_OLS.BV-1)^2)
sum((beta_Med_RWGM.BV-1)^2)
sum((beta_Med_RWMM.BV-1)^2)
sum((beta_MM_RWGM.BV-1)^2)
sum((beta_MM_RWMM.BV-1)^2)

rmseols.BV <-sqrt((sum((beta_OLS.BV-1)^2))/R)
rmseGM.Medbv<- sqrt((sum((beta_Med_RWGM.BV-1)^2))/R)
rmserwmm.Medbv <- sqrt((sum((beta_Med_RWMM.BV-1)^2))/R)
rmseGM.MMbv <- sqrt((sum((beta_MM_RWGM.BV-1)^2))/R)
rmserwmm.MMbv <- sqrt((sum((beta_MM_RWMM.BV-1)^2))/R)

rmseols/rmseols.BV
rmseols/rmseGM.Medbv
rmseols/rmserwmm.Medbv
rmseols/rmseGM.MMbv
rmseols/rmserwmm.MMbv

# Block Leverage
sum(beta_Med_RWGM.BL)
sum(beta_Med_RWMM.BL)
sum(beta_MM_RWGM.BL)
sum(beta_MM_RWMM.BL)

sum((beta_OLS.BL-1)^2)
sum((beta_Med_RWGM.BL-1)^2)
sum((beta_Med_RWMM.BL-1)^2)
sum((beta_MM_RWGM.BL-1)^2)
sum((beta_MM_RWMM.BL-1)^2)

rmseols.BL <-sqrt((sum((beta_OLS.BL-1)^2))/R)
rmseGM.Medbl<- sqrt((sum((beta_Med_RWGM.BL-1)^2))/R)
rmserwmm.Medbl <- sqrt((sum((beta_Med_RWMM.BL-1)^2))/R)
rmseGM.MMbl <- sqrt((sum((beta_MM_RWGM.BL-1)^2))/R)
rmserwmm.MMbl <- sqrt((sum((beta_MM_RWMM.BL-1)^2))/R)

rmseols/rmseols.BL
rmseols/rmseGM.Medbl
```

```
rmseols/rmserwmm.Medbl
rmseols/rmseGM.MMbl
rmseols/rmserwmm.MMbl

# Vertical
sum(beta_Med_RWGM.Ver)
sum(beta_Med_RWMM.Ver)
sum(beta_MM_RWGM.Ver)
sum(beta_MM_RWMM.Ver)

sum((beta_OLS.Ver-1)^2)
sum((beta_Med_RWGM.Ver-1)^2)
sum((beta_Med_RWMM.Ver-1)^2)
sum((beta_MM_RWGM.Ver-1)^2)
sum((beta_MM_RWMM.Ver-1)^2)

rmseols.Ver <-sqrt((sum((beta_OLS.Ver-1)^2))/R)
rmseGM.MedVer<- sqrt((sum((beta_Med_RWGM.Ver-1)^2))/R)
rmserwmm.MedVer <- sqrt((sum((beta_Med_RWMM.Ver-1)^2))/R)
rmseGM.MMVer <- sqrt((sum((beta_MM_RWGM.Ver-1)^2))/R)
rmserwmm.MMVer <- sqrt((sum((beta_MM_RWMM.Ver-1)^2))/R)

rmseols/rmseols.Ver
rmseols/rmseGM.MedVer
rmseols/rmserwmm.MedVer
rmseols/rmseGM.MMVer
rmseols/rmserwmm.MMVer

# Leverage
sum(beta_Med_RWGM.Lev)
sum(beta_Med_RWMM.Lev)
sum(beta_MM_RWGM.Lev)
sum(beta_MM_RWMM.Lev)

sq_beta_diff.olsLev<-sum((beta_OLS.Lev-1)^2)
sq_beta_diff_rwgm.MedLev<-sum((beta_Med_RWGM.Lev-1)^2)
sq_beta_diff_rwmm.MedLev<-sum((beta_Med_RWMM.Lev-1)^2)
sq_beta_diff_rwgm.MMLev<-sum((beta_MM_RWGM.Lev-1)^2)
```

```
sq_beta_diff_rwmm.MMLev<-sum((beta_MM_RWMM.Lev-1)^2)

rmseols.Lev <-sqrt((sum((beta_OLS.Lev-1)^2))/R)
rmseGM.MedLev<- sqrt((sum((beta_Med_RWGM.Lev-1)^2))/R)
rmserwmm.MedLev <- sqrt((sum((beta_Med_RWMM.Lev-1)^2))/R)
rmseGM.MMLev <- sqrt((sum((beta_MM_RWGM.Lev-1)^2))/R)
rmserwmm.MMLev <- sqrt((sum((beta_MM_RWMM.Lev-1)^2))/R)

rmseols/rmseols.Lev
rmseols/rmseGM.MedLev
rmseols/rmserwmm.MedLev
rmseols/rmseGM.MMLev
rmseols/rmserwmm.MMLev
```

2 Model Selection with WLAD-LASSO (FOR API)

```
library(glmnet)
library(robustbase)
library(L1pack)
library(stats)
library(MASS)

rm(list=ls())
setwd("D:/User/Dataset/")

p<-9
N <-360
data5s <- read.csv("data_10.csv")

raw_x <-
cbind(data5s$PM10_Conc,data5s$Co_Conc,data5s$Humidity,data5s$No,data5s$No
2_Conc,data5s$Nox,data5s$O3_Conc,data5s$So2_Conc,data5s$Temp)
colnames(raw_x) <-
c("PM10","CO","Hum","NO","NO2","NOx","O3","SO2","Temp")
raw_y <- data5s$API

ladlasso<-function(x,y){
  x<-as.matrix(x)
  n<-nrow(x)
```

```
  p<-ncol(x)

  #unpenalized LAD estimator
  beta0<-rq(y~0+x)$coeff
  lambda<-log(n)/(n*abs(beta0))
  #lambda=1/((abs(beta0)^7))

  #augmented data matrix
  y0<-rep(0,p) ; y1<-c(y,y0)
  x0<-n*lambda*diag(p) ; x1<-rbind(x,x0)
  x1<-as.matrix(x1)
  y1<-as.vector(y1)
  # LAD-lasso estimator : it needs library "quantreg"
  as.vector(rq(y1~0+x1)$coeff)
}

Croux.weight<-function(x,n,p){
  #Croux's weight
  rob.est<-CovMve(x,alpha=0.85)
  RDi<-mahalanobis(x,rob.est$center,rob.est$cov)
  chi.cut<-qchisq(0.95,p)
  pmin(1,chi.cut/RDi)
}
wts<-Croux.weight(x,n,p)
wladlasso<-function(x,y,wts){
  n<-nrow(x)
  p<-ncol(x)
  # unpenalized wLAD estimator
  wts<-Croux.weight(x,n,p)
  beta0<-rq(y~0+x,weights=wts)$coeff
  #lambda<-log(n)/(n*abs(beta0))
  lambda=1/(abs(beta0)^7)
  # augmented data matrix
  y0<-rep(0,p) ; y1<-c(y,y0)
  x0<-n*lambda*diag(p) ;
  x1<-rbind(x,x0)
  # wLAD-lasso estimator : it needs library "quantreg"
  wts<-c(wts,rep(1,p))
```

```
  as.vector(rq(y1~0+x1,weights=wts)$coeff)
}
```

3 Real Data Analysis

```
library(MASS)
library(robustbase)
library(MuMIn)
library(dplyr)
library(complmrob)

rm(list=ls())
setwd("D:/User/Dataset")
data <- read.csv("data_10.csv")

## 70% of the sample size
ID<-levels( factor( data$ï..Site_Id) )
split <- sample.split(ID,SplitRatio=0.7)
train <- subset(ID,split==TRUE)
test <- subset(ID,split==FALSE)
train;test

train_data<-read.csv("train_data10s.csv")
test_data<-read.csv("test_data10s.csv")

#Train Data
x1 <- train_data$Co_Conc
x2 <- train_data$Humidity
x3 <- train_data$No
x4 <- train_data$No2_Conc
x5 <- train_data$Nox
x6 <- train_data$O3_Conc
x7 <- train_data$So2_Conc
x8 <- train_data$Wd

x<-cbind(x1,x2,x3,x4,x5,x6,x7,x8)
colnames(x) <- c("CO","Hum","NO","NO2","NOX","O3","SO2","WD")

y <- train_data$PM10_Conc
```

```
p=8
n.train=25
n.test=11
t=10
N.train = n.train*t
N.test = n.test*t

beta1.LM<- vector("numeric",1000)
beta2.LM<- vector("numeric",1000)
beta3.LM<- vector("numeric",1000)
beta4.LM<- vector("numeric",1000)
beta5.LM<- vector("numeric",1000)
beta6.LM<- vector("numeric",1000)
beta7.LM<- vector("numeric",1000)
beta8.LM<- vector("numeric",1000)

beta1.GM<- vector("numeric",1000)
beta2.GM<- vector("numeric",1000)
beta3.GM<- vector("numeric",1000)
beta4.GM<- vector("numeric",1000)
beta5.GM<- vector("numeric",1000)
beta6.GM<- vector("numeric",1000)
beta7.GM<- vector("numeric",1000)
beta8.GM<- vector("numeric",1000)

beta1.MM<- vector("numeric",1000)
beta2.MM<- vector("numeric",1000)
beta3.MM<- vector("numeric",1000)
beta4.MM<- vector("numeric",1000)
beta5.MM<- vector("numeric",1000)
beta6.MM<- vector("numeric",1000)
beta7.MM<- vector("numeric",1000)
beta8.MM<- vector("numeric",1000)

beta1.MM.GM<- vector("numeric",1000)
beta2.MM.GM<- vector("numeric",1000)
beta3.MM.GM<- vector("numeric",1000)
```

```
beta4.MM.GM<- vector("numeric",1000)
beta5.MM.GM<- vector("numeric",1000)
beta6.MM.GM<- vector("numeric",1000)
beta7.MM.GM<- vector("numeric",1000)
beta8.MM.GM<- vector("numeric",1000)

beta1.MM.MM<- vector("numeric",1000)
beta2.MM.MM<- vector("numeric",1000)
beta3.MM.MM<- vector("numeric",1000)
beta4.MM.MM<- vector("numeric",1000)
beta5.MM.MM<- vector("numeric",1000)
beta6.MM.MM<- vector("numeric",1000)
beta7.MM.MM<- vector("numeric",1000)
beta8.MM.MM<- vector("numeric",1000)

GMM_Reg<-function(x,y,iter=50,c=4.685,SEED=TRUE){
 x<- as.matrix(x)
 xx<- cbind(1,x)
 covariance<-cov.rob(x,method="mcd")
 robust<-mahalanobis(x,covariance$center,covariance$cov)
 chi.sq<-qchisq(0.95,p)
 w.weightage<-chi.sq/robust
 in_weightage<-c(ifelse(w.weightage<1,w.weightage,1))
 init_gm<-ltsReg(y~x)
 residuals_gm<-init_gm$residuals
 scale_gm<-1.4826*(1+5/(N.train-p-1))*median(abs(residuals_gm))
 for(it in 1:iter){
  res_std<-abs(residuals_gm/(scale_gm*in_weightage))
  weighted_gm<-c(ifelse(res_std<=c,(1-(res_std/c)^2)^2,0))
  new_gm<-lsfit(x,y,weighted_gm)
  if(max(abs(new_gm$coef-init_gm$coef))<0.0001)
   break
  init_gm$coef <- new_gm$coef
  residuals_gm <- new_gm$residuals
 }
 residuals_gm <- y - xx%*%new_gm$coef
 if(max(abs(new_gm$coef-init_gm$coef))>=0.0001)
  warning(paste("failed to converged in", iter, "step"))
```

```
 list(coef= new_gm$coef, residuals=residuals_gm, w= weighted_gm)
}

# Mean Centering
ly.mean <- y - y.mean_matrix
ly <- as.vector(ly.mean)

lx1.mean <- x1.CO - x1.mean_matrix
lx1 <- as.vector(lx1.mean)

lx2.mean <- x2.HUM - x2.mean_matrix
lx2 <- as.vector(lx2.mean)

lx3.mean <- x3.NO - x3.mean_matrix
lx3 <- as.vector(lx3.mean)

lx4.mean <- x4.NO2 - x4.mean_matrix
lx4 <- as.vector(lx4.mean)

lx5.mean <- x5.NOX - x5.mean_matrix
lx5 <- as.vector(lx5.mean)

lx6.mean <- x6.O3 - x6.mean_matrix
lx6 <- as.vector(lx6.mean)

lx7.mean <- x7.SO2 - x7.mean_matrix
lx7 <- as.vector(lx7.mean)

lx8.mean <- x8.WD - x8.mean_matrix
lx8 <- as.vector(lx8.mean)

lm.mean <- lm(ly~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8)

ly.new <- summary(lm.mean)$coefficients[2,1]*lx1 +
summary(lm.mean)$coefficients[3,1]*lx2 +
summary(lm.mean)$coefficients[4,1]*lx3 +
summary(lm.mean)$coefficients[5,1]*lx4 +
summary(lm.mean)$coefficients[6,1]*lx5 +
```

```
summary(lm.mean)$coefficients[7,1]*lx6 +
summary(lm.mean)$coefficients[8,1]*lx7 +
summary(lm.mean)$coefficients[9,1]*lx8
ly.e <- ly-ly.new

for(i in 1:1000){
 ly.s <- sample(ly.e, length(ly.e), replace=T)
 ly.a <- ly.s + ly.new
 beta1.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[2,1])
 beta2.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[3,1])
 beta3.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[4,1])
 beta4.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[5,1])
 beta5.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[6,1])
 beta6.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[7,1])
 beta7.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[8,1])
 beta8.LM[i] <-
(summary(lm(ly.a~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8))$coefficients[9,1])

}
mean(beta1.LM)
mean(beta2.LM)
mean(beta3.LM)
mean(beta4.LM)
mean(beta5.LM)
mean(beta6.LM)
mean(beta7.LM)
mean(beta8.LM)

#Standard Error
#1. Mean Centering (LM)
sqrt(var(beta1.LM))
```

```
sqrt(var(beta2.LM))
sqrt(var(beta3.LM))
sqrt(var(beta4.LM))
sqrt(var(beta5.LM))
sqrt(var(beta6.LM))
sqrt(var(beta7.LM))
sqrt(var(beta8.LM))

# Median Centering
cy.median <- y - y.median_matrix
cy <- as.vector(cy.median)

cx1.median <- x1.CO - x1.median_matrix
cx1 <- as.vector(cx1.median)

cx2.median <- x2.HUM - x2.median_matrix
cx2 <- as.vector(cx2.median)

cx3.median <- x3.NO - x3.median_matrix
cx3 <- as.vector(cx3.median)

cx4.median <- x4.NO2 - x4.median_matrix
cx4 <- as.vector(cx4.median)

cx5.median <- x5.NOX - x5.median_matrix
cx5 <- as.vector(cx5.median)

cx6.median <- x6.O3 - x6.median_matrix
cx6 <- as.vector(cx6.median)

cx7.median <- x7.SO2 - x7.median_matrix
cx7 <- as.vector(cx7.median)

cx8.median <- x8.WD - x8.median_matrix
cx8 <- as.vector(cx8.median)

cm.med_gm <- GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy)
```

```
cy.new_gm <- cm.med_gm$coef[2]*cx1 + cm.med_gm$coef[3]*cx2 +
cm.med_gm$coef[4]*cx3 + cm.med_gm$coef[5]*cx4 + cm.med_gm$coef[6]*cx5 +
cm.med_gm$coef[7]*cx6 + cm.med_gm$coef[8]*cx7 + cm.med_gm$coef[9]*cx8
cy.e_gm <- cy-cy.new_gm

for(i in 1:1000){
  cy.s_gm <- sample(cy.e_gm, length(cy.e_gm), replace=T)
  cy.a_gm <- cy.s_gm + cy.new_gm

  beta1.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[2]
  beta2.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[3]
  beta3.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[4]
  beta4.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[5]
  beta5.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[6]
  beta6.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[7]
  beta7.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[8]
  beta8.GM[i] <-
GMM_Reg(cbind(cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8),cy.a_gm)$coef[9]
  }
mean(beta1.GM)
mean(beta2.GM)
mean(beta3.GM)
mean(beta4.GM)
mean(beta5.GM)
mean(beta6.GM)
mean(beta7.GM)
mean(beta8.GM)

cm.med_mm <- lmrob(cy~cx1+cx2+cx3+cx4+cx5+cx6+cx7+cx8, method="SMM")

cy.new_mm <- summary(cm.med_mm)$coefficients[2,1]*cx1 +
```

```
summary(cm.med_mm)$coefficients[3,1]*cx2 +
summary(cm.med_mm)$coefficients[4,1]*cx3 +
summary(cm.med_mm)$coefficients[5,1]*cx4 +
summary(cm.med_mm)$coefficients[6,1]*cx5 +
summary(cm.med_mm)$coefficients[7,1]*cx6 +
summary(cm.med_mm)$coefficients[8,1]*cx7 +
summary(cm.med_mm)$coefficients[9,1]*cx8
cy.e_mm <- cy-cy.new_mm

for(i in 1:1000){
  cy.s_mm <- sample(cy.e_mm, length(cy.e_mm), replace=T)
  cy.a_mm <- cy.s_mm + cy.new_mm

  beta1.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+cx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[2,1]
  beta2.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+cx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[3,1]
  beta3.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+cx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[4,1]
  beta4.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+lx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[5,1]
  beta5.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+lx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[6,1]
  beta6.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+lx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[7,1]
  beta7.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+lx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[8,1]
  beta8.MM[i] <-
summary(rlm(cy.a_mm~cx1+cx2+cx3+lx4+cx5+cx6+cx7+cx8,method="MM"))$co
ef[9,1]
}
```

```
mean(beta1.MM)
mean(beta2.MM)
mean(beta3.MM)
mean(beta4.MM)
mean(beta5.MM)
mean(beta6.MM)
mean(beta7.MM)
mean(beta8.MM)

# MM Centering
MMinit<-(lmrob(ly~lx1+lx2+lx3+lx4+lx5+lx6+lx7+lx8,
method="SMM")$rweights)
w.dm <-  matrix(MMinit,ncol=n.train,nrow=t)

wx1.dm <- w.dm*x1.CO
wx2.dm <- w.dm*x2.HUM
wx3.dm <- w.dm*x3.NO
wx4.dm <- w.dm*x4.NO2
wx5.dm <- w.dm*x5.NOX
wx6.dm <- w.dm*x6.O3
wx7.dm <- w.dm*x7.SO2
wx8.dm <- w.dm*x8.WD

wy.dm <- w.dm*y

mean_x1.MM <- apply(wx1.dm,2,mean)
mean_x1_dm.MM <- t(mean_x1.MM*t(matrix(1,t,n.train)))
mean_cent_x1.MM<- x1.CO - mean_x1_dm.MM
mmx1 <- as.vector(mean_cent_x1.MM)

mean_x2.MM <- apply(wx2.dm,2,mean)
mean_x2_dm.MM <- t(mean_x2.MM*t(matrix(1,t,n.train)))
mean_cent_x2.MM<- x2.HUM - mean_x2_dm.MM
mmx2 <- as.vector(mean_cent_x2.MM)

mean_x3.MM <- apply(wx3.dm,2,mean)
mean_x3_dm.MM <- t(mean_x3.MM*t(matrix(1,t,n.train)))
mean_cent_x3.MM<- x3.NO - mean_x3_dm.MM
```

```
mmx3 <- as.vector(mean_cent_x3.MM)

mean_x4.MM <- apply(wx4.dm,2,mean)
mean_x4_dm.MM <- t(mean_x4.MM*t(matrix(1,t,n.train)))
mean_cent_x4.MM<- x4.NO2 - mean_x4_dm.MM
mmx4 <- as.vector(mean_cent_x4.MM)

mean_x5.MM <- apply(wx5.dm,2,mean)
mean_x5_dm.MM <- t(mean_x5.MM*t(matrix(1,t,n.train)))
mean_cent_x5.MM<- x5.NOX - mean_x5_dm.MM
mmx5 <- as.vector(mean_cent_x5.MM)

mean_x6.MM <- apply(wx6.dm,2,mean)
mean_x6_dm.MM <- t(mean_x6.MM*t(matrix(1,t,n.train)))
mean_cent_x6.MM<- x6.O3 - mean_x6_dm.MM
mmx6 <- as.vector(mean_cent_x6.MM)

mean_x7.MM <- apply(wx7.dm,2,mean)
mean_x7_dm.MM <- t(mean_x7.MM*t(matrix(1,t,n.train)))
mean_cent_x7.MM<- x7.SO2 - mean_x7_dm.MM
mmx7 <- as.vector(mean_cent_x7.MM)

mean_x8.MM <- apply(wx8.dm,2,mean)
mean_x8_dm.MM <- t(mean_x8.MM*t(matrix(1,t,n.train)))
mean_cent_x8.MM<- x8.WD - mean_x8_dm.MM
mmx8 <- as.vector(mean_cent_x8.MM)

mean_y.MM <- apply(wy.dm,2,mean)
mean_y_dm.MM <- t(mean_y.MM*t(matrix(1,t,n.train)))
mean_cent_y.MM<- y - mean_y_dm.MM
mmy <- as.vector(mean_cent_y.MM)

#combine_mm <- cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmy)
#combine_mm.df<-data.frame(combine_mm)
#colnames(combine_mm.df)<-c("X1","X2","X3","X4","X5","X6","X7","Y")

mm.med_gm <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8), mmy)
```

```
mmy.new_gm <- mm.med_gm$coef[2]*mmx1 + mm.med_gm$coef[3]*mmx2 +
mm.med_gm$coef[4]*mmx3 + mm.med_gm$coef[5]*mmx4 +
mm.med_gm$coef[6]*mmx5 + mm.med_gm$coef[7]*mmx6 +
mm.med_gm$coef[8]*mmx7 + mm.med_gm$coef[9]*mmx8
mmy.e_gm <- mmy-mmy.new_gm

for(i in 1:1000){
  mmy.s_gm <- sample(mmy.e_gm, length(mmy.e_gm), replace=T)
  mmy.a_gm <- mmy.s_gm + mmy.new_gm

  beta1.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[2]
  beta2.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[3]
  beta3.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[4]
  beta4.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[5]
  beta5.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[6]
  beta6.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[7]
  beta7.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[8]
  beta8.MM.GM[i] <-
GMM_Reg(cbind(mmx1,mmx2,mmx3,mmx4,mmx5,mmx6,mmx7,mmx8),
mmy.a_gm)$coef[9]
}
mean(beta1.MM.GM)
mean(beta2.MM.GM)
```

```
mean(beta3.MM.GM)
mean(beta4.MM.GM)
mean(beta5.MM.GM)
mean(beta6.MM.GM)
mean(beta7.MM.GM)
mean(beta8.MM.GM)

## MM Centering with MM method
mm.med_mm <-
lmrob(mmy~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx7+mmx8,
method="SMM")

mmy.new_mm <- summary(mm.med_mm)$coefficients[2,1]*mmx1 +
summary(mm.med_mm)$coefficients[3,1]*mmx2 +
summary(mm.med_mm)$coefficients[4,1]*mmx3 +
summary(mm.med_mm)$coefficients[5,1]*mmx4 +
summary(mm.med_mm)$coefficients[6,1]*mmx5 +
summary(mm.med_mm)$coefficients[7,1]*mmx6 +
summary(mm.med_mm)$coefficients[8,1]*mmx7 +
summary(mm.med_mm)$coefficients[9,1]*mmx8
mmy.e_mm <- mmy-mmy.new_mm

for(i in 1:1000){
 mmy.s_mm <- sample(mmy.e_mm, length(mmy.e_mm), replace=T)
 mmy.a_mm <- mmy.s_mm + mmy.new_mm

beta1.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[2,1]
 beta2.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[3,1]
 beta3.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[4,1]
 beta4.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[5,1]
```

```
  beta5.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[6,1]
  beta6.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[7,1]
  beta7.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[8,1]
  beta8.MM.MM[i] <-
coef(summary(rlm(mmy.a_mm~mmx1+mmx2+mmx3+mmx4+mmx5+mmx6+mmx
7+mmx8, method="MM")))[9,1]
}
mean(beta1.MM.MM)
mean(beta2.MM.MM)
mean(beta3.MM.MM)
mean(beta4.MM.MM)
mean(beta5.MM.MM)
mean(beta6.MM.MM)
mean(beta7.MM.MM)
mean(beta8.MM.MM)

#Standard Error
#2. Median Centering
#a) RWGM
sqrt(var(beta1.GM))
sqrt(var(beta2.GM))
sqrt(var(beta3.GM))
sqrt(var(beta4.GM))
sqrt(var(beta5.GM))
sqrt(var(beta6.GM))
sqrt(var(beta7.GM))
sqrt(var(beta8.GM))

#Standard Error
#2. Median Centering
#b) RWMM
sqrt(var(beta1.MM))
```

```
sqrt(var(beta2.MM))
sqrt(var(beta3.MM))
sqrt(var(beta4.MM))
sqrt(var(beta5.MM))
sqrt(var(beta6.MM))
sqrt(var(beta7.MM))
sqrt(var(beta8.MM))

#2. MM Centering
#a) RWGM
sqrt(var(beta1.MM.GM))
sqrt(var(beta2.MM.GM))
sqrt(var(beta3.MM.GM))
sqrt(var(beta4.MM.GM))
sqrt(var(beta5.MM.GM))
sqrt(var(beta6.MM.GM))
sqrt(var(beta7.MM.GM))
sqrt(var(beta8.MM.GM))

#2. MM Centering
#b) RWMM
sqrt(var(beta1.MM.MM))
sqrt(var(beta2.MM.MM))
sqrt(var(beta3.MM.MM))
sqrt(var(beta4.MM.MM))
sqrt(var(beta5.MM.MM))
sqrt(var(beta6.MM.MM))
sqrt(var(beta7.MM.MM))
sqrt(var(beta8.MM.MM))
```

4 Real Data Analysis (For Data Transformation Purposes)

```
#Data in matrix form
y <- matrix(y,t,n.train)

x1.CO <- matrix(x1,t,n.train)
x2.HUM <- matrix(x2,t,n.train)
x3.NO <- matrix(x3,t,n.train)
x4.NO2 <- matrix(x4,t,n.train)
```

```
x5.NOX <- matrix(x5,t,n.train)
x6.O3 <- matrix(x6,t,n.train)
x7.SO2 <- matrix(x7,t,n.train)
x8.WD <- matrix(x8,t,n.train)

#To find mean
y.mean <- apply(y,2,mean)
y.mean_matrix <- t(y.mean*t(matrix(1,t,n.train)))

x1.mean <- apply(x1.CO,2,mean)
x1.mean_matrix <- t(x1.mean*t(matrix(1,t,n.train)))

x2.mean <- apply(x2.HUM,2,mean)
x2.mean_matrix <- t(x2.mean*t(matrix(1,t,n.train)))

x3.mean <- apply(x3.NO,2,mean)
x3.mean_matrix <- t(x3.mean*t(matrix(1,t,n.train)))

x4.mean <- apply(x4.NO2,2,mean)
x4.mean_matrix <- t(x4.mean*t(matrix(1,t,n.train)))

x5.mean <- apply(x5.NOX,2,mean)
x5.mean_matrix <- t(x5.mean*t(matrix(1,t,n.train)))

x6.mean <- apply(x6.O3,2,mean)
x6.mean_matrix <- t(x6.mean*t(matrix(1,t,n.train)))

x7.mean <- apply(x7.SO2,2,mean)
x7.mean_matrix <- t(x7.mean*t(matrix(1,t,n.train)))

x8.mean <- apply(x8.WD,2,mean)
x8.mean_matrix <- t(x7.mean*t(matrix(1,t,n.train)))

#To find median
y.median <- apply(y,2,median)
y.median_matrix <- t(y.median*t(matrix(1,t,n.train)))

x1.median <- apply(x1.CO,2,median)
```

```
x1.median_matrix <- t(x1.median*t(matrix(1,t,n.train)))

x2.median <- apply(x2.HUM,2,median)
x2.median_matrix <- t(x2.median*t(matrix(1,t,n.train)))

x3.median <- apply(x3.NO,2,median)
x3.median_matrix <- t(x3.median*t(matrix(1,t,n.train)))

x4.median <- apply(x4.NO2,2,median)
x4.median_matrix <- t(x4.median*t(matrix(1,t,n.train)))

x5.median <- apply(x5.NOX,2,median)
x5.median_matrix <- t(x5.median*t(matrix(1,t,n.train)))

x6.median <- apply(x6.O3,2,median)
x6.median_matrix <- t(x6.median*t(matrix(1,t,n.train)))

x7.median <- apply(x7.SO2,2,median)
x7.median_matrix <- t(x7.median*t(matrix(1,t,n.train)))

x8.median <- apply(x8.WD,2,median)
x8.median_matrix <- t(x7.median*t(matrix(1,t,n.train)))
```