Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrie & Operationele Research

Shooting S-estimator and cellwise robust M regression estimator: an extensive simulation study of cellwise robust techniques

Amber Cuijpers (577305ac)

Supervisor:	Archimbaud, A.
Second assessor:	Gruber, K.
Date final version:	2nd July 2023

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Contents

Abstract

The cellwise robust M regression estimator (CRM) introduced by Filzmoser et al. (2020), and the shooting S-estimator of Öllerer et al. (2016) are among the first estimators designed to deal with cellwise outliers. In this research, these two estimators are compared with each other, as well as their casewise counterparts MM-regression and S-regression. Besides that, CRM with S-regression as initial step (CRSM) is also examined. In a simulation study, CRM has an equally predictive performance and bias as MM-regression. For a higher amount of variables, CRM outperforms CRSM. However, CRSM has the highest recall, but the longest execution time. Shooting S outperforms S-regression and the other cellwise robust methods in terms of low bias when the number of variables is high. Nevertheless, the imputation accuracy of shooting S is considerably lower compared to CRM, which is also shown by an empirical application about cars.

1 Introduction

The least squares estimator is commonly used for regression analysis, although this estimator may not be optimal when the data deviates from the normal distribution assumptions. Most of the time, when a multivariate observation deviates from the model, the entire observation is identified as an outlier. This means that the whole row, thus all variables corresponding to that observation are seen as outlier, and this is called a casewise outlier. However, it can also happen that only one single variable contributes to the outlyingness of the observation. In such a situation, it is a loss of information to treat all variables of that observation as outlier. It could then be more useful to look at individual variables as outliers, which are called cellwise outliers.

Filzmoser et al. (2020) introduces the cellwise robust M regression estimator (CRM), which is a new estimator for regression analysis. Unlike other robust methods that treat an entire observation as an outlier when it deviates from the model, CRM looks at cellwise outliers. It starts with weights derived from robust estimates, and then iteratively detects and downweights the cells that contribute most to outlyingness.

In this research, we aim to extend the work of Filzmoser et al. (2020). As highlighted in their paper, there is still a lot of research to be done in this domain, since cellwise outliers is a relatively new field compared to casewise outliers. Throughout a simulation study they observed that CRM outperforms Ordinary Least Squares (OLS) regression, OLS regression with a DDC-imputed matrix, and the casewise regression methods MM regression (Yohai, 1987) and MM regression with a DDC-imputed matrix, particularly in cases with a low fraction of contamination. Conversely, for a higher fraction of contamination DDC-MM and MM outperformed the other methods.

Here, DDC stands for Detecting Deviating Data Cells (DDC), which is a method introduced by Rousseeuw and Bossche (2018). DDC is made to detect deviating data cells in a multivariate sample. Remarkably, it has the characteristic of not imposing restrictions on the number of clean rows, and remains reliable even when more than 50% of the cases contain outlying cells.

Building upon the suggestion of the paper, S-regression is also a valid method to consider in this context (Salibian-Barrera and Yohai, 2006). Therefore, the objective of this research is to expand the paper by examining two additional cellwise robust techniques. Firstly, the

CRM regression estimator with S-regression as initial estimator is considered, instead of MM regression. Secondly, the shooting S-estimator will be introduced, as proposed by Öllerer et al. (2016).

Consequently, the research question is formulated as follows:

How does the shooting S-estimator perform compared to the cellwise robust M regression estimator, and the modified cellwise robust M regression estimator with S-regression as initial step, in terms of the robustness and performance when evaluated through an extended simulation study?

To answer this research question, we will have a simulation study based on the one of Filzmoser et al. (2020). However, in addition to the regression methods used in that study, we will include the cellwise robust M regression estimator with S-regression as initial estimator (CRSM), and the shooting S-estimator. Besides that, different simulation settings are considered, namely different amount of variables, varying magnitudes of contamination and different case-and cellwise contamination percentages. It is found that CRM has an equally predictive performance and bias compared to its casewise counterpart MM-regression, and for a lower amount of variables it also outperforms CRSM in terms of precision and predictive performance. However, CRSM has the highest recall, but longest average execution time. The imputation accuracy of shooting S is the lowest.

The performance of the different methods is tested on a real-world Auto dataset, which consists of 8 predictor variables and 74 sales of 1987 vintage cars in the United States, just as used in Öllerer et al. (2016). Here, it is also found that the imputation of shooting S performs quite bad compared to CRM and DDC.

In Section 2, we review the literature, and Section 3 explains the cellwise robust models. Next, Section 4 explains the criteria used to evaluate the performances and discusses the results of the simulations. In Section 5 an empirical application is used to evaluate the performances of the models. Lastly, Section 6 mentions the conclusions, limitations and further research.

2 Literature Review

Numerous robust methods have been proposed to address the issue of casewise deviations (Rousseeuw and Leroy, 2005; Alma, 2011; Huber, 2011; Maronna et al., 2019). As mentioned by Filzmoser et al. (2020), these casewise deviations can arise due to a fraction ϵ of the data being generated from a different distribution, indicating the presence of outliers, or when the data satisfies the linear model with a non-normal error term, such as a Cauchy or Student's t.

Casewise robust methods treat an entire row as contaminated in the multivariate case, meaning that an entire observation is downweighted. This is useful when an observation is outlying compared to the other observations, meaning that most of the predictor variables corresponding to that row are outlying. Nevertheless, in certain cases, such deviations of an observation may be attributed to only a small subset of predictor variables. Treating every variable of the observation as an outlier can lead to a loss of valuable information and may not reflect the reality. In such cases, it is preferable to detect outliers on a cellwise basis rather than casewise. This approach allows for optimal utilization of the non-contaminated part of the dataset.

The field of cellwise robust techniques is still in its early days, where there is a lot to be discovered. Some more recent work has investigated cellwise outliers. For instance, Hubert et al. (2019) proposed a new PCA method that handles both cellwise outliers and missing values. Štefelová et al. (2021) investigated a robust regression method that handles both cellwise and rowwise outliers, with compositional and real-valued explanatory variables.

Moreover, Filzmoser et al. (2020) introduces the cellwise robust M regression estimator (CRM). This method handles cellwise contamination by starting with a robust estimate obtained from MM regression, which handles vertical outliers (outliers in the dependent/y-variable), and leverage points (outliers in an explanatory variable). Subsequently, it uses an iteratively reweighted least squares procedure with SPADIMO incorporated in each iteration (Debruyne et al., 2019). SPADIMO is a procedure that is able to detect the cells that contribute the most to the outlyingness of an observation, and only these are downweighted instead of the whole row. In the case where only a few variables are outlying, this method is more efficient than casewise robust estimators, since it only downweights the necessary cells.

The shooting S-estimator (Öllerer et al., 2016) is also a cellwise robust regression estimator, which uses a combination of the S-regression and the coordinate descent algorithm, which performs a regression variable by variable. This estimator addresses cellwise deviations, while the original S-estimator only considers casewise deviations.

Firstly, it is of interest to investigate the performance of these two cellwise robust estimators in relation to their corresponding casewise robust methods. As mentioned earlier, shooting S uses casewise S-regression, while CRM utilizes MM regression. Therefore, it is interesting to compare performances of S-regression with shooting S and MM regression with CRM. Hence, our first sub-question is as follows:

S1: *How does the shooting S-estimator perform compared to the S-estimator and CRM compared to the MM-estimator?*

We first start with this sub-question, to focus on implementing the shooting S-estimator and CRM regression estimator. Both of these estimators handle cellwise contamination, in contrast to the S-regression and MM-regression estimators, which solely deal with casewase contamination. The hypothesis is that the shooting S-estimator outperforms the S-estimator when the majority of observations are only contaminated in a small number of variables (Öllerer et al., 2016), same as for the CRM regression estimator.

Since this research mainly focuses on extending the research of Filzmoser et al. (2020), the next question that arises will be:

S2: *How does the shooting S-estimator perform compared to the CRM regression estimator within the framework of the identical simulation study as Filzmoser et al. (2020)?*

This sub-question evaluates the performance of the shooting S-estimator in contrast to the CRM regression estimator within the context of the simulation study conducted by Filzmoser et al. (2020), where all parameters have fixed values. By exploring this comparison, valuable insights can be gained regarding the robustness and performance of these two estimators under similar conditions.

After having evaluated these two regression estimators, it becomes interesting to compare them with the other regression methods demonstrated in the simulation study of Filzmoser et al. (2020). Besides that, an additional regression method of interest, is the cellwise robust M regression estimator with as initial step S regression (CRSM). The only difference from CRM is that S regression is used as the starting estimate instead of MM regression. This alternative choice is also recommended by Filzmoser et al. (2020) as a valid option. This leads to our third sub-question:

S3: *How does the cellwise robust M regression estimator with S regression for the initial estimator perform compared to the shooting S-estimator and the CRM regression estimator with MM regression for the initial estimator in the context of the same simulation study as Filzmoser et al. (2020)?*

These sub-questions examine the performance during the same simulation settings, while it can also be important to look at the performances of these estimators when the simulation settings change, especially when the number of variables increase or the fraction of contamination changes. Therefore, our next and final sub-question is:

S4: *When looking at different simulation settings, which robust regression estimator performs best?*

Different settings include varying the magnitude of contamination, increasing the amount of variables, increasing the percentage of casewise contamination and the percentage of cellwise contamination. The magnitude of contamination and percentage of casewise outliers are considered in the same way as is done in Filzmoser et al. (2020).

The outcomes of this comparative analysis are relevant for several reasons. Firstly, previous studies have not conducted research comparing these three estimators, meaning that this study can provide new insights. By investigating the robustness and performance of these estimators, and by using an extensive simulation, we can gain a better understanding of which estimators perform better in which scenarios or if one estimator outperforms the others. Additionally, as mentioned in Section 1 the field of cellwise robust techniques itself is still relatively unexplored, meaning that new contributions will improve our knowledge of this specific domain. Secondly, in practice robust regression is widely used, and detecting the presence of cellwise outliers becomes increasingly important. For optimal portfolio allocation, Avagyan and Mei (2022) demonstrated that applying robust techniques when there are cellwise outliers improves the out-of-sample performance. Furthermore, Segaert et al. (2019) researched a robust statistical method to identify target genes and outliers in triple-negative breast cancer data. Lastly, by comparing the shooting S-estimator and CRM estimators, a deeper understanding can be gained about how they respond to different types of contamination, number of variables and magnitudes of contamination.

3 Methodology

This Section provides a brief explanation of the Cellwise robust M regression (CRM) estimator, an adjusted CRM estimator and the shooting S-estimator. Lastly, the simulation settings of this research will be discussed.

3.1 Cellwise robust M regression estimator

The cellwise robust M (CRM) regression estimator is the first estimator that provides both a map of deviating cells, and regression coefficients that are robust against cellwise and casewise outliers (Filzmoser et al., 2020). This method is especially useful when only a few predictor variables of an observation are outlying.

Within the class of robust estimators, MM estimators are performing well in terms of robustness-efficiency trade-off, making them widely adopted Abonazel and Rabie (2019). Consequently, CRM uses MM estimators as an initial step to ensure robustness in handling leverage points. However, MM estimators are not the most efficient nor robust against vertical outliers. Therefore, an iteratively reweighted least squares (IRLS) routine is used, with the MM estimator as the starting point.

Prior to starting with the MM regression estimator, as preprocessing step the data should be centered and scaled. In this research, centering is done with the L_1 median, and scaling with the Q_n scale estimator (Rousseeuw and Croux, 1993). Then, with the MM regression, some observations are flagged as outliers. This is when:

$$\frac{|r_i|}{cmed_j|r_j|} > z_{0.95} \tag{1}$$

Here, r_i represents the i'th residual coming from the initial MM estimator: $r_i = y_i - \mathbf{x}_i^T\hat{\beta}$, with $\mathbf{x}_i$ the i'th row of the data matrix, and $\hat{\beta}$ the least squares estimator. Besides that, c = 1.4826 for consistency between the Mean Absolute Deviation and standard deviation, *med* is the median and lastly, $z_{0.95}$ is the 95% quantile of the standard normal distribution.

These flagged observations can be truly casewise outliers. However, in some cases only some variables of an observations are outlying, and the other variables are not. In order to investigate which cells contribute most to the outlyingness of an observation, Sparse Directions of Maximal Outlyingness (SPADIMO) is applied in each iteration of IRLS (Debruyne et al., 2019). SPADIMO is a method that identifies the variables that contribute most to the outlyingness of the observation, by estimating the direction where there is Maximal Outlyingness, which is the norm of the solution of least squares regression. When SPADIMO detects that only some cells are outlying, the values of these cells are imputed.

The imputation of the cells that are detected is as follows:

1. An observation that is detected as outlier is notated as $\mathbf{x_i}$. In total there are n cases and p predictor variables, thus i can not be more than n.

2. $\mathcal{C}$ is the set of all cellwise outliers detected by SPADIMO for one single $\mathbf{x_i}$. Since there are p predictor variables, this se, denoted by q, is smaller than p.

3. To be robust against vertical outliers, the Hampel function is used (Filzmoser et al., 2020).

This reweighting representation, with r as the regression residual, is given by:

$$w_H(r) = \begin{cases} 1, & \text{if } |r| \leq Q_1 \\ \frac{Q_1}{|r|}, & \text{if } Q_1 < |r| \leq Q_2 \\ \frac{Q_3 - r}{Q_3 - Q_2} \frac{Q_1}{|r|}, & \text{if } Q_2 < |r| \leq Q_3 \\ 0, & \text{if } Q_3 < |r| \end{cases} \tag{2}$$

In this representation, Q_1, Q_2 and Q_3 are assumed to be standard normally distributed with the values 0.95, 0.975 and 0.999. This assumption is logical, since the data is also already standardized.
Case weights of the Hampel weight function (ω_i) can also be calculated with c = 1.4826

$$\omega_i = w_H\left(\frac{|\tilde{r}_j|}{cmed_j|\tilde{r}_j|}\right) \tag{3}$$

4. Only among the observations $\mathbf{x_j}$ where $\omega_j = 1$, and only among all predictor variables that are not detected as cellwise outliers, the two closest neighbours of $\mathbf{x_i}$ have to be detected. These two closest neigbours are denoted as $\mathbf{x_{k_1}}$ and $\mathbf{x_{k_2}}$. These can be found with the 'FNN' package in R, which has the function get.knn, an algorithm that searches for the k-nearest neighbours (Beygelzimer et al., 2019).

5. The cellwise outliers are then imputed with the column means of these two closest neighbours:

$$\tilde{x}_{iq} = \frac{x_{k_{1q}} + x_{k_{2q}}}{2}, \text{with } q \in \mathcal{C} \tag{4}$$

In this way, the cellwise outliers are actually handled as missing values, since they are imputed by taking the column means of the two closest neighbours.
In each iteration, the estimates from the previous iteration are used for the least square regression estimates. This means, that in the first iteration, the initial estimators are used, and in the subsequent iterations, the updated estimators. In each step, residuals are calculated and SPADIMO is applied to the observations flagged as outliers. If the conclusion is that not all variables of an observation contribute to the outlyingness, then the values are imputed. With these newly imputed data matrix, the residuals and the case weights with the Hampel weight function are calculated. The imputed data can then be updated for the next iteration by multiplying the $\mathbf{X}$ and $\mathbf{Y}$ matrix with a diagonal matrix having the case weights as diagonal elements.

The IRLS algorithm is stopped when the mean absolute difference between the previous and current regression estimates is smaller than a tolerance bound.

The algorithm of the steps described above for IRLS can also be found in Filzmoser et al. (2020).

3.2 Cellwise robust M regression estimator with S-regression

According to Filzmoser et al. (2020) S-regression is considered a valid alternative to MM-regression for obtaining the initial estimators. Therefore, as an extension, S-regression will be included as an additional method to compare its performance with CRM and the other estimation methods, as mentioned in Section 4

The only difference with CRM lies in the initial step, where S regression is performed instead of MM-regression. This S-regression estimator is robust against leverage points, but not against vertical outliers, just as the MM-regression estimator. Hence, the IRLS procedure will remain unchanged, with S estimator as starting point.

3.3 Shooting S-estimator

The shooting S-estimator, proposed by (Öllerer et al., 2016) is another cellwise robust method, just as CRM. It combines the coordinate descent algorithm (Fu, 1998), also known as the 'shooting algorithm' with S-regression (Susanti et al., 2014). However, similar to MM-regression for CRM, S-regression handles only casewise outliers and therefore needs to be combined with an algorithm to ensure robustness against vertical outliers.

The shooting algorithm updates an lasso coefficient in every iteration until it converges to the lasso estimate (Tseng, 2001). When updating one lasso coefficient, $\hat{\beta}_j$, the other coefficients $\hat{\beta}_k$ are kept fixed, where k $\neq$ j and j = 1,...,p.

$$\hat{\beta}_{j,Lasso} = \underset{\beta_j \in \mathbb{R}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\left(y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k \right) - x_{ij} \beta_j \right)^2 + 2\lambda |\beta_j| \tag{5}$$

Here, y_i is the dependent variable for the i'th row, and x_{ij} is the j'th cell in the i'th row. The first term is the least squares error term, and the second term is the L1 penalty term. λ is the shrinkage parameter, where a larger λ shrinks more coefficients towards zero (Tibshirani, 1996).

Only, this lasso estimate is non-robust, so it is replaced with a S-estimate to obtain a robust shooting S-estimator (Öllerer et al., 2016). The advantage of combining S-regression with the shooting algorithm is that because of this algorithm all cells can be weighted differently.

The new response for the shooting S-estimator, instead of for the shooting estimator is

$$\tilde{y}_i^{(j)} = y_i - \sum_{k \neq j} \tilde{x}_{ik} \hat{\beta}_k, \text{with } \tilde{x}_{ik} = w_{ik} x_{ik} + (1 - w_{ik}) \hat{x}_{ik} \tag{6}$$

Here, w_{ik} determines the outlyingness of one cell x_{ik}, by scaling by a robust residual scale $\hat{\sigma}_k$. The interval is only between [0,1]. The definition is as follows:

$$w_{ik} = w \left(\frac{|\tilde{y}_i^k - x_{ik} \hat{\beta}_k|}{\hat{\sigma}_k} \right) \tag{7}$$

By using hard rejection, with $w(r) = 1$ if $r \leq c$ and 0 otherwise, an observation is assigned weight 1 when it is not detected as outlier, and weight 0 when it is flagged as an outlier. Here, the cut-off value c = 3, thus not more than 0.3% are expected to be flagged as outlier. In the case of an outlying observation, thus weight 0, $\tilde{x}_{ik}$ equals

$$\hat{x}_{ik} = \frac{\tilde{y}_i^{(k)}}{\hat{\beta}_k} \tag{8}$$

In the case of weight 1, it is assumed that the observation is not outlying, and that the observed value can be kept as coefficient.

The shooting S-estimator looks like the one in Equation 5, but now with the S-regression estimator instead of the lasso estimator Öllerer et al. (2016).

$$\hat{\beta}_j = argmin_{\beta \in \mathbb{R}} \hat{\theta}_j(\beta) \tag{9}$$

with $\hat{\theta}_j(\beta)$ as solution s of

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{\widetilde{y_i}^{(j)} - x_{ij}\beta}{s}\right) = \sigma \tag{10}$$

where $\sigma = \mathbb{E}[\rho(Z)]$, with $Z \sim \mathcal{N}(0,1)$. A larger value of σ results in less efficiency but a higher breakdown point (Rousseeuw and Leroy, 2005).

For the choice of the ρ function, Öllerer et al. (2016) mentions the skipped Huber and Tukey'biweight, where both are robust to outliers. However, there is a difference between them, namely that Tukey's biweight can better weaken the effect of outliers. Observations far from the center consistently contribute the same amount to the loss function (Chen, 2020). Therefore, we first implement the Tukey's biweight function:

$$\rho_{BI}(z) = \begin{cases} \frac{k_{BI}^2}{6}\left(1 - \left(1 - \left(\frac{z}{k_{BI}}\right)^2\right)^3\right), & \text{if } |z| \leq k_{BI} \\ \frac{k_{BI}^2}{6}, & \text{if } |z| > k_{BI} \end{cases} \tag{11}$$

Here, z represents the standardized residuals, and k_{BI} is a cut-off value for when the function becomes only a constant.

For further research, it can be interesting to look at the skipped Huber as well, as mentioned in Section 4:

$$\rho_{skH}(z) = \begin{cases} \frac{1}{2}z^2, & \text{if } |z| \leq k_{skH} \\ \frac{k_{skH}^2}{2}, & \text{if } |z| > k_{skH} \end{cases} \tag{12}$$

The shooting S-estimator algorithm described above an also be found in Öllerer et al. (2016).

4 Simulation study

This research focuses on a simulation study. The simulation settings from Hoppner (2020) are used, which are based on the simulations of Filzmoser et al. (2020) and made available on GitHub. These settings will first be replicated for the first sub-questions, and then some settings will be changed. This repository also contains the R package crmREG, for the cellwise robust M regression. For the shooting S-estimator the implementation of Aalfons (2019) is used. A short description of the code can be found in Appendix A. The R-version is 4.2.2.

4.1 Evaluation criteria and simulation settings

As input, for CRM, CRSM and shooting S the same simulation settings as Filzmoser et al. (2020) are used. This includes as design matrix the design matrix with contamination, as can be found in Hoppner (2020). The numerical tolerance for convergence was set to 0.01, and the maximum number of iterations in coordinate descent loop was set to 100. For the shooting algorithm, the method name was set to ”biweight” with a default value of 3.420. Besides that, the simulation consisted of 400 cases, 50 predictor variables, a percentage of 5% casewise outliers, and a percentage of 10% cellwise outliers for each casewise outlier. The simulation was repeated for 50 times.

The four evaluation criteria as mentioned in Filzmoser et al. (2020) will also stay the same, since it is still interesting to compare relative performances and quality of identification of cellwise outliers.

The first evaluation criteria is the Mean Squared Error of Prediction (MSEP), calculated over the set of uncontaminated cases:

$$MSEP = \frac{1}{n_{clean}} \sum_{i \in I} (\hat{y}_i - y_i)^2 \tag{13}$$

Here, I represents the indices of uncontaminated cases, and n_{clean} denotes the number of uncontaminated cases. The MSEP allows us to compare the predictive performances of the different methods.

The Mean Absolute Error (MAE) measures the difference between estimated coefficients and true values. This measure is chosen, since it also is robust to extreme outliers as it takes the absolute difference instead of the squared difference. Here, p represents the number of variables.

$$MAE = \frac{1}{p} \sum_{j=1}^{p} |\hat{\beta}_j - \beta_j| \tag{14}$$

Thus, while MAE is commonly used to asses bias in regression coefficients, MSEP focuses on the predictive performance.

The Root Mean Squared Error of Imputation (RMSEI) measures the difference between the imputed values and the true values. The imputed matrix is denoted as X^{imp}, and the simulated uncontaminated matrix as X, and the number of cases is denoted with n:

$$RMSEI(\mathbf{X}^{imp}, \mathbf{X}) = \sqrt{\frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} \left(x_{ij}^{imp} - x_{ij} \right)^2} \tag{15}$$

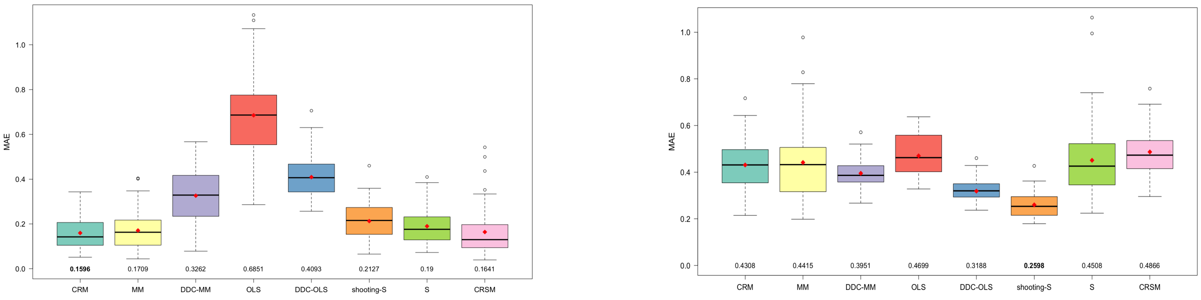
The last evaluation measure is about the ability of methods to identify cellwise outliers. This can assessed using two metrics: *recall*, which is the proportion of cellwise outliers that have been correctly detected as outliers, and *precision*, which is the proportion of flagged outliers that are actually cellwise outliers.

During the simulations, the methods will be compared against other (robust) regression estimators. The following regressions methods will be used in this research: OLS regression, OLS regression with a DDC-imputed matrix, MM regression, MM regression with a DDC-imputed matrix, CRM regression (with MM regression as starting estimate), S regression, shooting S and CRM regression (with S regression as starting estimate). For the latter, constructing the cellwise robust S regression estimator is done by applying S regression on the original, uncontaminated observations, to obtain the initial estimator $\hat{\beta}$, instead of using MM regression. The subsequent steps, as described in Filzmoser et al. (2020) remain the same, including running the IRLS algorithm, applying SPADIMO, imputing values in outlying variables, and using the Hampel weight function.

4.2 Evaluation of CRM, shooting S and CRSM

Firstly, CRM is compared with MM-regression and shooting S with S-regression. In Figure 1, it can be observed that there is almost no difference between CRM and MM-regression, which aligns with the findings of Filzmoser et al. (2020). However, CRM is a better option than MM-regression when there are cellwise outliers, as indicated by the lower bias in Figures 1a and 1b for both 50 and 200 variables. On the other hand, when there are only 50 variables, shooting S exhibits slightly higher bias compared to S regression, while shooting S demonstrates the lowest bias of all for 200 variables. This can be due to the fact that with a larger number of variables, the ability to detect cellwise outliers is better, whereas for fewer variables, the outliers have a larger impact on the variables and the accuracy.

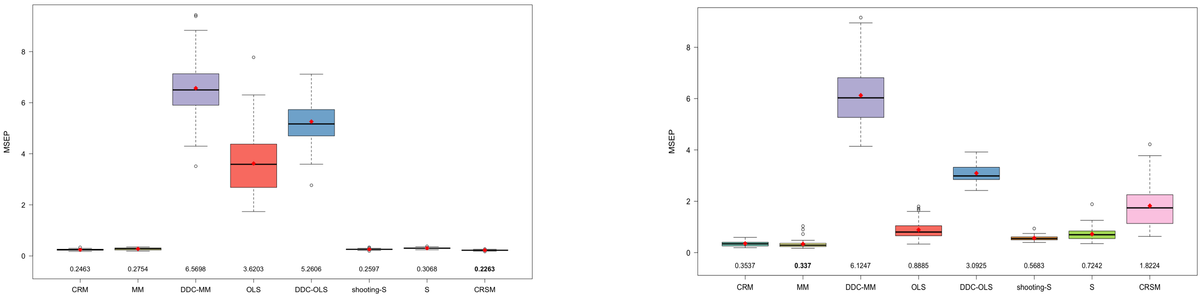
Figure 2 illustrates that there is again almost no difference between CRM and MM-regression in terms of MSEP, where a possible explanation is that they are both robust and designed to handle outliers. Conversely, shooting S generally performs slightly better than S-regression when looking at these two different amount of predictor variables.



(a) MAE when the amount of predictor variables is set to 50

(b) MAE when the amount of predictor variables is set to 200

Figure 1: Comparison of MAE for 50 and 200 predictor variables



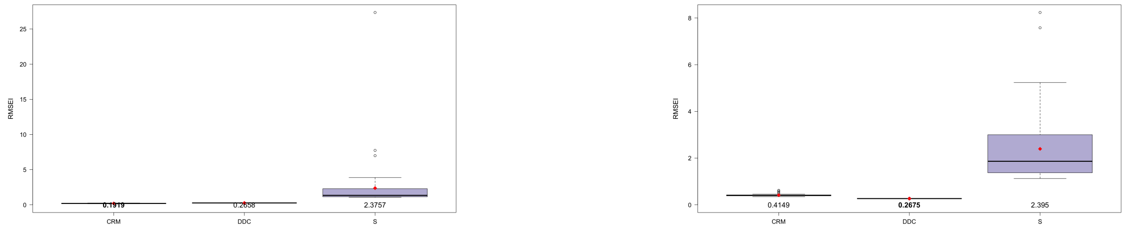
(a) MSEP when the amount of predictor variables is set to 50

(b) MSEP when the amount of predictor variables is set to 200

Figure 2: Comparison of MSEP for 50 and 200 predictor variables

To compare the performances of shooting S, CRM and CRSM with each other and the other robust methods, the simulation settings of Filzmoser et al. (2020) are used, as described in Section 4.1. Within this framework, we observe in Figure 1a and 2a that CRM performs better than shooting S in terms of bias assessment, while the difference between these two methods in predictive performance is minimal. However, when there are 200 predictor variables, shooting S outperforms all other methods in terms of a low bias, even surpassing CRM. When looking at CRSM, the MAE and MSEP of this extended method perform quite the same as CRM itself, with the lowest MSEP among all methods. OLS consistently has a relatively low performance in both MAE and MSEP, while DDC-MM and DDC-OLS are not the best methods, but their performance remain relatively stable.

Moving on to Figure 3 it can be seen that for a small number of variables, CRM performs the best in terms of imputation. As expected, consistent with Filzmoser et al. (2020), CRM outperforms DDC in terms of imputation accuracy, but it is also much better than the imputation in shooting S. This means that the imputed values of CRM have smaller deviations from the true values, and therefore provide more accurate estimates of ”missing” values. When the predictor variables are set to 200, DDC and CRM perform quite the same with DDC being a little better, since its RMSEI stays almost at the same level and does not increase much. On the other hand, the RMSEI of shooting S has the poorest performance, meaning that it is not good at imputation compared to DDC and CRM.

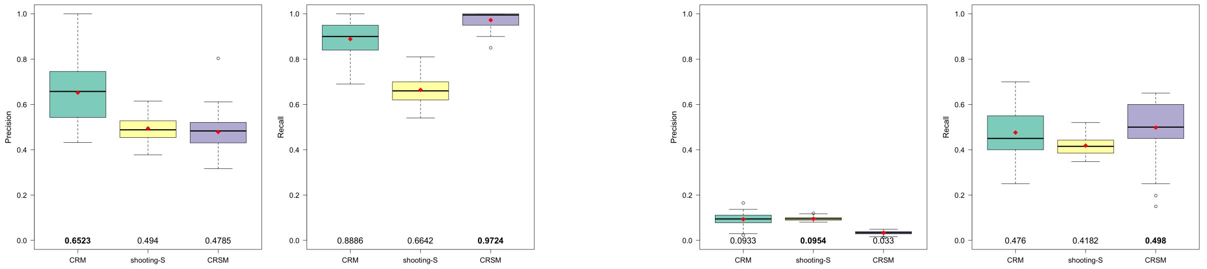


(a) RMSEI when the amount of predictor variables is set to 50

(b) RMSEI when the amount of predictor variables is set to 200

Figure 3: Comparison of RMSEI for 50 and 200 predictor variables

When looking at the precision and recall in Figure 4, CRSM has for both amount of predictor variables the best recall. In terms of precision, CRM is better for 50 predictor variables, while the difference between CRM and shooting S is marginal for 200 variables. Notably, there is a trade-off between precision and recall (Buckland and Gey, 1994). However, the precision and recall are both relatively high for CRM, consistent with the results of Filzmoser et al. (2020). This indicates that CRM detects more of the cellwise outliers that are present, but also flags outliers that are true cellwise outliers. It looks like CRSM does have this trade-off, with lower precision and higher recall.



(a) Precision and recall when the amount of predictor variables is set to 50

(b) Precision and recall when the amount of predictor variables is set to 200

Figure 4: Comparison of precision and recall for 50 and 200 predictor variables

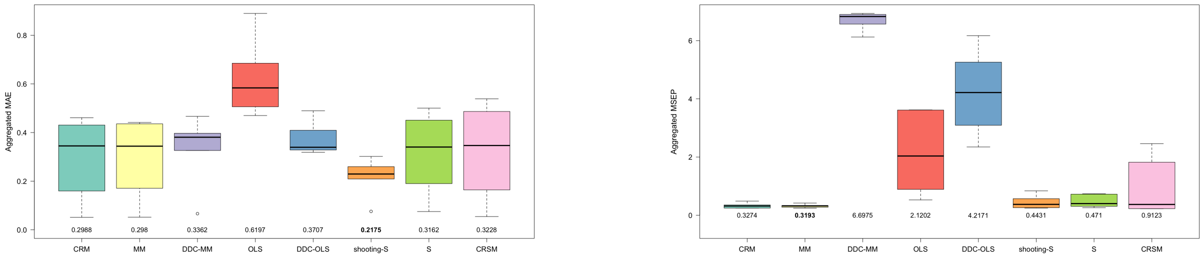
The average execution time of CRSM (25.81 seconds) is substantially longer compared to shooting S (8.99 seconds) and CRM (13.44 seconds). This can be due to the inefficiency of using S-regression as initial step. The lmrob function itself, coming from the package "robustbase" (Maechler et al., 2023), acknowledges that S-regression (lmrob.S) is not efficient, so it is possible that the combination of CRM with S-regression slows down the computation time. In the Appendix B.1 the Figures of the average execution time can be found. When one does not care about the execution time, and wants a high recall, CRSM is recommended. However, if the focus is on controlling bias, achieving high predictive power, or attaining high precision, CRM or shooting S are preferred over CRSM.

4.3 Comparison between amount of predictor variables

As extension of the simulation of Filzmoser et al. (2020), we consider varying the number of predictor variables to 10, 50, 100, 150, 200 and 250, where the other simulation settings remain unchanged. As Filzmoser et al. (2020) also found, CRM outperforms DDC-regressions in detecting and imputing cells that are also cellwise outliers in reality. Most likely this will also be the case for a larger amount of variables. However, the shooting S-estimator may potentially perform even better when the amount of cases stay the same. Öllerer et al. (2016) mentions that shooting S can also be applied to sample sizes even if n < p, indicating its ability to deal with a large amount of predictor variables. It should be noted that due to SPADIMO, CRM/CRSM could not handle the case of 300 variables, since there was no positive definite matrix and SPADIMO is unable to handle such cases.

In Figure 5a shooting S clearly outperforms all other methods in general when aggregating over the amount of variables in terms of maintaining a low bias. Notably, there is almost no difference between CRM and MM, and CRSM even performs slightly worse than S-regression itself. Figure 5b shows that CRM and MM have the best predictive performance, followed by shooting S and S-regression. Once again, CRSM performs even worse than the casewise robust methods MM- and S-regression.

In terms of biases and predictive accuracy, CRSM is not the best cellwise robust method. However, in terms of precision and recall, in Figure 6 the precision is better than shooting S and it even has the best recall of all three cellwise robust methods, consistent with the findings in Subsection 4.2. The aggregated RMSEI and average execution time also yield similar results and can be found in Appendix B.2.1. When considering aggregation, DDC outperforms the imputation techniques of CRM and shooting S, while CRM has a higher predictive performance and lower bias in regression coefficients.



(a) Aggregated MAE

(b) Aggregated MSEP

Figure 5: Aggregated MAE (a) and MSEP (b) when the amount of predictor variables are 10, 50, 100, 150, 200 and 250

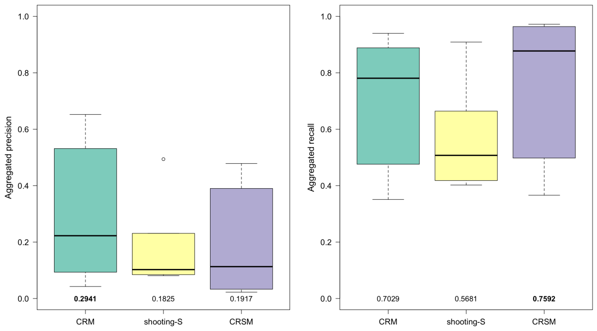
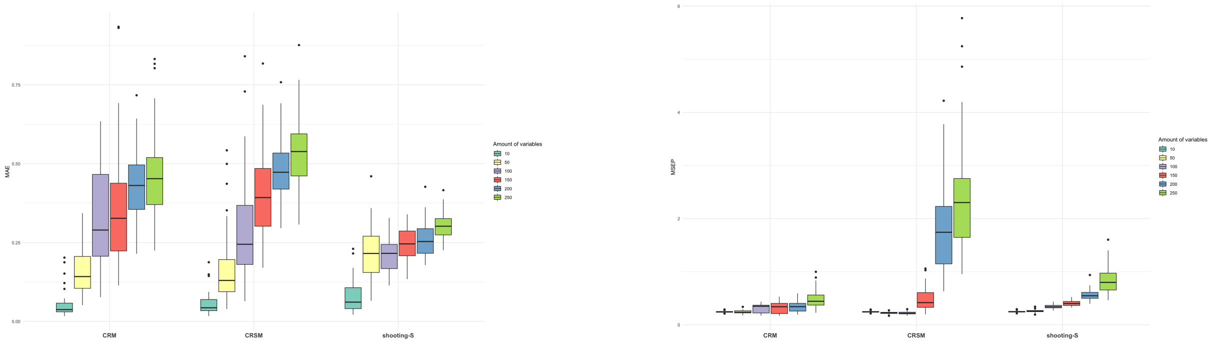


Figure 6: Aggregated precision (left) and recall (right) when the amount of predictor variables are 10, 50, 100, 150, 200 and 250

In the Appendix B.2.2 the boxplots of all robust regression methods for all predictor variables are presented. This allows us to focus on the three cellwise robust methods: CRM, CRSM and shooting S, which have the main focus of our research.

When looking at the grouped boxplot where the number of variables forms different subgroups, some methods are preferred for a low amount of variables, and others for a higher amount. This can also be seen in Figure 7a, where CRM exhibits the lowest bias, closely followed by CRSM, for 10 and 50 variables. After that, for a higher amount of variables, shooting S becomes more effective in assessing biases. For the predictive accuracy, in Figure 7b, CRSM outperforms the other methods until 100 variables. However, for higher amounts, CRM is better. While shooting S surpasses CRSM for a larger number of variables, CRM still remains slightly better.

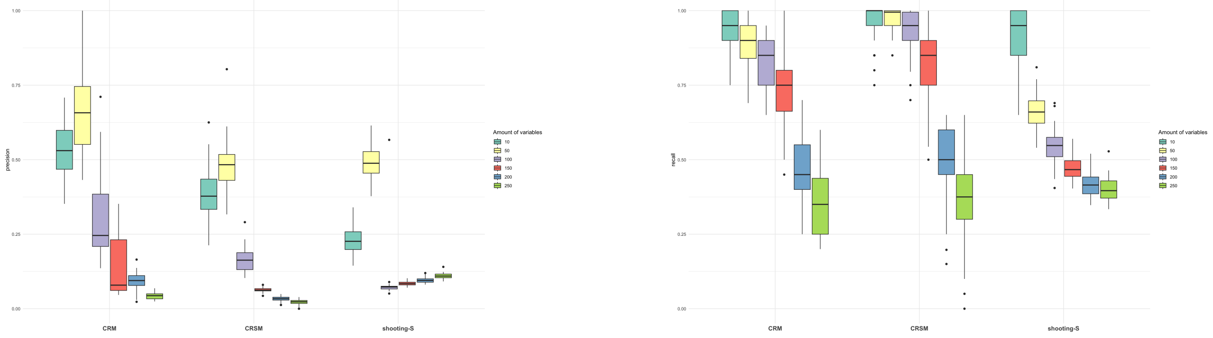


(a) MAE

(b) MSEP

Figure 7: MAE (a) and MSEP (b) for the predictor variables 10, 50, 100, 150, 200 and 250

For all three methods, CRM, CRSM and shooting S, the precision drops after 50 variables as illustrated in Figure 8. CRM is still the best option until 150 variables. After that, shooting S performs better in precision. For recall, CRSM has the highest rate until 200 variables. Although shooting S slightly outperforms CRSM for 250 variables, CRSM generally remains the preferred option when recall is important.



(a) Precision

(b) Recall

Figure 8: Precision (a) and recall (b) for the predictor variables 10, 50, 100, 150, 200 and 250

The amount of predictor variables does not affect the comparison of RMSEI and average execution time. The conclusion remains the same as in Figure 3 and 16, namely that CRM imputation outperforms the imputation method for shooting S, and that shooting S has the fastest execution time, followed by CRM, while CRSM has the slowest. The Figures of the RMSEI and average execution time are in Appendix B.2.3.

4.4 Comparison between magnitudes of contamination

Another setting to change, is the magnitude of contamination, as also done in Filzmoser et al. (2020). There, the contaminated matrix for cellwise contamination is constructed as follows:

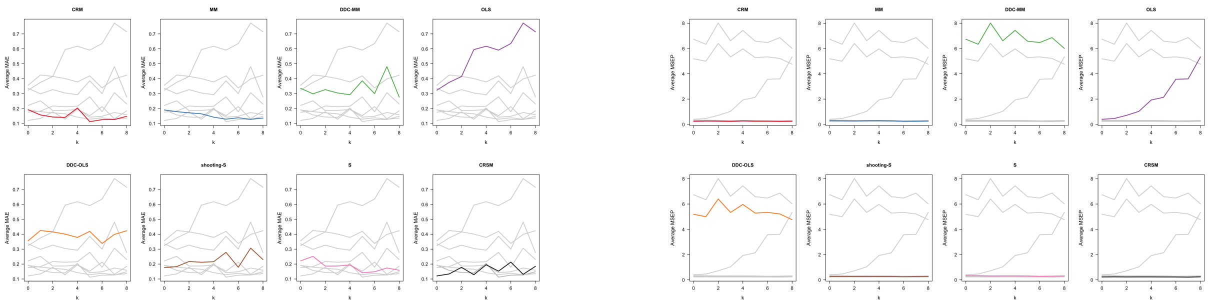
$$x_{ij}^c = \overline{x}_j + ks_j + e \tag{16}$$

with s_j being the standard deviation of variable j

$$s_j = \sqrt{\frac{1}{n-1}\sum_{l=1}^{n}(x_{lj} - \overline{x}_j)^2} \tag{17}$$

Here, $\overline{x}_j$ is the mean value, n is the amount of cases and e is the error of a standard normal distribution. In previous simulations, k is fixed to 6. Increasing this k, would mean that cells are getting more extreme. In this research, k is varied from 0 until 8 and the number of simulations for each k is set to 10, just like Filzmoser et al. (2020). These same results, when fixing the amount of predictor variables, are obtained. This means that CRM performs better for a higher k, and OLS performs only good for a small k, which can be explained by the fact that OLS can not deal good with outliers, while CRM is designed to handle cellwise outliers which are deviating too much from the model.

In Figure 9a, CRM, MM and S all three decline slightly in terms of MAE when the magnitude of contamination increases. Shooting S and CRSM are also low, but deviating more upwards and downwards between different magnitudes. DDC-OLS and DDC-MM stay around the same level, but have a higher MAE than the other methods just mentioned. Only, as the magnitude increases, the spikes up-and downwards also become bigger. As expected, just as in Filzmoser et al. (2020), OLS only performs good for smaller magnitudes, where the cells are close to the true values. Figure 9b shows that the magnitude of contamination does not matter for the predictive performance, since CRM, MM, CRSM, shooting S and S all remain around the same level. DDC-MM and DDC-OLS have less spikes as the magnitude increases, and OLS is increasing in MSEP when the magnitude increases, as was already expected.

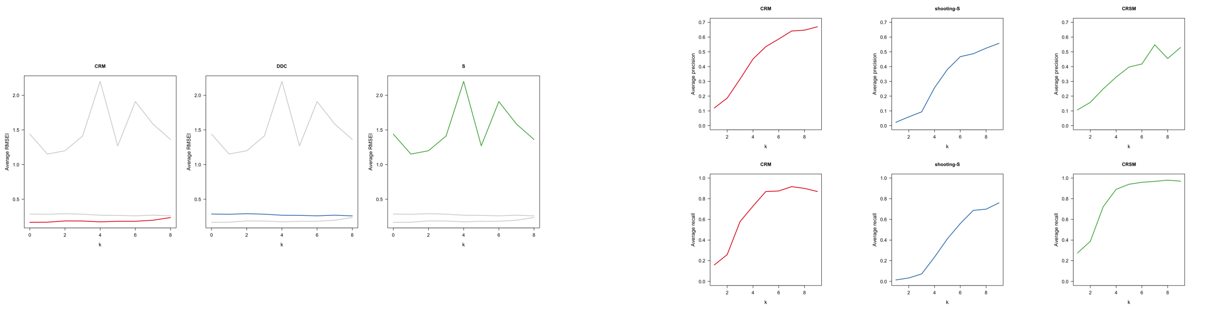


(a) MAE

(b) MSEP

Figure 9: MAE (a) and MSEP (b) when the magnitude of contamination (k) differs, where the amount of predictor variables is set to 50

The RMSEI in Figure 10a shows that the magnitude of contamination does not matter for CRM and DDC-imputation, sincec they stay relatively the same, which is also found by Filzmoser et al. (2020). Only shooting S varies a lot and has higher values for every k. In Figure 10b the precision and recall increases for all three cellwise robust regression methods as the magnitude increases. An explanation for this is that when the magnitude is higher, the outliers are more outlying and easier detected as outlier, than when they are good leverage points. This problem is also mentioned by Öllerer et al. (2016).



(a) RMSEI

(b) Precision and recall

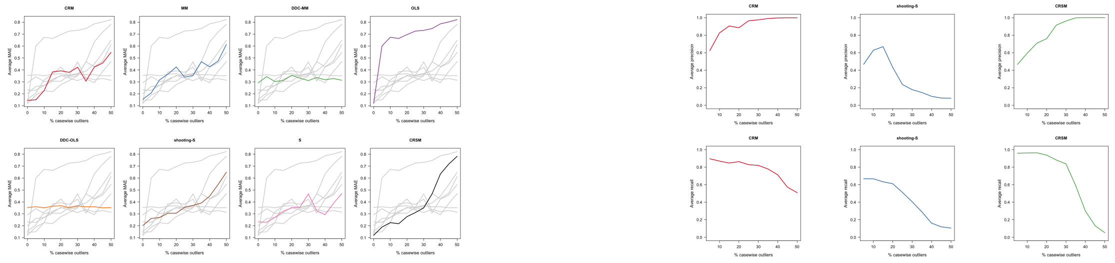
Figure 10: RMSEI (a) and precision and recall (b) when the magnitude of contamination (k) differs, where the amount of predictor variables is set to 50

When aggregating over the amount of variables and over the magnitude of contamination, the conclusions about which robust methods performs best in terms of MAE, MSEP, precision and recall are the same as in Subsection 4.3. Therefore, these boxplots, and the ones of RMSEI and average execution time can be found in Appendix B.3.1. This is also the case for the grouped boxplots, aggregated over the magnitudes of contamination, where the amount of variables form different subgroups. The conclusions about which method performs best remain the same, as in the case where the amount of magnitude did not vary. The only exception is the recall, where CRSM has now the best recall for all amount of variables (also for 250). The boxplots can be found in Appendix B.3.3, as well as the Figures when the amount of predictor variables was set to 250 in Appendix B.3.2.

4.5 Comparison between different percentages of casewise contamination

Changing the fraction of contamination, while keeping other parameters fixed (including k = 6), except for the number of simulations which is set to 10, is a setting that is already investigated by Filzmoser et al. (2020) on a casewise level. This fraction ranges from 0% to 50% with increments of 5%, while the cellwise contamination remains fixed at 10%. In this simulation study, the same results are obtained as Filzmoser et al. (2020), with the addition of shooting S, CRSM and S-regression.

Figure 11a illustrates the breakdown behavior, as mentioned by Filzmoser et al. (2020). OLS is the most efficient for a low percentage of contamination, while CRM, MM, shooting S, S and CRSM are better until around 40% contamination. When the casewise contamination exceeds 40%, the DDC-imputed methods are preferred, since they are robust and designed to handle cellwise outliers, even when over 50% of the cases contain cellwise outliers. However, these DDC-imputed methods have a higher bias at lower contamination levels compared to CRM and CRSM. In terms of bias, CRSM outperforms CRM and shooting S until around 30%, while shooting S outperforms CRM between 10-30%. The precision rate in Figure 11b shows that shooting S is not advised after around 20% of casewise contamination, as both precision and recall decrease rapidly. A possible explanation for this can be that the RMSEI for shooting S is the highest for every simulation. Both CRM and CRSM have higher precision, but lower recall rates as contamination increases, where CRM performs slightly better.

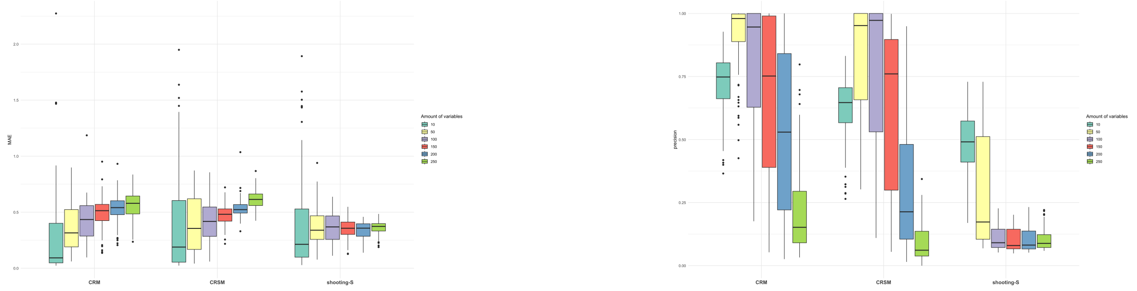


(a) MAE

(b) Precision and recall

Figure 11: MAE (a) and precision and recall (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables is set to 50

When aggregating over the percentages of casewise contamination, in terms of MAE, when looking at the grouped boxplot in Figure 12a, CRM is recommended for 10 and 50 variables. However, for higher amount of variables shooting S outperforms CRM and CRSM with a smaller average MAE. The grouped boxplot of the precision in Figure 12b shows that CRM is preferred for every amount of predictor variables, followed by CRSM with slightly lower performance. Shooting S has the lowest precision of all three.



(a) MAE

(b) Precision and recall

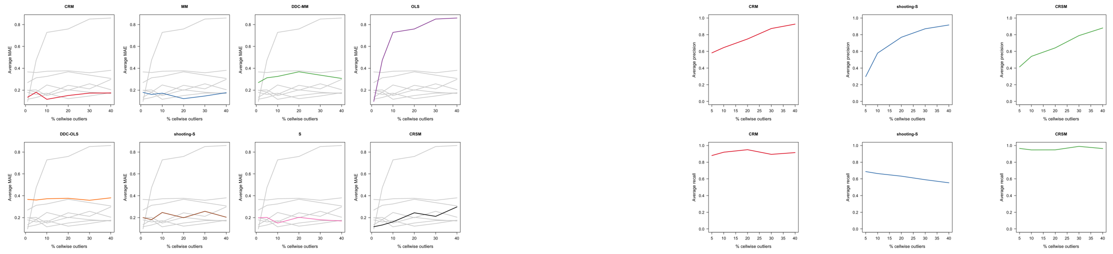
Figure 12: MAE (a) and precision (b) when the percentage of casewise contamination differs between 0% and 50%, for the predictor variables 10, 50, 100, 150, 200 and 250

The other Figures when the amount of variables is 50 can be found in Appendix B.4.1 the Figures when there are 250 variables are in Appendix B.4.2, the aggregated plots are in Appendix B.4.3, and the grouped boxplots in Appendix B.4.4.

4.6 Comparison between different percentages of cellwise contamination

As last setting, the fraction of cellwise contamination will be varied, taking the values 1, 5, 10, 20, 30 and 40%. The other parameters are kept fixed, meaning that the fraction of casewise contamination is 5%. The number of simulations will be set to 10. This is an extension of the simulation study of Filzmoser et al. (2020), which did not investigate the effect of different fractions of cellwise contamination.

In the case of 50 variables, CRSM performs the best until approximately 10% cellwise contamination, as illustrated in Figure 13a. After that, CRM is the best method between 10%-20%, while MM-regression outperforms the ohter methods for percentages higher than 20%. This can be due to the fact that a high amount of cellwise contamination means that most of the row is contaminated, which is looking like rowwise/casewise contamination. Shooting S and S-regression also perform quite well, only OLS increases sharply as the percentage of contamination increases. DDC-MM and DDC-OLS remain relatively stable because of their robustness, but have a higher MAE than the cellwise robust methods. The precision of shooting S starts really low for a low percentage of cellwise contamination in Figure 13b, but improves and reaches around the same level as CRM for 40%. CRM performs slightly better than CRSM. For the recall, it is the other way around, where CRSM outperforms CRM slightly. Here, shooting S has a decreasing rate of recall, as the percentage of contamination increases.

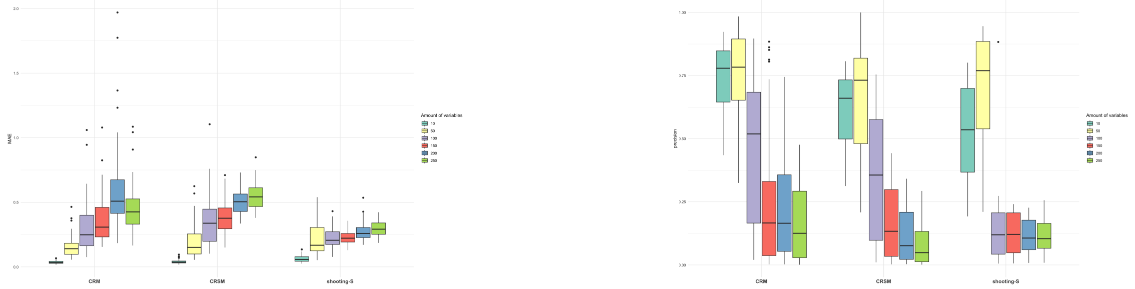


(a) MAE

(b) Precision and recall

Figure 13: MAE (a) and precision and recall (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables is set to 50

When aggregating over the fraction of cellwise contamination, the results in Figure 14a indicate that CRM is on average the best in assessing biases for 10 and 50 variables, while shooting S outperforms CRM and CRSM for 100 variables or higher. The precision in Figure 14b shows that CRM consistently achieves the highest precision for all variable amounts. However CRSM is still slightly better than shooting S except for 250 variables.



(a) MAE

(b) Precision and recall

Figure 14: MAE (a) and precision (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

The other Figures for 50 predictor variables are in Appendix B.5.1, whereas the Figures for 250 variables can be found in Appendix B.5.2. The aggregated plots are in Appendix B.5.3, and the grouped boxplots in Appendix B.5.4.

5 Empirical application

For the real data example, the dataset is from Öllerer et al. (2016), allowing for a comparison of performances with actual data instead of only having simulations. The dataset is called *Auto* and consists of 8 predictor variables and 74 sales of 1987 vintage cars in the US. A description of the variables, can be seen in Öllerer et al. (2016)

In Figure 15 the first column displays heatmaps for CRM (top), shooting S (middle) and CRSM (bottom) of the rows which are detected to have cellwise outliers. These outliers are the blue or red boxes and show whether their value deviates downwards or upwards. Subsequently, CRM, shooting S and CRSM impute the blue boxes with bigger values, and red boxes with smaller values. The heatmap of this imputation can be seen in the second column of this Figure.

From the figure, it is clear that most of the imputations by CRM improve the value of the cell, same as for CRSM. However, for shooting S the imputation is sometimes worse than the original value, including negative values and values much bigger than the other values of that variable in the same column. This aligns with the RMSEI for shooting S which is way higher than for CRM and CRSM.

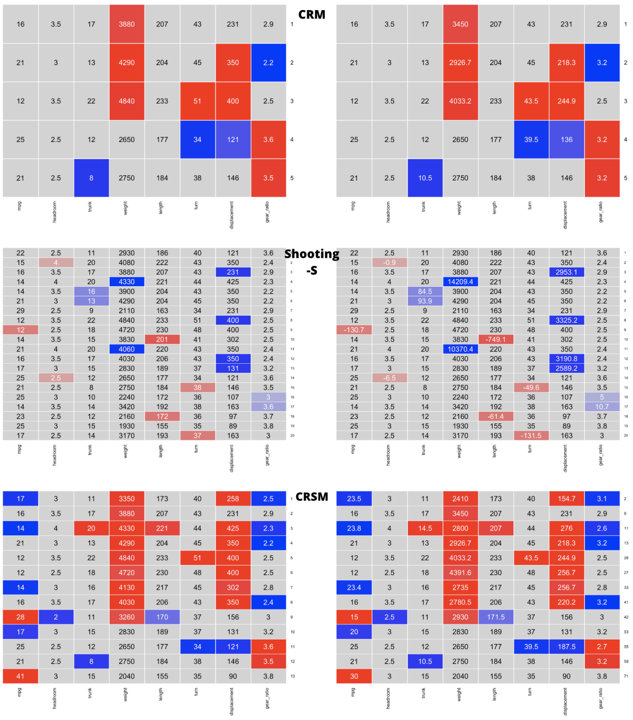


Figure 15: Heatmaps of outliers detected by CRM, shooting S and CRSM (left), and the heatmaps of the imputation, with the imputed values (right). Blue cells are contaminated cells where their values are deviating downwards and red cells are contaminated cells whose values are deviating upwards

6 Conclusion

The cellwise robust M regression (CRM) estimator, the CRM estimator with S-regression as initial step (CRSM) and the shooting S-estimator are three cellwise robust regression methods. CRM originally starts with MM regression, where S-regression is another option. It uses an iteratively reweighted least squares procedure, where SPADIMO is applied in each iteration to detect and downweight only the outlying cells, rather than an entire observation. On the other hand, the shooting S-estimator combines S-regression with the coordinate descent algorithm to handle cellwise outliers. Both have the advantage to maintain a larger fraction of uncontaminated data cells compared to casewise robust methods. While the performance of CRM and shooting S has already been investigated, this research is the first to compare these cellwise robust techniques, and to introduce CRM with S-regression.

A simulation study, starting with the same settings as Filzmoser et al. (2020), found that CRM shows comparable bias and predictive performance compared to MM-regression, its corresponding casewise robust method. Shooting S has a lower bias than its casewise robust method S-regression, particularly with many variables (more than 50). It outperforms CRM and CRSM in precision for high amounts of variables (more than 150). In general, CRM has the best precision. However, in terms of imputation accuracy, CRM outperforms shooting S and DDC-imputation for a small number of variables, while DDC shows a better performance for higher amounts of predictor variables. CRSM performs similar to CRM in controlling bias and predictive power for low amounts of variables, but has poorer performance for higher amounts. Additionally, average execution time of CRSM is always much longer than shooting S and CRM. On the other hand, if recall is important, CRSM is preferred since it has the highest rate (until 200).

When the magnitude of contamination changes, the cellwise robust methods consistently have around the same predictive performance and low bias. However, as contamination increases, there more deviations up-and downwards. As the magnitude increases, the precision and recall also increase as outliers become easier to detect. For low percentages of casewise contamination, OLS is the best. Until 40% the cellwise robust methods are preferred and for higher percentages DDC-imputed methods are better, since they are designed to handle high levels of contamination. Up to 10% cellwise contamination, CRSM has the lowest bias, after which CRM becomes the best for assessing biases until 20%. For higher fractions of cellwise contamination, MM regression outperforms other methods, as it can be considered as casewise contamination. The recall rate decreases for shooting S, while CRSM outperforms CRM in recall and vica verse in precision.

An empirical application with heatmaps shows that shooting S can worsen cell deviations during imputation, whereas CRM and CRSM are better in imputing cells.

One possible limitation is that we kept the parameters identical to those in Filzmoser et al. (2020) to obtain the same results. However, alternative parameter settings could lead to different conclusions. For instance, changing the number of cases, the tolerance parameter or the outlyingness factor could impact the performance. Besides that, it can also be interesting to investigate the influence of different fractions of casewise contamination when the percentage of cellwise contamination varies, and vica versa.

When there are missing values, the row is just removed, while they are often present in empirical applications. Removing these rows could lead to information loss and can lead to biased results. Handling missing values, such as mean substitution, single imputation or multiple imputation (Acock, 2005), could improve estimation.

For further research, skipped Huber, Tukey's biweight for CRM or the Hampel function for shooting S can be considered, instead of the Hampel function for CRm and Tukey's biweight for shooting S. Filzmoser et al. (2020) namely mentions that the choice of this function can affect the performance.

As extension for the empirical application, cross validation can give more insights into the performance of the cellwise robust methods in a real dataset.

Lastly, Öllerer et al. (2016) suggests as another option the shooting MM-estimator as cellwise robust regression estimator, instead of the shooting S-estimator. Therefore, comparing the performance of this estimator with the other robust methods could be interesting.

References

Aalfons, C. (2019). Aalfons/shootings: Implementation of the shooting s-estimator for cellwise robust regression.

Abonazel, M. and Rabie, A. (2019). The impact of using robust estimations in regression models: An application on the egyptian economy. *Journal of Advanced Research in Applied Mathematics and Statistics*, 4(2):8–16.

Acock, A. C. (2005). Working with missing values. *Journal of Marriage and family*, 67(4):1012–1028.

Alma, Ö. G. (2011). Comparison of robust regression methods in linear regression. *Int. J. Contemp. Math. Sciences*, 6(9):409–421.

Avagyan, V. and Mei, X. (2022). Precision matrix estimation under data contamination with an application to minimum variance portfolio selection. *Communications in Statistics-Simulation and Computation*, 51(4):1381–1400.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., and Li, S. (2019). Fnn: Fast nearest neighbor search algorithms and applications.

Buckland, M. and Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19.

Chen, D. (2020). Tukey's biweight estimation for uncertain regression model with imprecise observations. *Soft Computing*, 24(22):16803–16809.

Debruyne, M., Höppner, S., Serneels, S., and Verdonck, T. (2019). Outlyingness: Which variables contribute most? *Statistics and Computing*, 29:707–723.

Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., and Verdonck, T. (2020). Cellwise robust m regression. *Computational Statistics & Data Analysis*, 147:106944.

Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416.

Hoppner, S. (2020). Sebastiaanhoppner/crm: Cellwise robust m-regression.

Huber, P. J. (2011). Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer.

Hubert, M., Rousseeuw, P. J., and Van den Bossche, W. (2019). Macropca: An all-in-one pca method allowing for missing values as well as cellwise and rowwise outliers. *Technometrics*, 61(4):459–473.

Maechler, M., Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., Verbeke, T., Koller, M., Conceicao, E. L. T., and Anna di Palma, M. (2023). *robustbase: Basic Robust Statistics*. R package version 0.95-1.

Maronna, R. A., Martin, R. D., Yohai, V. J., and Salibián-Barrera, M. (2019). *Robust statistics: theory and methods (with R)*. John Wiley & Sons.

Öllerer, V., Alfons, A., and Croux, C. (2016). The shooting s-estimator for robust regression. *Computational Statistics*, 31:829–844.

Rousseeuw, P. J. and Bossche, W. V. D. (2018). Detecting deviating data cells. *Technometrics*, 60(2):135–145.

Rousseeuw, P. J. and Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283.

Rousseeuw, P. J. and Leroy, A. M. (2005). *Robust regression and outlier detection*. John wiley & sons.

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.

Segaert, P., Lopes, M. B., Casimiro, S., Vinga, S., and Rousseeuw, P. J. (2019). Robust identification of target genes and outliers in triple-negative breast cancer data. *Statistical methods in medical research*, 28(10-11):3042–3056.

Štefelová, N., Alfons, A., Palarea-Albaladejo, J., Filzmoser, P., and Hron, K. (2021). Robust regression with compositional covariates including cellwise outliers. *Advances in Data Analysis and Classification*, 15(4):869–909.

Susanti, Y., Pratiwi, H., Sulistijowati, S., Liana, T., et al. (2014). M estimation, s estimation, and mm estimation in robust regression. *International Journal of Pure and Applied Mathematics*, 91(3):349–360.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475.

Yohai, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. *The Annals of statistics*, pages 642–656.

A Code description

For the code description, there is a README to describe the scripts and functions used in R. Besides that, there is a explanation of how to results of this research can be obtained.

A.1 README

CRM, CRSM and shooting S

These files contain the scripts for the different simulations of the thesis: Shooting S-estimator and cellwise robust M regression estimator: an extensive simulation study of cellwise robust techniques, by Cuijpers, A. (577305ac)

Three cellwise robust regression estimators are considered, namely shooting S, cellwise robust M regression with MM-regression as initial step (CRM) and CRM with S-regression as initial step (CRSM).

The shooting S algorithm is already introduced by:

V. Öllerer, A. Alfons, C. Croux (2016). The shooting S-estimator for robust regression. Computational Statistics, 31(3), 829-844

And the cellwise robst M regression (CRM) is already introduced by:

Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., and Verdonck, T. (2020). Cellwise Robust M Regression. Computational Statistics and Data Analysis, 147:106944. DOI: 10.1016/j.csda.2020.106944

R scripts of CRM & CRSM

These scripts are based on the crmReg package, with some modifications:

- cellwiseheatmap*: function that creates the cellwise heatmaps
- crmAuto: implementation of the CRM algorithm, which is modified such that S-regression as initial step also can be chosen, and it can be used for dataset, such as the Auto dataset used in this thesis.
- daprpr*: function to preprocess the data
- HampelWeightFunction*: function that calculates the weights of the Hampel weight function
- impute_outlying_cells*: imputes the cells that are detected as outlying
- predict.crm*: function that predicts response values
- scaleResidualsByMAD*: function to scale the residuals by the Mean Absolute Deviation
- spadimo*: consists of multiple functions to apply the SPArse DIrections of Maximal Outlyingness (SPADIMO) algorithm, that detects which cells are outlying

*Exactly the same script as can be found in the package crmReg, without any modifications

R script of shooting S

The script to obtain the shooting S-estimator is: shootingS, and it consists of multiple functions:

- shooting: function to obtain the shooting S-estimator for either the biweight loss or skipped Huber loss, which returns the estimates

- univ_est*: function which calculates one step of the loop of the shooting algorithm
- scale_iter*: function that computes the M-scale estimate
- lmrob.hubx*: function which first standardizes and Huberizes the data and then computes the MM-estimate
- w_bi*: calculates the value of the biweight weight function for an input value
- rho_bi*: calculates the value of the biweight rho function for an input value
- w_skh*: calculates the skipped Huber weight function for an input value
- rho_skh*: calculates the skipped Huber rho function for an input value

*Exactly the same function as can be found in the file function_shootingS.R of Aalfons (2019)

R scripts of the simulations

The regression models are: CRM, MM, DDC-MM, OLS, DDC-OLS, S, shooting S, and CRSM

- boxplots: contains all the code to create for all the simulations, for every evaluation criteria, the aggregated figures and the grouped boxplots
- simulation1: The first simulation, where all simulation settings are kept fixed, with 50 simulations in total. It fits the regression models, and evaluates the performance of these models.
- simulation2: Second simulation, where the magnitude of contamination differs. It fits the regression models, and evaluates the performance of these models
- simulation3: Third simulation, where the percentage of casewise contamination differs between 0% and 50% in steps of 5%. It fits the regression models, and evaluates the performance of these models
- simulation4: Fourth simulation, where the percentage of cellwise contamination differs, by taking the percentages 1%, 5%, 10%, 20%, 30% and 40%

R script of empirical application

auto_data is also included, which is an Excel file that consists of all the different vintage cars from the United States in 1987. It consists of 8 predictor variables and 74 sales.
There is only one script for this application:

- autoCode: the script for the empirical application, which uses the Auto dataset to obtain cellwise heatmaps of the imputation methods of CRM, shooting S and CRSM.

A.2 Explanation for obtaining results

It should be stated that the results of simulation 1, 2, 3 and 4 can already be obtained by running these scripts for the number of 50 predictor variables. In Subsection 4.1 the settings are described. However, in the scripts where a source function is used, the own path of the user have to be inserted. In this case, the path to crmAuto was: "/Volumes/data/Documenten/Econometrie/Bachelor scriptie/Rcode/crmAuto.R".

CRM, shooting S and CRSM can be evaluated with each other by running the script "simulation1". The comparison between amount of predictor variables is performed by running "simulation1", but by changing the amount of predictor variables every time. Thus, after every run p is set to another amount of variables, where it takes in total the values 10, 50, 100, 150, 200 and 250. The outputs of every run are stored as an RData file, where for example the file for 10 variables is called "simulation1p10.RData". The aggregated Figures and the grouped boxplots can be obtained after having all the RData files of all the variables, by running the *simulation 1* parts of the script "boxplots".

The comparison between magnitudes of contamination can be done by running "simulation2". The Aggregated Figures and grouped boxplots of this corresponding Subsection in Appendix B.3 are obtained by running the *simulation 2* parts of "boxplots", after storing all outputs of all amount of variables in RData files. For the evaluation criteria with 50 and 250 predictor variables, besides p=50, the script also needs to be compiled with p=250.

For the comparison between different percentages of casewise contamination, "simulation3" needs to be used. And again, for the aggregated figures and grouped boxplots, the parts of *simulation 3* of "boxplots" is used.

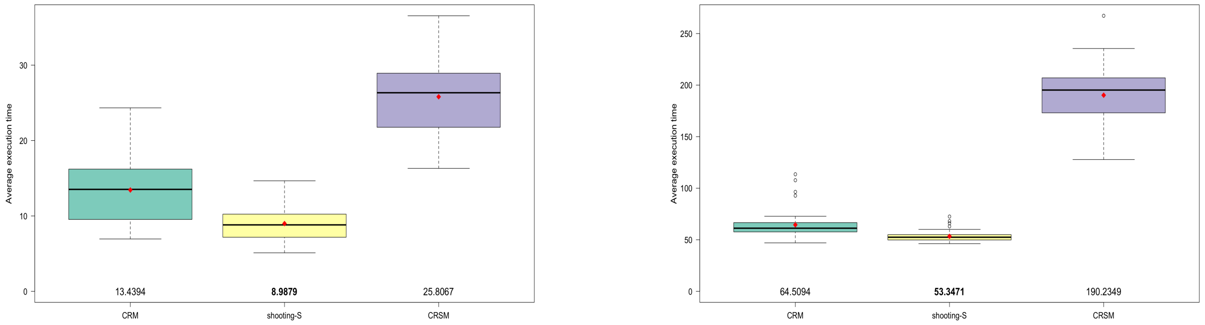
"simulation4" is runned for the comparison between different percentages of cellwise contamination. The *simulation 4* part of "boxplots" can be used, after having all the necessary RData files.

The "autoCode" reads an excel file, thus for this the path to the users "auto_data" Excel file needs to be inserted. After this, the rest of the code can immediately be runned to obtain the cellwise heatmaps.

B Simulation results

B.1 Evaluation of CRM, shooting S and CRSM

The methods perform the same for 50 or 200 predictor variables when looking at the average execution time in Figure 16. Shooting S always is the fastest, followed by CRM and CRSM takes the longest time.



(a) Average execution time when the amount of predictor variables is set to 50

(b) Average execution time when the amount of predictor variables is set to 200

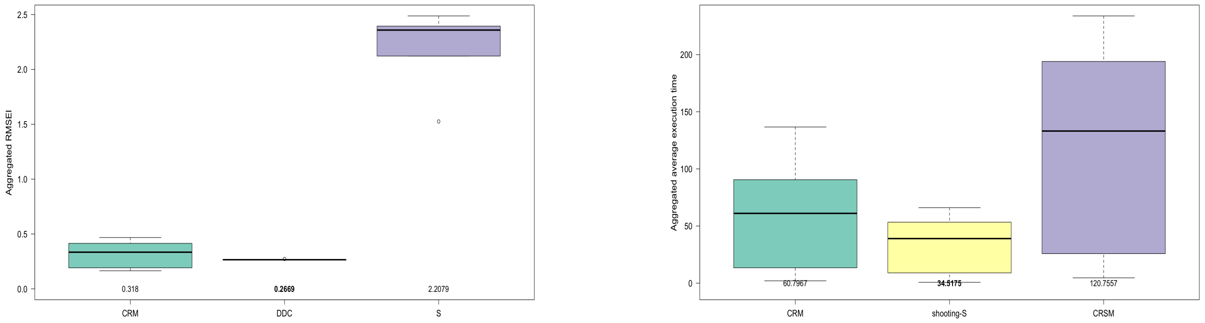
Figure 16: Comparison of average execution time for 50 and 200 predictor variables

B.2 Comparison between amount of predictor variables

In this Subsection the comparison is considered where the amount of predictor variables is varying between 10 and 250.

B.2.1 Aggregated RMSEI and execution time

Figure 17 shows the aggregated RMSEI and average execution time, where the results are the same as Subsection 4.2. The imputation method for shooting S-regressions has a much higher RMSEI, than for the DDC- and CRM imputation methods. The execution time for CRSM takes the longest, while shooting S is the fastest.



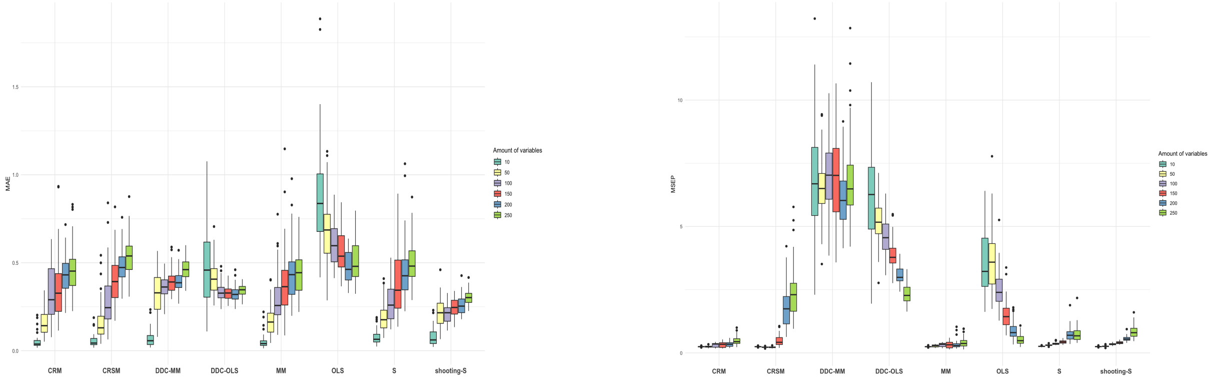
(a) Aggregated RMSEI

(b) Aggregated average execution time

Figure 17: Aggregated RMSEI (a) and average execution time (b) when the amount of predictor variables are 10, 50, 100, 150, 200 and 250

B.2.2 Boxplots all predictor variables

Here all the boxplots are shown, where all the predictor variables (10, 50, 100, 150, 200 and 250) are shown next to each other for each method, to compare them with each other. In Figure 18 it can be seen that shooting S has the lowest MAE for all predictor variables, while OLS always has the biggest values. However, OLS and DDC-OLS are the only methods that improve in MAE when there are more variables. For the MSEP, DDC-MM is the highest for all amount of variables, and stays around the same level. As the amount of variables increase, CRSM will perform significantly less than CRM and MM. DDC-OLS and OLS are again decreasing in MSEP, as the predictor variables increase.

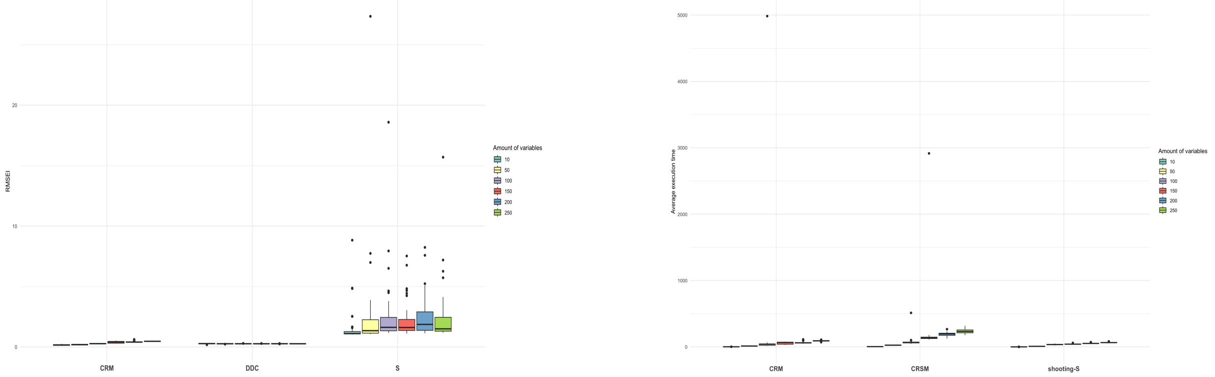


(a) MAE

(b) MSEP

Figure 18: MAE (a) and MSEP (b) for the predictor variables 10, 50, 100, 150, 200 and 250

Figure 19 shows that the RMSEI for S is higher for every setting of predictor variables, while DDC remains the most at the same level. However, CRM still performs quite well compared to the imputation method for shooting S. The conclusion of the fastest and slowest method when looking at the execution time, also remains the same, thus that CRSM takes the longest and shooting S is the fastest. Also, when the amount of predictor variables increases, shooting S increases in a slower rate than CRSM and CRM.



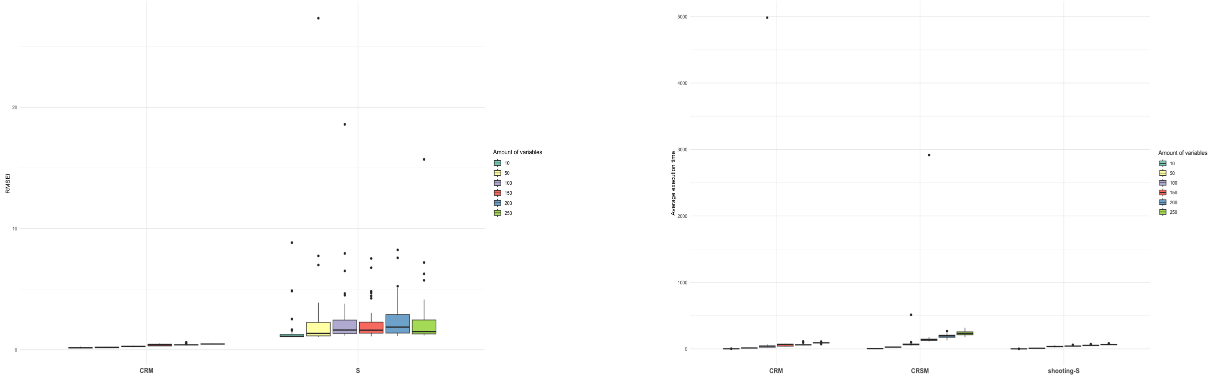
(a) RMSEI

(b) Average execution time

Figure 19: RMSEI (a) and average execution time (b) for the predictor variables 10, 50, 100, 150, 200 and 250

B.2.3 Boxplots CRM, CRSM and shooting S

In Figure 20 we see that the RMSEI for shooting S is not as good as compared to CRM. However, the execution time of shooting S is the best of all three.



(a) RMSEI

(b) Average execution time

Figure 20: RMSEI (a) and average execution time (b) for CRM, CRSM and shooting S

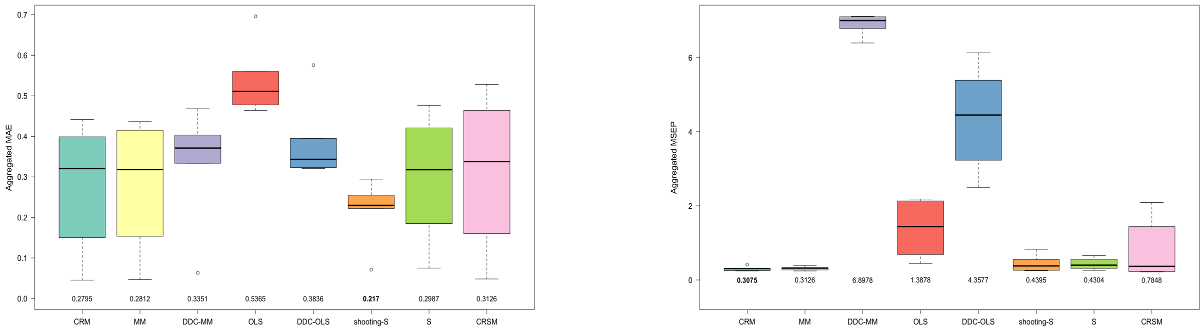
B.3 Comparison between magnitudes of contamination

In this Section, the results are shown in the case where the magnitude of contamination differs between k = 1,...,8.

B.3.1 Aggregated evaluation criteria

Here, there is aggregated over the amount of predictor variables (10, 50, 100, 150, 200 and 250) and the magnitude of contamination (1,..8).

In Figure 21 the MAE is the lowest for shooting S, and the MSEP for CRM. Thus, when the priority is assessing biases, then shooting S is preferred. When it is important to have a high predictive performance, CRM is preferred.



(a) MAE

(b) MSEP

Figure 21: Aggregated MAE (a) and MSEP (b) when the magnitude of contamination (k) differs, when the amount of predictors are 10, 50, 100, 150, 200 and 250

Figure 22 shows that CRM has the best precision, and CRSM the best recall when aggregating over the amount of variables and magnitude. Shooting S has both the lowest precision and recall.

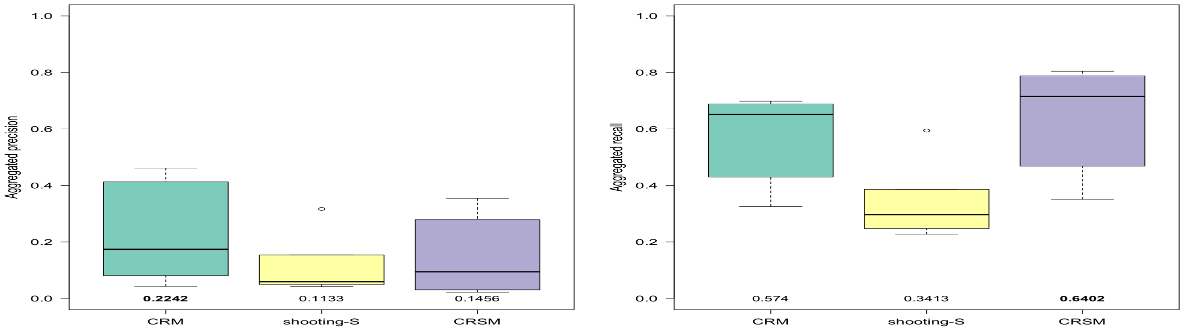
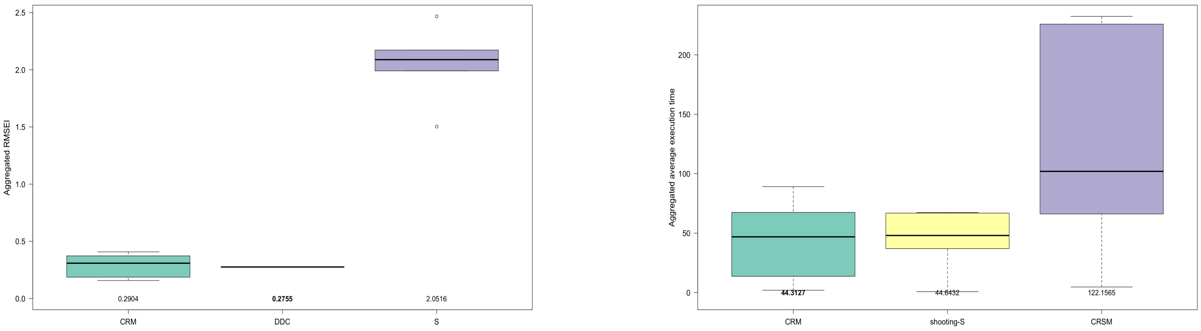


Figure 22: Aggregated precision (left) and recall (right) when the magnitude of contamination (k) differs, when the amount of predictors are 10, 50, 100, 150, 200 and 250

Again, in Figure 23, the imputation method of shooting S performs not as good as CRM and DDC-imputation methods. However, the execution time is almost as good as CRM. CRSM takes the longest time to execute.



(a) RMSEI

(b) Average execution time

Figure 23: Aggregated RMSEI (a) and average execution time (b) when the magnitude of contamination (k) differs, when the amount of predictors are 10, 50, 100, 150, 200 and 250

B.3.2 Evaluation criteria with 50 and 250 predictor variables

The results for MAE, MSEP, RMSEI, precision and recall of 50 predictor variables when the magnitude varies, is discussed in Subsection 4.4. The average execution time can be seen in Figure 24, where all three methods take around the same time, except for k = 8, where shooting S and CRSM take longer than CRM.

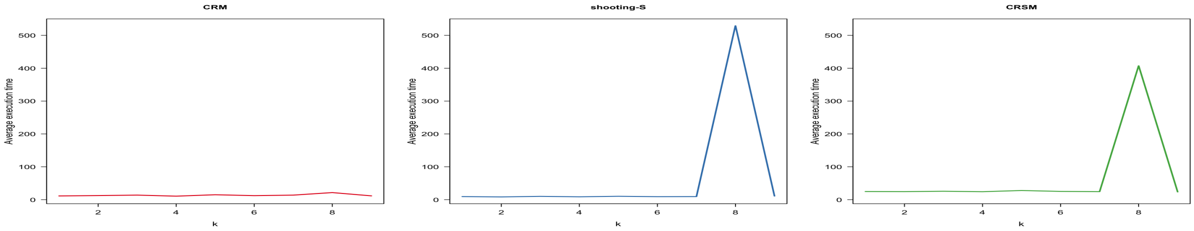
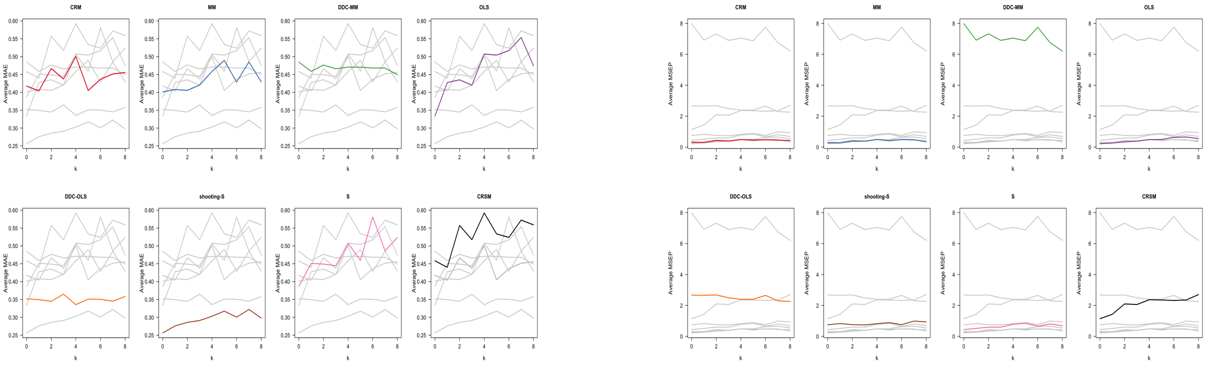


Figure 24: Average execution time when the magnitude of contamination (k) differs, when the amount of predictors is set to 50

When the amount of predictor variables are set to 250, the results of MAE and MSEP are in Figure 25, where shooting S performs the best in assessing bias for all different magnitudes. CRM, MM, shooting S, S and OLS perform good in terms of predictive accuracy.



(a) MAE

(b) MSEP

Figure 25: MAE (a) and MSEP (b) when the magnitude of contamination (k) differs, when the amount of predictors is set to 250

The precision and recall in Figure 26, both increase in rate as the magnitude increases. Shooting S has the highest precision and recall when there are 250 predictor variables.

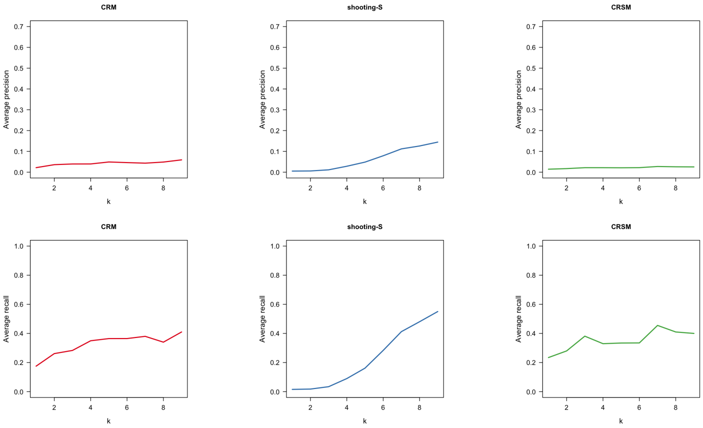
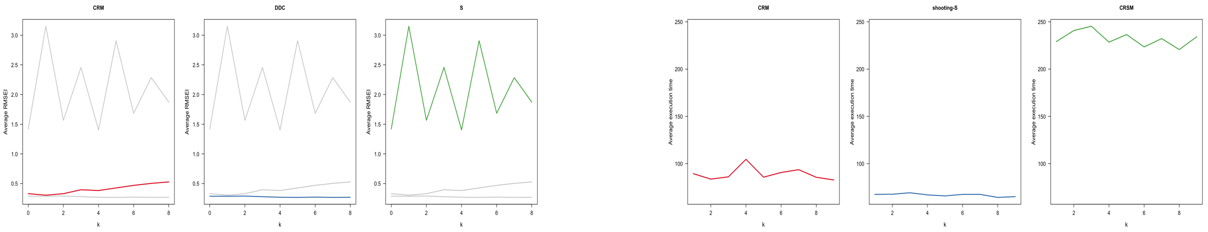


Figure 26: Precision (left) and recall (right) when the magnitude of contamination (k) differs, when the amount of predictors is set to 50

In Figure 27 we see again that the imputation method for S is not good compared to CRM and DDC. On the other hand, the execution time is again faster than both CRM and CRSM, where CRSM takes the longest time.



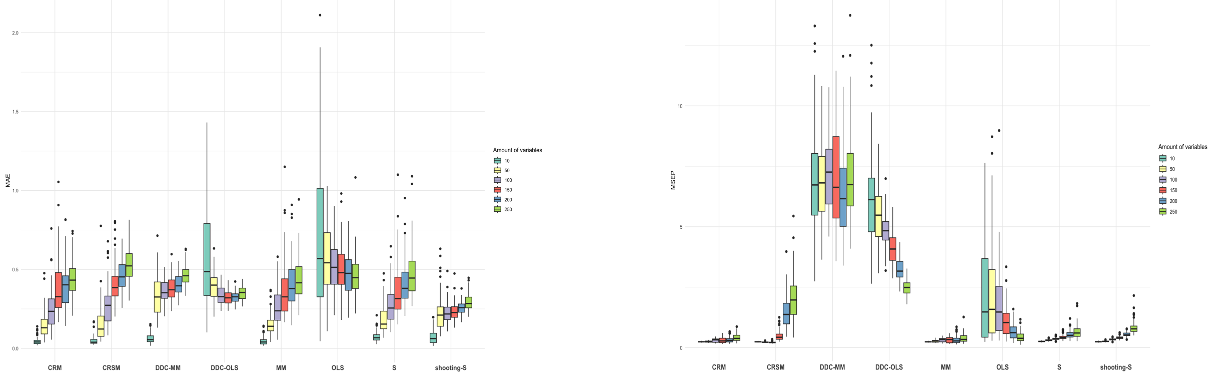
(a) RMSEI

(b) Average execution time

Figure 27: Aggregated RMSEI (a) and average execution time (b) when the magnitude of contamination (k) differs, when the amount of predictors is set to 250

B.3.3 Grouped boxplots

The grouped boxplots, where the amount of variables form different subgroups have the same conclusions as 4.3. The MAE in Figure 9 has the same result that CRM has the lowest bias for 10 and 50 variables, and shooting S for higher amounts. When looking at the MSEP, CRSM has the best predictive performance until 100 variables, and after that CRM has the best one.

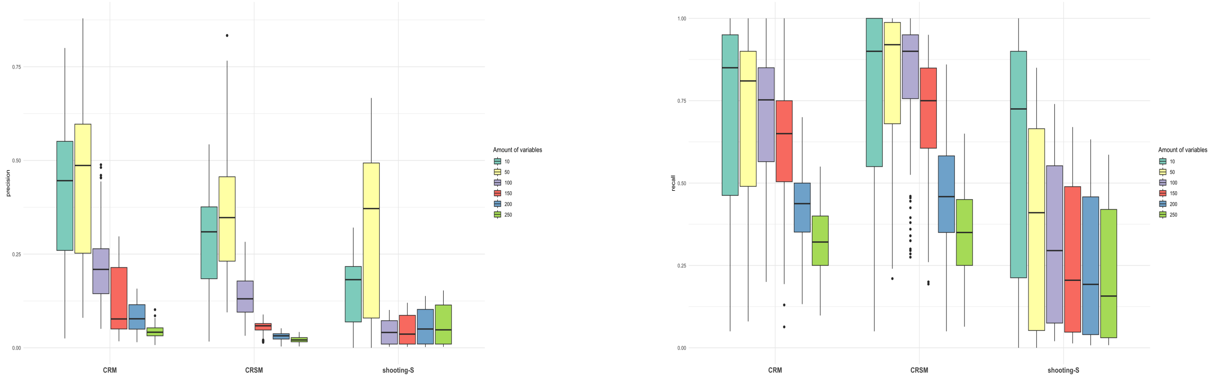


(a) MAE

(b) MSEP

Figure 28: MAE (a) and MSEP (b), aggregated over the magnitude of contamination (k), for the predictor variables 10, 50, 100, 150, 200 and 250

The precision in Figure 29 shows that the precision until 200 variables is best for CRM, and for 250 variables for shooting S. However, the recall is best for CRSM for all amount of variables.

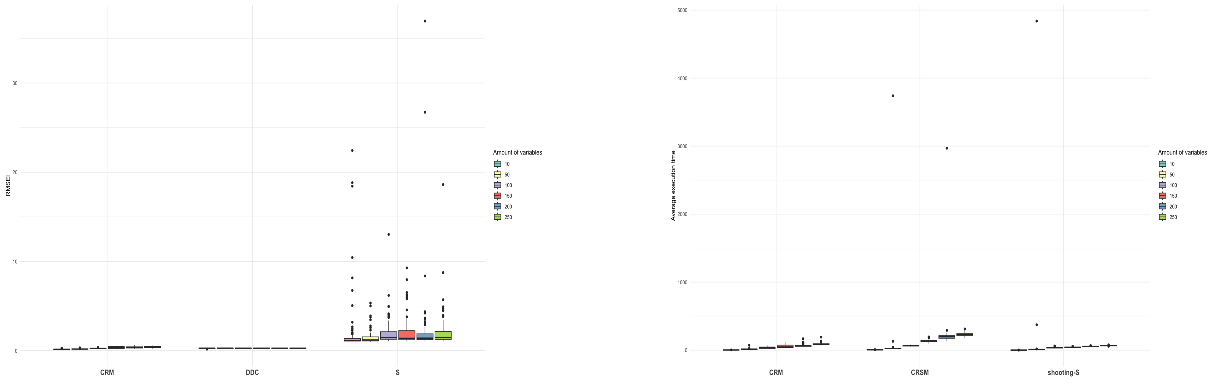


(a) Precision

(b) Recall

Figure 29: Precision (a) and recall (b), aggregated over the magnitude of contamination (k), for the predictor variables 10, 50, 100, 150, 200 and 250

Again, the RMSEI is the highest for the imputation method of shooting S, as can be seen in Figure 30, while CRM and DDC are quite low. On the other hand, the average execution time of shooting S is faster than CRM and CRSM.



(a) RMSEI

(b) Average execution time

Figure 30: RMSEI (a) and average execution time (b), aggregated over the magnitude of contamination (k), for the predictor variables 10, 50, 100, 150, 200 and 250

B.4 Comparison between different percentages of casewise contamination

In this Section, the results are shown in the case where the percentage of contamination varies from 0 to 50% in steps of 5%, where the level of cellwise contamination is fixed at 10%.

B.4.1 Evaluation criteria with 50 predictor variables

The MAE, precision and recall are already discussed in Subsection 4.5. The MSEP can be seen in Figure 31, where DDC-MM and DDC-OLS improve slightly as the percentage of contamination increases, while the other methods increase slighly in terms of MSEP. The only exception is OLS, which almost increases in a linear line as the contamination becomes more.

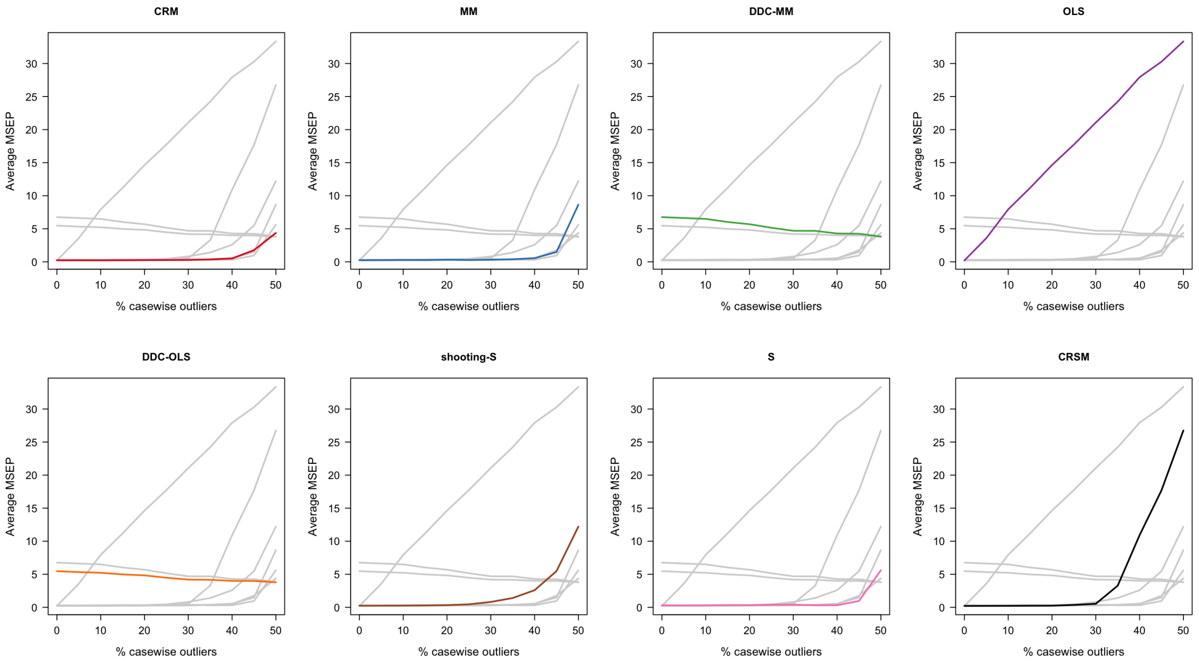
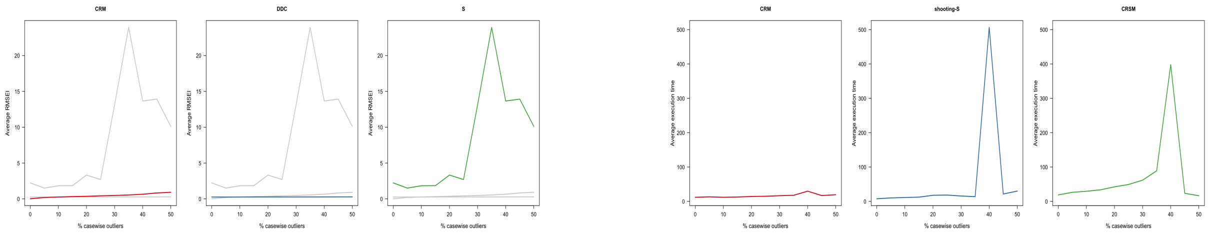


Figure 31: MSEP when the percentage of casewise contamination differs, where the amount of predictor variables is set to 50

The RMSEI and average execution time in Figure 32 show that shooting S again has the highest RMSEI compared to CRM and DDC, while these other stay almost at the same value. The average execution time increase more for shooting S and CRSM when the percentage exceeds 35%, while CRM stays relatively stable.



(a) RMSEI

(b) Average execution time

Figure 32: RMSEI (a) and average execution time (b) when the percentage of casewise contamination differs, where the amount of predictor variables is set to 50

B.4.2 Evaluation criteria with 250 predictor variables

When there are 250 variables, shooting S clearly outperforms the other methods in terms of assessing biases, as can be seen in Figure 33. After around 30% of casewise contamination, DDC-OLS outperforms shooting S, and is then the best method. CRM, MM, S and CRSM are even higher in terms of MAE compared to OLS.

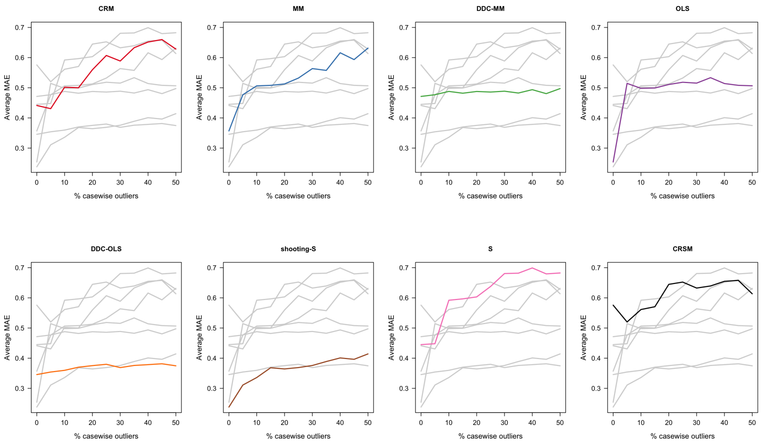
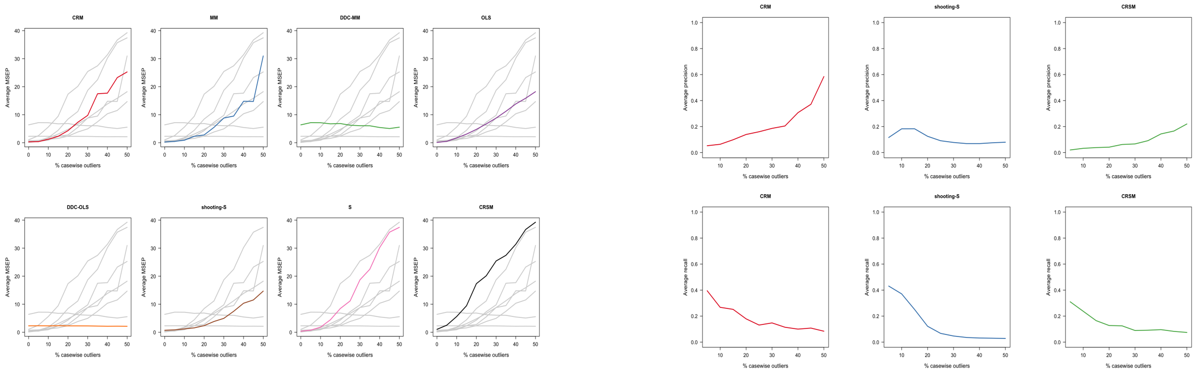


Figure 33: MAE when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables is set to 250

CRM, MM, S, shooting S, CRSM and OLS are all only good in terms of MSEP, while after already 10% of contamination, DDC-OLS and DDC-MM outperform the rest. This is shown in Figure 34. The precision and recall for shooting S again show that it is not advised to use shooting S when there is more than 20% of contamination. Also, CRM and CRSM are decreasing in terms of precision when the contamination percentage increases, while the recall increases.

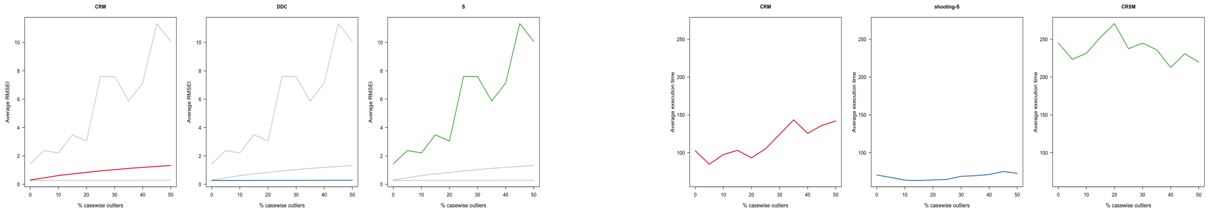


(a) MSEP

(b) Precision and recall

Figure 34: MSEP (a) and precision and recall (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables is set to 250

The RMSEI is the highest for shooting S, and DDC is the best after around 10% of casewise contamination, as can be seen in Figure 35. The execution time takes again the longest for CRSM, while shooting S is the fastest.



(a) RMSEI

(b) Average execution time

Figure 35: RMSEI (a) and average execution time (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables is set to 250

B.4.3 Aggregated evaluation criteria

DDC-OLS, DDC-MM and OLS perform the best in terms of MAE when aggregating over the amount of predictor variables and percentages of casewise contamination, which is shown in Figure 36.

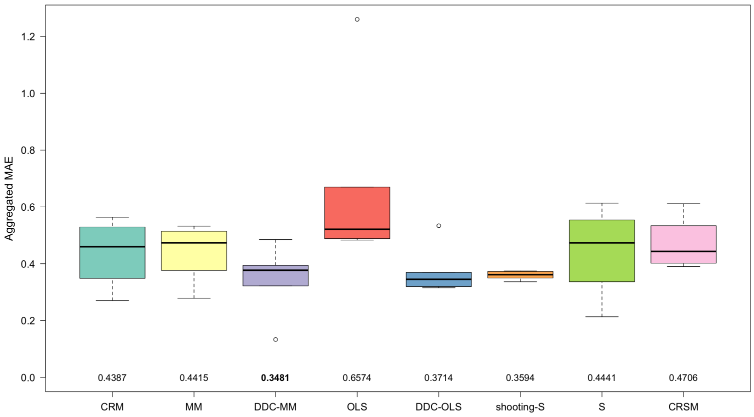
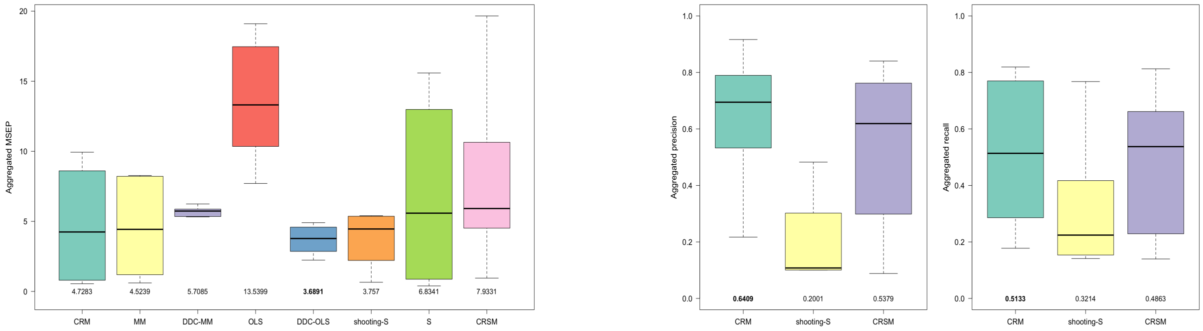


Figure 36: Aggregated MAE when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

In terms of MSEP, DDC-OLS and shooting S perform the best, followed by CRM and MM in Figure 37. OLS has the highest value, whereas CRSM and S are also quite high.

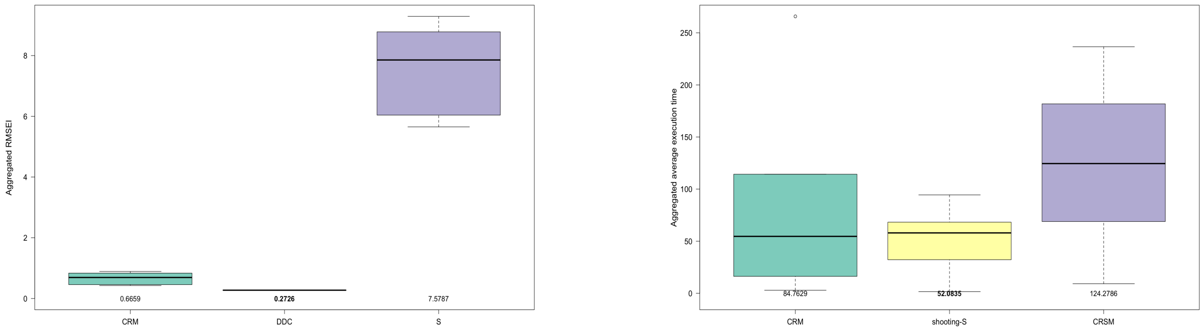


(a) MSEP

(b) Precision and recall

Figure 37: Aggregated MSEP (a) and precision and recall (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

Again, in Figure 38, the imputation method for shooting S-regressions performed the worst when aggregating over the percentages of contamination and amount of variables. Here, DDC is the best, followed by the imputation method of CRM. The aggregated average execution time is the best for shooting S, followed by CRM, and CRSM takes the longest time.



(a) RMSEI

(b) Average execution time

Figure 38: Aggregated RMSEI (a) and average execution time (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

B.4.4 Grouped boxplots

DDC-MM is the best in handling larger values of casewise contamination, therefore it performs the best in terms of MAE when aggregating over the percentages of casewise contamination, as can be seen in Figure 39. Only for 250 variables, shooting S is preferred.

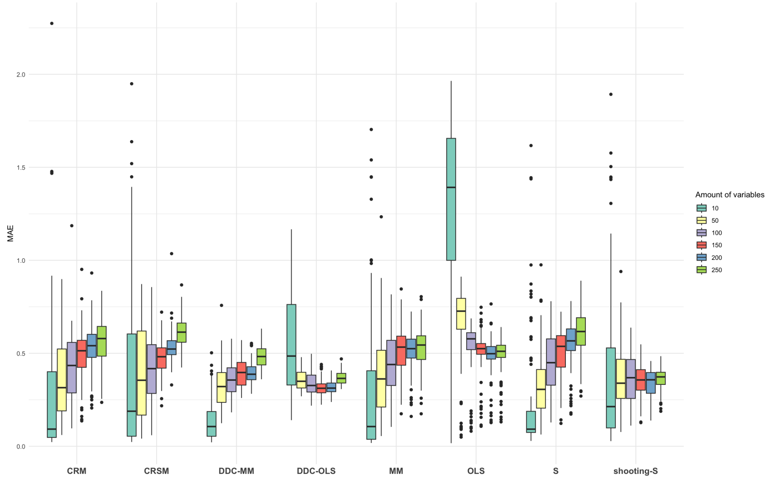
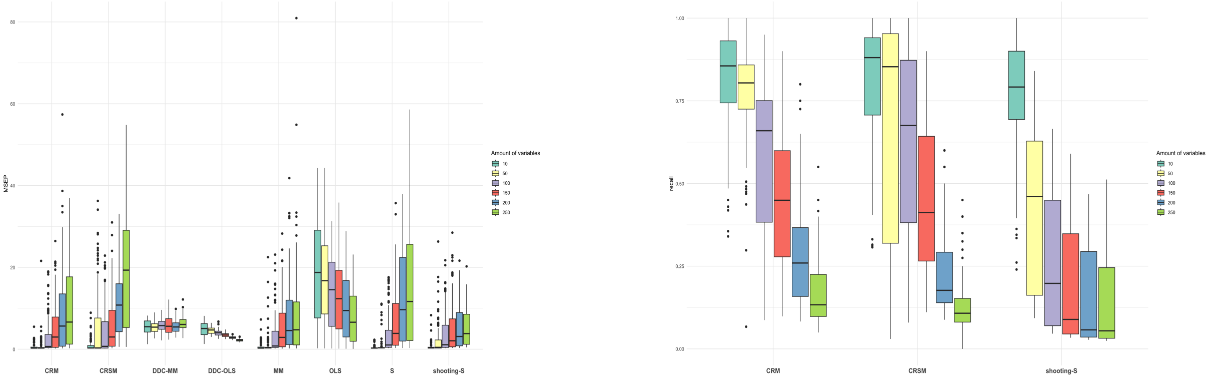


Figure 39: MAE when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

CRM, CRSM, MM and S are all increasing in MSEP when the amount of predictor variables increase, when aggregating over the percentages of casewise contamination. On the other hand, OLS and DDC-OLS decrease when the amount of variables are higher. DDC-MM stays relatively at the same level. These results can be seen in Figure 40. The recall is also shown in this Figure, and there CRM is on average the best for all amount of variables.

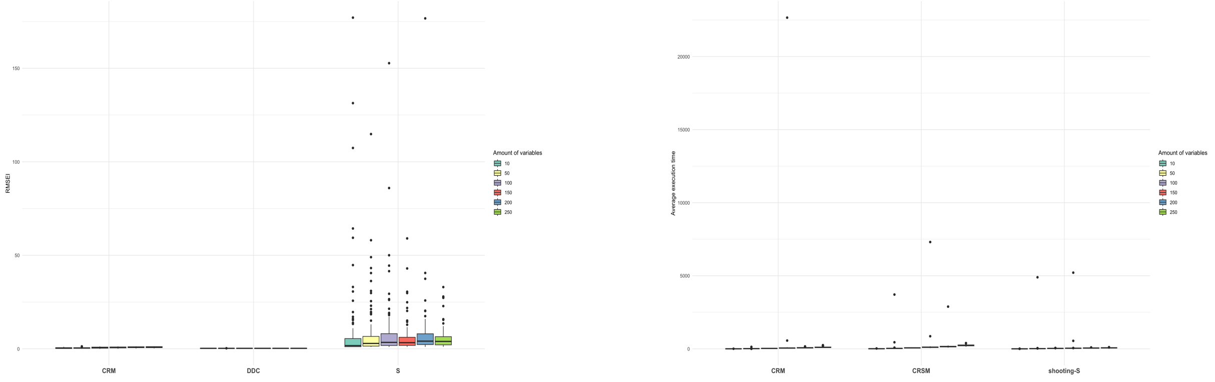


(a) MSEP

(b) Recall

Figure 40: MSEP (a) and recall (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

In Figure 41, for all amount of variables the RMSEI of shooting S is the highest. CRM is slightly higher in terms of RMSEI compared to DDC, but it does not differ much. The average execution time takes long for all three, with some large peaks.



(a) RMSEI

(b) Average execution time

Figure 41: RMSEI (a) and average execution time (b) when the percentage of casewise contamination differs between 0% and 50%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

B.5 Comparison between different percentages of cellwise contamination

In this Section, the results are shown in the case where the percentage of cellwise contamination are 1, 5, 10, 20, 30 and 40%, and the fraction of casewise contamination is fixed at 5%.

B.5.1 Evaluation criteria with 50 predictor variables

The MSEP for CRM, MM, shooting S, S and CRSM stay around the same value in Figure 42, when the percentage of cellwise contamination changes. However, OLS increases rapidly when this percentage increases. DDC-MM and DDC-OLS start at a high MSEP, but decrease when the percentage increases.

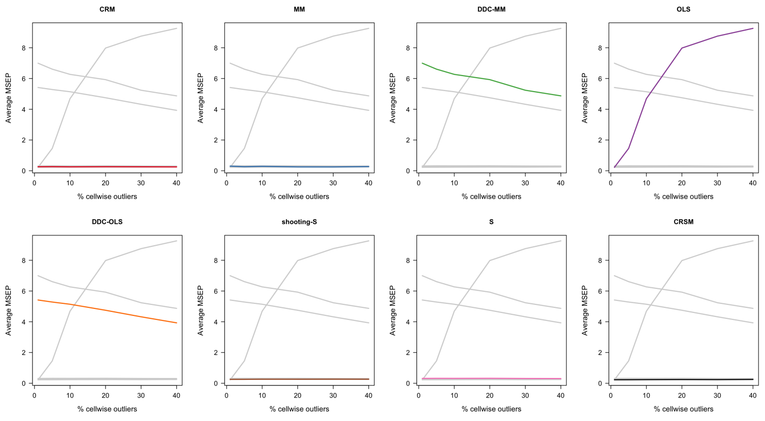
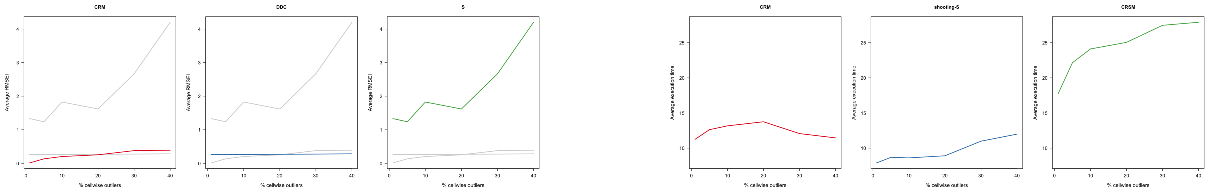


Figure 42: MSEP when the percentage of cellwise contamination differs, where the amount of predictor variables is set to 50

In Figure 43, shooting S is again a higher RMSEI compared to CRM and DDC. The average execution time of CRSM takes the longest for all fractions of cellwise contamination. On the other hand, shooting S is the fastest method, followed by CRM.



(a) RMSEI

(b) Average execution time

Figure 43: RMSEI (a) and average execution time (b) when the percentage of cellwise contamination differs, where the amount of predictor variables is set to 50

B.5.2 Evaluation criteria with 250 predictor variables

In Figure 44 shooting S is the best for all percentages of cellwise contamination, while CRSM performs the worst. CRM, MM, OLS, DDC-OLS and S all increase as the fraction increases. DDC-MM stays at around the same level.

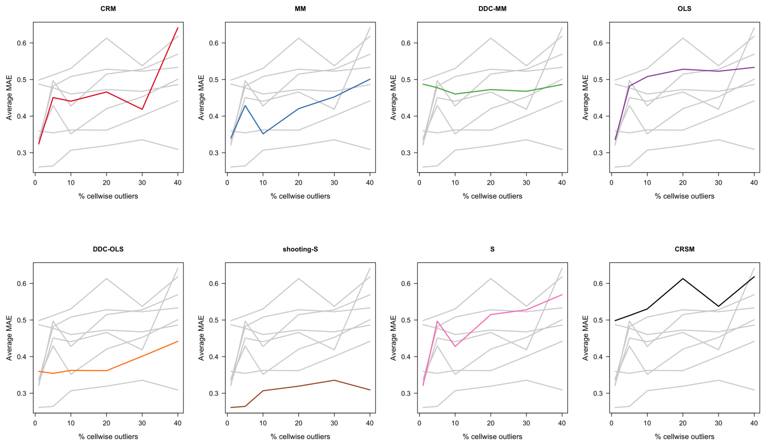
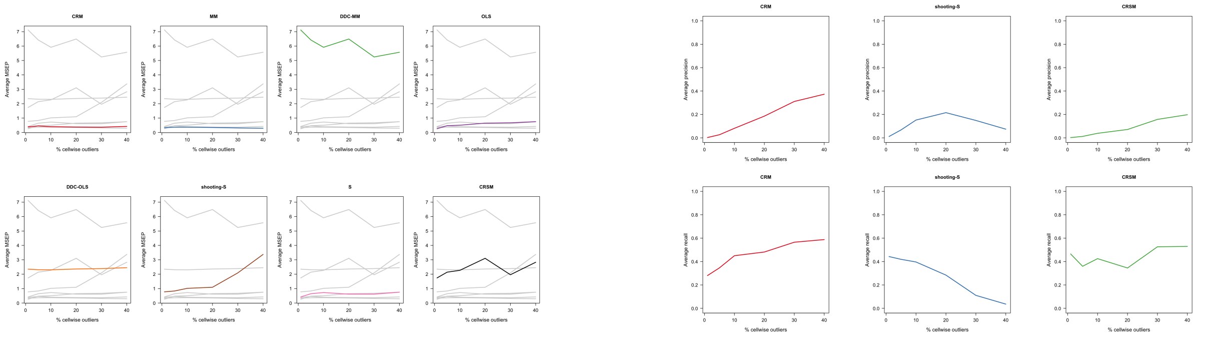


Figure 44: MAE when the percentage of cellwise contamination differs, where the amount of predictor variables is set to 250

For CRM, MM, OLS and S, the predictive performance is around the same, low value as we can see in Figure 45. However, shooting S increases after 20%, whereas CRSM and DDC-OLS have on average around the same MSEP. The precision and recall increase on average for CRM and CRSM. For shooting S the precision starts decreasing after 20%, while the recall is strictly decreasing already from 1%.

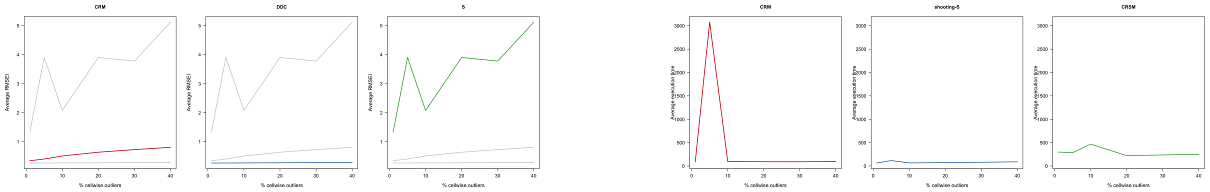


(a) MSEP

(b) Precision and recall

Figure 45: MSEP (a) and precision and recall (b) when the percentage of cellwise contamination differs, where the amount of predictor variables is set to 250

Figure 46 concludes that the imputation for shooting S again has the highest RMSEI for all fractions of cellwise contamination, while the average execution time is the fastest for all fractions. The DDC-imputation is slightly better than CRM. CRM also seems to have a high execution time for fractions until 10%. On the other hand, CRSM only has a little peak at 10%.



(a) RMSEI

(b) Average execution time

Figure 46: RMSEI (a) and average execution time (b) when the percentage of cellwise contamination differs, where the amount of predictor variables is set to 250

B.5.3 Aggregated evaluation criteria

When aggregating over the fractions of cellwise contamination and over the amount of variables, shooting S has the best MAE, as is shown in Figure 47. OLS has the highest value, followed by DDC-OLS. The other methods are around the same MAE values.

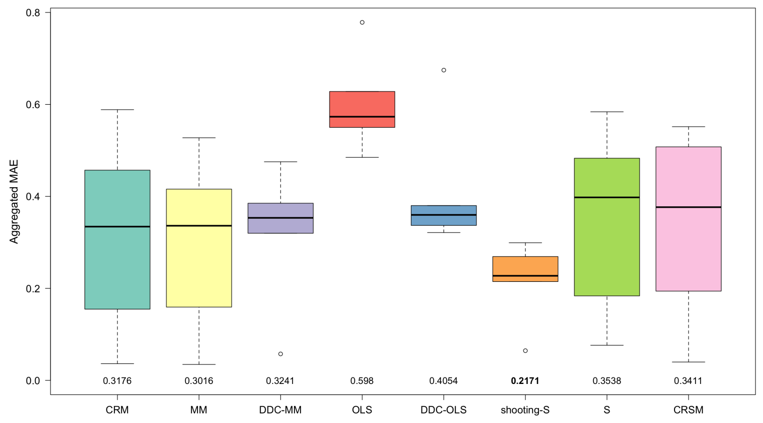
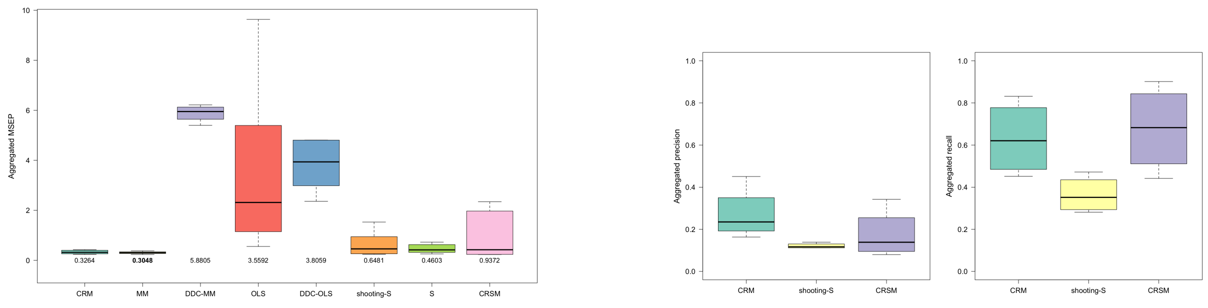


Figure 47: Aggregated MAE when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

MM has the best predictive performance, as can be seen in Figure 48, closely followed by CRM. DDC-MM has the highest MSEP, where DDC-OLS and OLS are also still quite high. The precision and recall of shooting S is the lowest, whereas CRM has the highest precision and CRSM the highest recall.

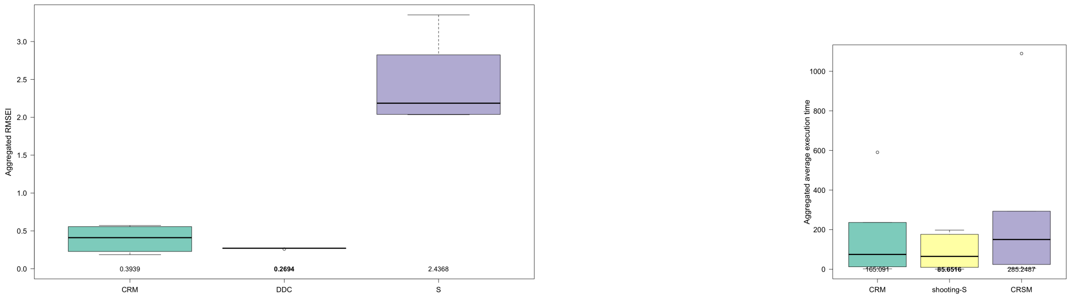


(a) MSEP

(b) Precision and recall

Figure 48: Aggregated MSEP (a) and precision and recall (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

DDC has the best RMSEI, followed by CRM in Figure 49. Shooting S has an average RMSEI that is much higher than these other two imputation techniques. However, again the execution time of shooting S is the best, followed by CRM. CRSM takes the most seconds to complete.



(a) RMSEI

(b) Average execution time

Figure 49: Aggregated RMSEI (a) and average execution time (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

B.5.4 Grouped boxplots

The grouped boxplots where the amount of variables form different subgroups, are aggregated over the fractions of cellwise contamination. It can then be seen in Figure 50 that CRM is the best until 100 variables, after that shooting S performs better. OLS and DDC-OLS are the only ones decreasing as the amount of variables increases. The other methods CRSM, DDC-MM, M and S-regression have around the same biases, which depend slightly on the amount of predictor variables.

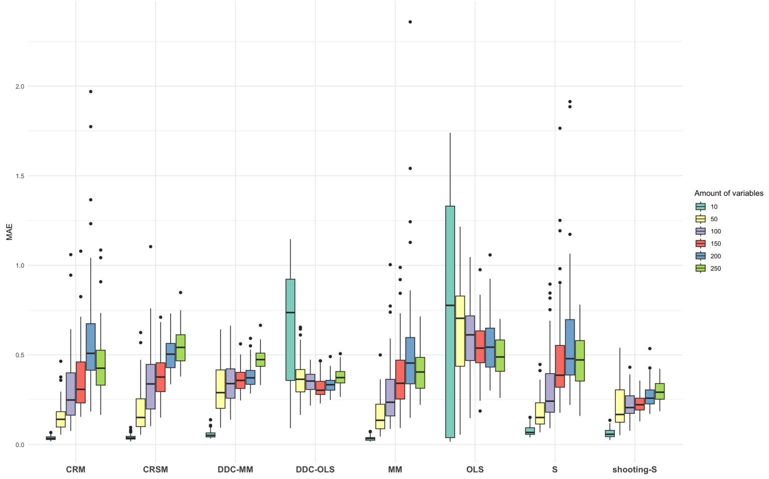
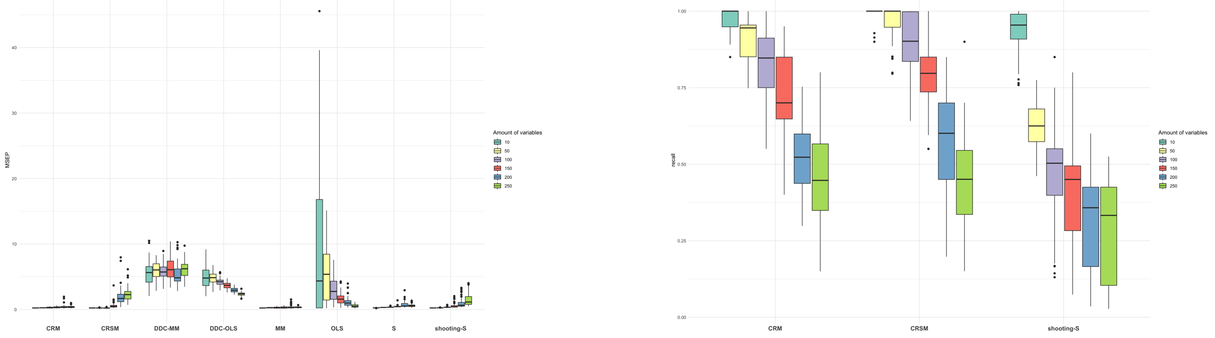


Figure 50: MAE when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

Until 100 variables CRM, CRSM and MM have about the same MSEP, which is visible in Figure 51. After that, CRSM is increasing more in MSEP, meaning that after 100 variables only CRM and CRSM are preferred in terms of predictive accuracy. Besides that, it can be seen that S performs better than shooting S after 150 variables, and DDC-OLS and OLS are again the only ones decreasing, as the amount of variables increases. DDC-MM stays around the same level. In terms of recall, CRSM outperforms CRM and shooting S for all amount of variables, except for 250 variables. In the latter case, CRM is moderately better. Shooting S has the worst rate for recall.

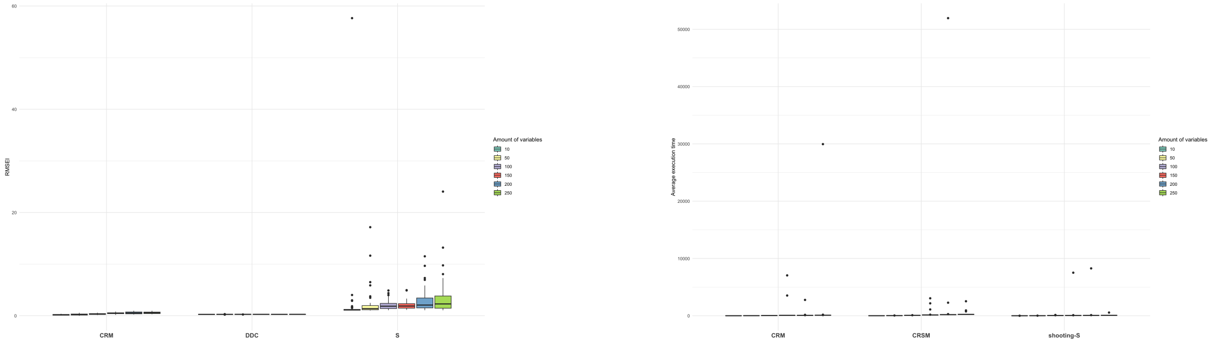


(a) MSEP

(b) Recall

Figure 51: MSEP (a) and recall (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250

As seen multiple times before, the imputation of shooting S is not as good compared to DDC and CRM, which is also visible in Figure 52. The average execution time of CRSM takes longer than CRM and shooting S, and there are some big spikes of simulations that took a lot of time.



(a) RMSEI

(b) Average execution time

Figure 52: RMSEI (a) and average execution time (b) when the percentage of cellwise contamination differs between 1% and 40%, where the amount of predictor variables are 10, 50, 100, 150, 200 and 250