ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus School of Economics

Bachelor Thesis Econometrics and Operations Research

Detecting Deviating Cells

Author:
Diederik Hagenbeek
Student Number:
377614

Supervisor:
A. Alfons
Second reader:
Z. Zhelonkin

July 2, 2017

Abstract

This thesis is a replication and extension to the paper: Detecting Deviating Cells (DDC) by Rousseeuw and Van den Bossche (2016). DDC is a new cellwise outlier detection method, that outperforms all other known outlier detection methods. In addition, the method can replace outliers in a data set by estimated values. The DDC algorithm is analyzed and implemented in the software program MATLAB, in order to replicate and validate the results and implementation of the authors. In addition, the DDC method effects on multi-linear regressions are evaluated. The preliminary conclusion is that the DDC method improves the prediction power of linear regressions, but falls short in comparison to robust regressions.

1 Introduction

Data sets are an important part of scientific research, as results and implications are derived from them. Data sets that consist of multiple variables with a number of observations for each variable are known as multivariate data sets. As data is gathered by observations, e.g. experiments, it may contain outliers. An outlier can be described as an observation that differs considerably compared to other observations of the same variable. An outlier might be an error in the data, caused for example by an incidental flaw in measurement equipment. However if it is not an error, then the outlier could provide the researcher with additional information about his data set. An outlier should therefore not be discarded right away in a dataset if an (visual) inspection is possible to determine its origin.

If standard statistical methods are performed on data sets containing outliers (e.g. regression, mean squared error, variance) the results will be affected by the outliers, as standard assumptions are violated. Implications attributed to such results will therefore be meaningless and have no external validity to them. By identifying the outliers first in a data set, one could decide if the outliers should be discarded or replaced by an estimated value, before standard statistical methods are applied. This would ensure that acquired results are useful. As certain data sets will be too large to inspect visually, a second method to deal with outliers is the use of robust statistics. Robust statistics are less dependent on the underlying distribution of the data and results are less affected by outliers than with standard statistical methods. A trade-off is made between the unbiasedness of classic estimators and the efficiency of robust estimators when outliers are present in data sets. An overview of the history, development and modern-day influence of robust statistics is given by Morgenthaler (2007). The author discusses the implications of outliers on standard statistical methods and the general adaptation nowadays of robust statistics in scientific research. A mathematical background behind robust statistics and the computation of certain robust estimates can be found in Huber (2005).

To inspect outliers visually, one first needs to determine if and where outliers are present in a data set. For now, I will assume that a multivariate data set consists of a matrix, where columns are variables and rows are observations. A comprehensive summary and analysis of outlier detection methods is given by Hodge and Austin (2004). The authors define three approaches when dealing with outliers in data sets: a clustering approach, a classification approach or a novelty approach. Depending on the chosen approach, an appropriate outlier detection method should be chosen. The methods discussed are based on statistical models (i.e. parametric models and non-parametric models), neural networks, machine learning and hybrid systems. However, none of these methods use a cellwise approach to detect outliers. A cellwise method compared to standard outlier detection methods has the advantage that it does not flag entire rows as outliers, when only a small number of cells in that row are actual outliers. The method hereby avoids the pitfall of having too many contaminated rows, which would cause standard outlier detection methods to fail (Lopuhaä and Rousseeuw, 1991). Furthermore, cellwise paradigms could be used to estimate values for outliers by using the relationships between variables. The first introduction of a cellwise outlier detection method was given by Alqallaf et al. (2009). Further contributions were made to develop a cellwise paradigm by Danilov (2010), Van Aelst et al. (2012), Agostinelli et al. (2015), Leung et al. (2016) and Ollerer et al. (2016), according to Rousseeuw and Van den Bossche (2016).

Based upon these contributions, Rousseeuw and Van den Bossche themselves introduced a new outlier detection method called: Detecting Deviating Cells (DDC). It is the first cellwise approach to detect deviating cells in a multivariate data set using the correlations between variables (Rousseeuw and Van den Bossche, 2016). In addition, the method estimates a value for each cell in a matrix using, among others, the same correlations between variables. Furthermore, the method seems to outperform existing methods in terms of a lower rate of misclassified outliers and a lower mean squared error when predicting cell values.[1]

[1] In order to determine the predictive power of the DDC method, values in a data set were purposely replaced by random numbers. After the DDC method was applied to the transformed data set, its predictions for the replaced cells were compared to the original true value of the cell by using the mean squared error.

The purpose of this thesis is to replicate and validate the results of Roussseeuw and Van den Bossche (2016), by implementing their method and applying it to the same data sets they used. If the results are exactly the same, then one can conclude that the authors successfully implemented their DDC algorithm.

In addition to the validation of their results, I will determine the effect of the DDC method on linear regressions of multivariate data sets. By performing regressions on the original data set and on the imputed data set (computed by the DDC method), I can determine the influence of the DDC algorithm by the relative differences in the results. The imputed data set hereby consists of the matrix where all the values have been estimated by the DDC method. Furthermore, I executed linear regressions with the use of robust statistics, i.e. iteratively assigning weight functions to values in variables to minimize the influence of outliers on the results.

The lay-out of this thesis is as follows. In segment 2 the replication of the DDC method of Rousseeuw and Van den Bossche will be discussed, consisting of the actual implementation and an evaluation of the results; segment 3 features an extension to the method where the regressions will be compared to each other using the different data sets.

2 Replication

2.1 Detecting Deviating Cells Method

The focus of this section will be put on the replication of the method of Rousseeuw and Van den Bossche (2016). A brief explanation of the method will follow, continued by an extensive review of each step in the algorithm.

In the continuation of this paper, each time a dataset is mentioned, its layout is structured as columns containing the variables and rows containing observations. Each value hereby in the dataset can be seen as a cell in a matrix, with a corresponding row i and column j. The Detecting Deviating Cells (DDC) algorithm, as described by Rousseeuw and Van den Bossche (2016), essentially compares each column with all other columns to conclude if a particular cell can be considered as an outlier. It does that by first standardizing the data and using the correlations between variables to determine the relationships between cells. The method applies robust statistics in its steps, thereby avoiding the pitfalls of traditional statistical tools when handling data with outliers.

Replicating the implementation of the DDC algorithm is a straightforward process. The method used by the authors is described in their paper and consists of eight steps.[1] To validate the authors results, I will implement each step to create a new program and use the same data sets the authors used. The original algorithm implementation has been written both in the software programs Matlab and R by the authors. In this thesis, the algorithm has been implemented only in Matlab. Although the description of the method can be called extensive, it omits several necessary details in order to replicate the exact results. One can find the omitted details at the bottom paragraph of this page. The complete implementation of the method can be found online at the website of the KU Leuven, including the omitted actions needed to get the authors' results.[2] In order to avoid possible differences in the final results between Rousseeuw and Van den Bossche (2016) and this thesis, the omitted actions were added to my method. As the method used by Rousseeuw and Van den Bossche (2016) is described in detail in their paper, I will repeat some part of their writings in the following section, together with some added clarity and details on certain steps.

In order to perform the eight-step method on a dataset, that dataset should satisfy several requirements. The main structure should consist of a matrix, where the columns are variables and the rows are the observations belonging to the variables. As the DDC method only works on non-categorical and non-discrete data, the data has to be preprocessed first. Furthermore, Rousseeuw and Van den Bossche recommend to transform very non-Gaussian variables, which can be found using QQ plots and histograms.

The first step to filter the data is to remove any categorical data. Next, rows and/or columns containing case or variable numbers are removed from the data set, this second step is not mentioned in Rousseeuw and Van den Bossche (2016). In step 3 and 4, the amount of 'Not a Number' (NaN's) cells in each column and row are counted. If the amount of NaNs divided by the total amount of cells in the corresponding row or column exceeds a predetermined fraction, then that row or column is removed. The authors maintained a cut-off value of 0.5 in their algorithm. Both steps were not mentioned by the authors. In step 5, variables that contain three or fewer different values are removed, this includes binary variables. Finally, columns are removed when more than half the data contains the same value, this step was once more not mentioned by Rousseeuw and Van den Bossche (2016). The complete procedure to filter a dataset can be summarized in six steps and can be found in the appendix. Note that it is important that these steps are followed in consecutive order, otherwise the preprocessed matrix will contain a different set of rows and columns.

[1] Section 4: Detailed Description of the Algorithm (Rousseeuw and Van den Bossche (2016)

[2] Link to file exchange DDC algorithm: https://wis.kuleuven.be/stat/robust/Programs/DDC

Once these actions are carried out, the data has been filtered and the eight-step method can be performed on the matrix. I will refer to the preprocessed matrix as **C**.

Step 1 standardizes each cell of the matrix, using robust estimators to estimate the location and scale of each column. The exact computation of the estimates of location and scale can be found in the appendix of Rousseeuw and Van den Bossche (2016). Among others, the estimators use Tukey's biweight function and the median of each column. All operations are performed column wise and the estimator to calculate the scale uses matrix **C** minus the earlier computed location as input. The matrix **Z** will refer to the newly created standardized matrix. It is important to note that the value for δ to estimate the robust scale is not specified exactly in the paper. Instead of the written down value $\delta = 0.845$, the value used in the algorithm is $\delta = 0.844472$, although the difference is minor, it does influence the final results.

Step 2 determines which values are outliers at first glance, using a cut-off equal to the squared root of the 99^{th} percentile of the χ^2 distribution with one degree of freedom. If the absolute value of a cell is below or equal to the cut-off value, nothing happens. If the absolute value is higher than the cut-off value, the cell is replaced by NaN. The matrix where the cut-off value has been applied to, will be called **U**.

Step 3 determines the bivariate relations between the variables. These will be determined by their correlation with each other using a robust statistic. An initial estimate for the correlation between two variables is given by Gnanadesikan and Kettenring (1972). The authors use the definition of the covariance, expressed in equation (1) and apply a transformation to the variables. Note that the variances inside the covariance equation have been substituted for the robust scale estimate from step 1.

$$cov(Y_1, Y_2) = \frac{1}{4}[(robScale(Y_1 + Y_2))^2 - (robScale(Y_1 - Y_2))^2] \tag{1}$$

The applied transformation consists of standardizing the variables according to step 1, causing the variances to be equal to one for each variable. If the Pearson correlation coefficient (2) is now computed for each pair of standardized variables, the denominator in (2) will be equal to one.

$$\rho_{Y_1,Y_2} = \frac{cov(Y_1, Y_2)}{\sigma_{Y_1}\sigma_{Y_2}} \tag{2}$$

Therefore the Pearson correlation will be equal to the covariance equation expressed in (1) with columns of the matrix **U** as input. The complete equation is given in (3).

$$\hat{\rho}_{U_1,U_2} = \frac{1}{4}[(robScale(U_1 + U_2))^2 - (robScale(U_1 - U_2))^2] \tag{3}$$

The values of the correlations should lie between -1 and 1. The correlation $\hat{\rho}_{j,h}$ defines the shape of an ellipse with its centre at $(0, 0)$ and has a coverage probability equal to the cut-off value in step 2 (Rousseeuw and Van den Bossche, 2016). By calculating the robust distances between two variables, one can determine which cells are inside or outside the ellipse (Hubert and Debruyne, 2010). The equation of the robust distances can be found at (4). The cut-off value used for the robust distances is equal to the 99^{th} percentile of the χ^2 distribution with two degrees of freedom, because of the bivariate relations.

$$RD(X) = \sqrt{(X - \hat{\mu}_{MCD})^t \hat{\Sigma}^{-1}_{MCD} (X - \hat{\mu}_{MCD})} \tag{4}$$

Distances that are below the cut-off value correspond to cells that are kept in each variable, while the other cells are discarded. As one has now determined which cells are inside the ellipse, a new covariance matrix can be computed with the censored variables. Using the Pearson correlation coefficient of equation (2) once more, the robust correlation between two variables is calculated. Step 3 has to be performed for each possible combination of two variables. More details on the correlation method can be found in the appendix of Rousseeuw and Van den Bossche (2016).

A combination of two variables with an absolute correlation value larger or equal to the correlation limit are called 'connected' variables, while variables below the correlation limit are called

’standalone’ variables (Rousseeuw and Van den Bossche, 2016). The correlation limit is set to 0.5 in the DDC algorithm. For the connected columns the slope between the variables is computed using a robust slope estimate, specified in the appendix of Rousseeuw and Van den Bossche (2016).

Step 4 predicts a value for each cell of the matrix $\mathbf{U}$, using the correlations and slopes of the variables from step 3. The predicted values are stored in the matrix $\hat{\mathbf{Z}}$ and the indices of the rows and columns are given by $i = 1, \ldots, n$ and $j = 1, \ldots, n$. For each cell u_{ij} the connected variables j with $h = 1, \ldots, n$ are multiplied by their corresponding value in row i. If a connected variable u_{ih} happens to have a cell value equal to NaN, it is discarded in the calculations. As one now has a vector the size of the connected columns with variable j to predict the single value of u_{ij}, a weighted sum has to be taken over the entire vector. The assigned weights are based on the absolute value of the correlations between column j and the corresponding columns $h = 1, \ldots, n$. Each value is multiplied by its own weight and here after divided by the total sum of the weights, this causes the sum of the vector to be equal to one. In the rare case that a value u_{ij} has no connected columns and is equal to NaN, the value NaN will be kept.

Step 5 deshrinks the predicted values of the new matrix $\hat{\mathbf{Z}}$ from step 4. Each column j of $\hat{\mathbf{Z}}$ is multiplied by the slope a_j and this operation is over all rows. The variable a_j is defined as the slope between the columns j of $\mathbf{Z}$ and $\hat{\mathbf{Z}}$ and is computed for all variables $j = 1, \ldots, n$. The slopes are calculated using the robust slope estimator from step 3.

Step 6 determines the cellwise outliers in the dataset by calculating the cell residuals between the matrices $\mathbf{Z}$ and $\hat{\mathbf{Z}}$. The cell residuals are computed according to equation (5) and are stored in matrix $\mathbf{R}$. If the values of r_{ij} are larger than the 99^{th} percentile of the χ^2_1 distribution, then these cells are flagged as outliers. Thereafter a new matrix $\mathbf{Z}_{imp}$ is created, which contains the same values as matrix $\mathbf{Z}$. However cells that are anomalous according to $\mathbf{R}$ are replaced by the corresponding value of $\hat{z}_{ij}$. Furthermore NaN values are replaced by $\hat{z}_{ij}$ as well.

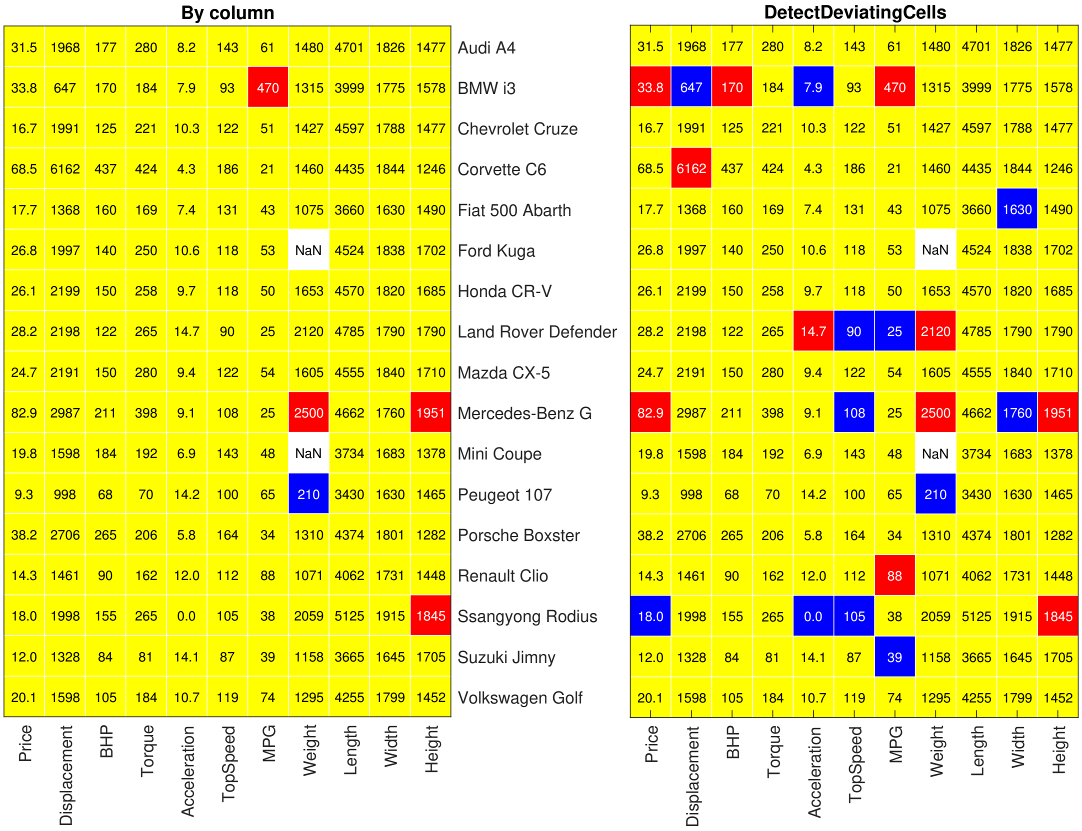
$$r_{ij} = \frac{z_{ij} - \hat{z}_{ij}}{s(z_{ij} - \hat{z}_{ij})} \tag{5}$$

Step 7 determines which rows are outliers in the dataset. The distribution of the residuals from step 6 are close to the standard normal distribution under the null hypothesis of a multivariate normal distributed dataset without outliers (Rousseeuw and Van den Bossche, 2016). As the standard normal distribution is equal to the χ^2_1 distribution, the cdf of the residuals can be calculated using the χ^2_1 cdf. By taking the cdf of this distribution for each r_{ij} and then determining the average of each row of r_{ij}, the criterion T is calculated. The vector T is standardized according to step 1 and values exceeding the cutoff value from step 1 are flagged as rowwise outliers.

Step 8 is the last step in the algorithm and reverses the standardization of the matrix $\mathbf{Z}_{imp}$. By applying step 1 backwards to $\mathbf{Z}_{imp}$, using the robust location and scale estimates of matrix $\mathbf{C}$, the matrix $\mathbf{X}_{imp}$ can be determined. The final outcomes of the algorithm are the locations of cellwise and rowwise outliers, the matrix $\mathbf{C}$, the imputed matrix $\mathbf{X}_{imp}$ and the matrix $\mathbf{R}$ containing the standardized residuals.

2.2 Evaluating the Results

As I have written a new implementation of the algorithm of Rousseeuw and Van den Bossche (2016), I want to compare its results to the authors' results by using the same data sets. If the produced outcomes of my implementation match the results of the authors, then it demonstrates that their eight step method was correctly implemented. The data sets used are named *TopGear*, *Philips*, *Mortality* and *Glass* and are available on the website of the KU Leuven.[1] Each dataset is analyzed by creating cellmaps of the data that indicate outliers. Two outlier detection methods are compared with each other, a standard rowwise or columnwise detection algorithm and the new DDC algorithm. Blue cells indicate outliers with values lower than expected, red cells indicate outliers with values higher than expected and yellow cells indicate cells that are not flagged as outliers. For a more thorough description, I suggest the reader to read section 3 of Rousseeuw and Van den Bossche (2016). The Matlab file *ddcExamples* produces the cellmaps of the data sets and is available on the KU Leuven website as well.[1] Unfortunately the file does not produce cellmaps of the *Philips* dataset, therefore I added certain commands to the file to produce these cellmaps. In order to execute the file, the input given by my DDC algorithm should contain a single structure, where the preprocessed matrix **C**, the matrix **R**, the outlying cells, the outlying rows, the used rows in the analysis and the used columns in the analysis are stored. The created cellmaps are displayed in the Figures 1, 2, 3 and 4. If one compares these cellmaps with the cellmaps in Rousseeuw and Van den Bossche (2016), then it can be concluded that the results are the same. I therefore conclude that Rousseeuw and Van den Bossche successfully implemented their DDC algorithm.



By column

Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height	
31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477	Audi A4
33.8	647	170	184	7.9	93	470	1315	3999	1775	1578	BMW i3
16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477	Chevrolet Cruze
68.5	6162	437	424	4.3	186	21	1460	4435	1844	1246	Corvette C6
17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490	Fiat 500 Abarth
26.8	1997	140	250	10.6	118	53	NaN	4524	1838	1702	Ford Kuga
26.1	2199	150	258	9.7	118	50	1653	4570	1820	1685	Honda CR-V
28.2	2198	122	265	14.7	90	25	2120	4785	1790	1790	Land Rover Defender
24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710	Mazda CX-5
82.9	2987	211	398	9.1	108	25	2500	4662	1760	1951	Mercedes-Benz G
19.8	1598	184	192	6.9	143	48	NaN	3734	1683	1378	Mini Coupe
9.3	998	68	70	14.2	100	65	210	3430	1630	1465	Peugeot 107
38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282	Porsche Boxster
14.3	1461	90	162	12.0	112	88	1071	4062	1731	1448	Renault Clio
18.0	1998	155	265	0.0	105	38	2059	5125	1915	1845	Ssangyong Rodius
12.0	1328	84	81	14.1	87	39	1158	3665	1645	1705	Suzuki Jimny
20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452	Volkswagen Golf

DetectDeviatingCells

Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
33.8	647	170	184	7.9	93	470	1315	3999	1775	1578
16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
68.5	6162	437	424	4.3	186	21	1460	4435	1844	1246
17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490
26.8	1997	140	250	10.6	118	53	NaN	4524	1838	1702
26.1	2199	150	258	9.7	118	50	1653	4570	1820	1685
28.2	2198	122	265	14.7	90	25	2120	4785	1790	1790
24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
82.9	2987	211	398	9.1	108	25	2500	4662	1760	1951
19.8	1598	184	192	6.9	143	48	NaN	3734	1683	1378
9.3	998	68	70	14.2	100	65	210	3430	1630	1465
38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
14.3	1461	90	162	12.0	112	88	1071	4062	1731	1448
18.0	1998	155	265	0.0	105	38	2059	5125	1915	1845
12.0	1328	84	81	14.1	87	39	1158	3665	1645	1705
20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

Figure 1: Cellmaps of the TopGear dataset

The maps display a selection of the total amount of car models. The left part of the figure displays the cellmap where a columnwise detection method has been applied to. The right part of the figure displays the cellmap where the Detecting Deviating Cells (DDC) method has been applied to.

[1] Link to file exchange DDC algorithm: https://wis.kuleuven.be/stat/robust/Programs/DDC

The cell maps illustrate the advantages of the DDC method over columnwise or rowwise outlier detection methods by uncovering the structure of data sets better and flagging outliers per cell instead of entire rows or columns. In contrast to a traditional detection method, the DDC method does not only flag outliers when they deviate strongly from the median of a variable. For example, in Figure 1, the right panel flags considerably more outliers than the left panel, by using the relationships between variables to determine if cells in the same row are anomalous. As an illustration, consider the BMW i3. The columnwise outlier detection method only flags the MPG as an outlier, while the DDC method flags the price, displacement, break horse power and acceleration as outliers, in addition to MPG. As the BMW i3 is an electric car with a petrol engine serving as a backup generator, its properties differ considerably from regular petrol or diesel cars; explaining why a large number of its cells are flagged as outliers. A second case is the Land Rover Defender, where the acceleration, top speed, MPG and weight are flagged by the DDC method, while only the value of weight is considered an outlier by the column wise detection method. As the Land Rover Defender is a heavy off-road vehicle, its large weight, poor acceleration, low top speed and low MPG make perfect sense. However, as off-road vehicles differ considerable from the majority of cars mentioned, such as saloons or hatchbacks, its properties are flagged as outliers. Both examples highlight how the DDC method may provide additional information over a data set to researchers.

Differences between the performances of methods are demonstrated in the cell maps of the French mortality dataset (Figure 3) as well. The outliers visible in the left panel are determined by a rowwise robust method for principle component analysis, while the flagged outliers in the right panel are computed using the DDC method (Rousseeuw and Van den Bossche, 2016). The rowwise detection method can only flag entire rows and therefore uncovers little structure in the dataset, especially compared to the right hand side of the figure. The DDC method flags clusters of cells and exposes patterns in the data that can be related to historical events and modern day health care. Both methods highlight the higher mortality rate during each world war, however the DDC method indicates that this increase was largely caused by the death of adults aged 20-50. Which makes sense, as soldiers were mostly young men. A second interesting pattern can be found in the past couple of decades, where the DDC method indicates that older persons die later than expected. This could be a cause of improved health care and the increased life expectancy it has led to.

The glass dataset, given in Figure 4, consists of glass samples plotted against wavelengths. Highlighted rows in the upper panel are outliers determined by a robust principal components method, while the lower panel is formed by applying the DDC method on the dataset (Rousseeuw and Van den Bossche, 2016). The rowwise outlier detection methods does not uncover any structure in the dataset as entire rows are flagged in the map. If one would use this dataset for further research, then a considerable large part of the rows would have to be removed according to this method. The DDC method on the other hands specifies clusters where glass samples differ significantly from the predicted wavelength. By looking at the specific wavelengths for flagged clusters of glass samples, one can determine the chemical components that contaminated the samples (Rousseeuw and Van den Bossche, 2016).

according to the authors, the results of the DDC method can be used in three ways. The first application is to apply the algorithm on a data set and inspect the outcomes visually to better understand it. A better comprehension might lead to dropping particular rows or columns, apply transformations on the variables or/and change the methods used to collect the data (Rousseeuw and Van den Bossche, 2016). If data sets are too large to inspect visually, the first application cannot be used. As a second application, flagged outliers can be replaced by missing values in a dataset. One then has to find a method that works well with data sets that contain numerous missing values. A possible choice could be the Lasso method (Rousseeuw and Van den Bossche, 2016). The third application replaces the flagged outliers by the algorithm with their estimated values in $\mathbf{X}_{imp}$. This produces no missing values in the dataset.

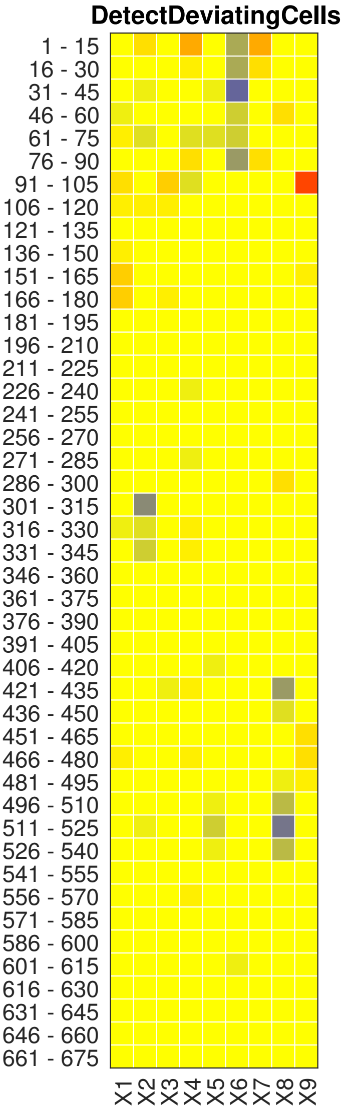


Figure 2: Cellmap of the Philips dataset

Displays the cellmap of a new production line for Philips televisions. The outliers are computed using the Detecting Deviating Cells (DDC) method. The cells are grouped in blocks of 15 x 1 for clarity.

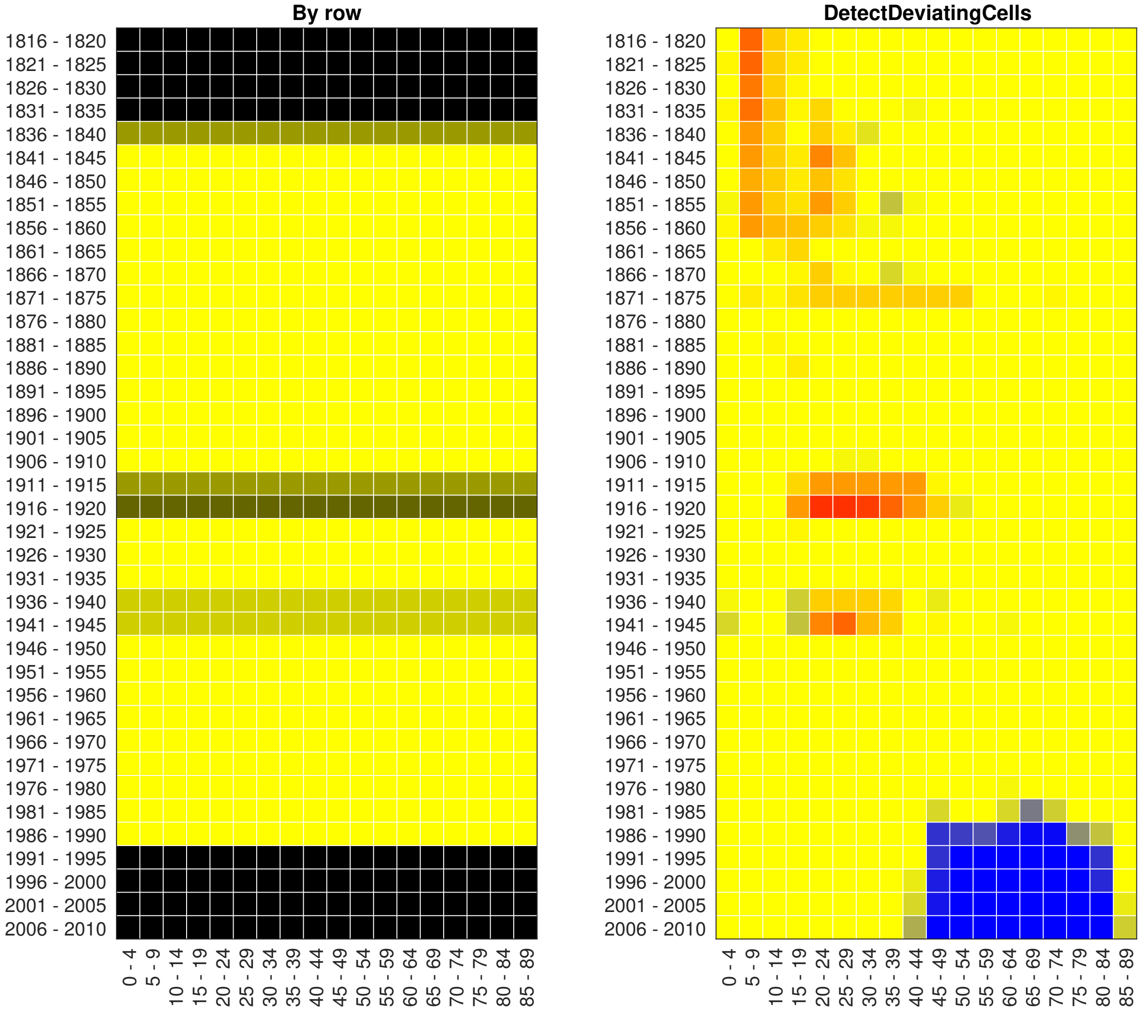


Figure 3: Cellmaps of the French mortality dataset

Displays the mortality rate of French citizens per age category over the period 1816-2010. The left part of the figure displays the cellmap where a rowwise outlier detection method has been applied to. The right part of the figure displays the cellmap where the Detecting Deviating Cells (DDC) method has been applied to. The cells are grouped in blocks of 5 x 5 for clarity. One can see that the DDC method uncovers the structure significantly better than the columnwise outlier method.

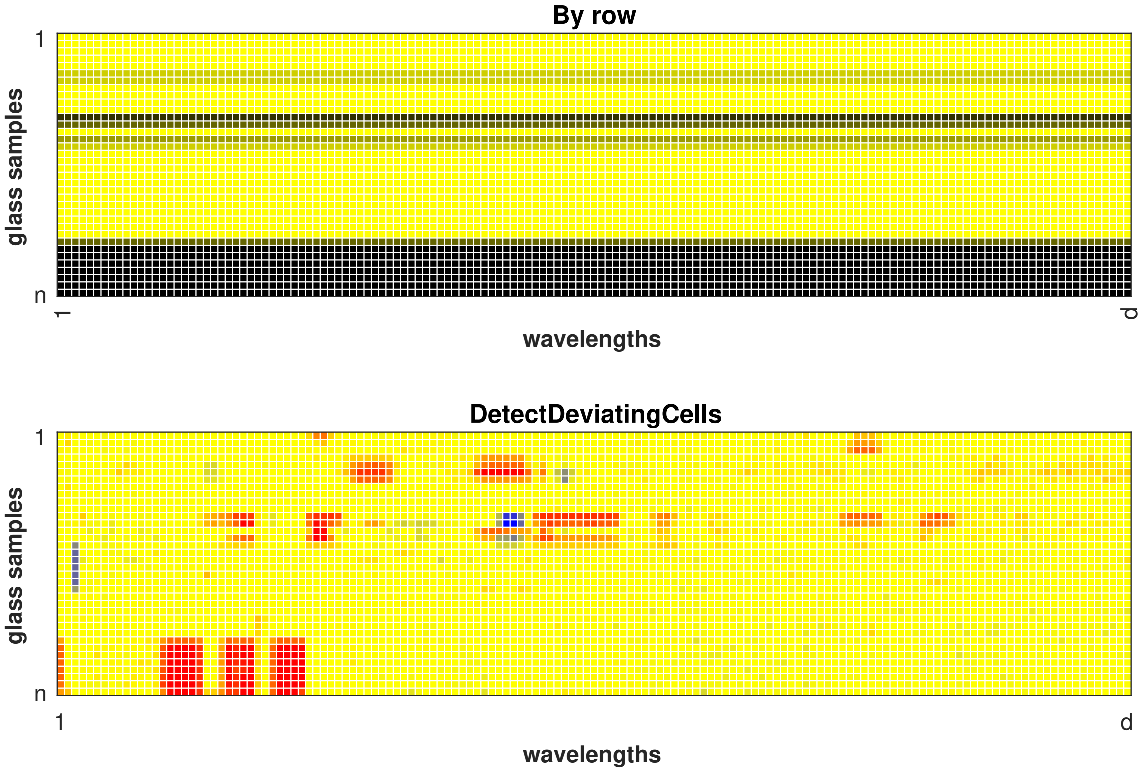


Figure 4: Cellmaps of the Glass dataset

Displays the cellmaps for the different wavelengths of archaeological glass and possible chemical contaminants. The top of the figure displays the cellmap where a rowwise outlier detection method has been applied to. The bottom of the figure displays the cellmap where the Detecting Deviating Cells (DDC) method has been applied to. The cells are grouped in blocks of 5 x 5 for clarity.

3 Extension

The DDC method, proposed by Rousseeuw and Van den Bossche (2016), has demonstrated to uncover the structure of data sets better than conventional outlier detection methods. To illustrate that the DDC method also produces quantitative better results than existing methods, the authors have extensively evaluated their method. In section 5 of their paper, the DDC method is compared to the GY filter of Gervini and Yohai (2002) using two simulated data sets, that are both Gaussian distributed. Both data sets have a mean μ equal to zero, but their covariances Σ differ as one has low correlations and the other has high correlations between its variables. Moreover, certain cells were multiplied by a value gamma to generate contaminated cells, this process was computed randomly (Rousseeuw and Van den Bossche, 2016). Both methods, the GY filter and DDC, were applied to the data sets and the number of non-flagged outliers was recorded. In addition, the estimated values computed for outliers were compared to their true values using the mean squared error. Their conclusions are that the DDC method performs equal or better in all situations compared to the GY filter, but especially outperforms the GY filter when correlations between variables are high. The comparisons were made for a range of different sizes in terms of observations, columns and the value of gamma, and were repeated up to 50 times to get reliable results.

Comparisons were made as well for data sets containing both rowwise- and cellwise outliers. The currently best considered method 2SGS (Rousseeuw and Van den Bossche, 2016) to detect outliers in such situations was compared to the DDC-2SGS method, a variation of 2SGS where the GY filter is replaced by the DDC method. Once more, the DDC-2SGS outperforms the 2SGS method when variables in data sets have a high correlation with each other, while low correlations produce equal results.

The DDC method seems to be the best method to replace current outlier detection algorithms based on the previous results. In order to investigate the practical implications of the method besides the detection of outliers, I want to find out its effect when using it in combination with linear regressions. Linear regressions are a popular statistical method to relate variables to each other, however results are very sensitive to outliers. This arrives from the fact that least squares uses linear functions to determine the value of the coefficients of the dependent variables (Morgenthaler, 2007). A single value strongly deviating from all other values could therefore influence the estimates by a disproportional amount. To study the effects of the DDC algorithm on the regressions, I have chosen two data sets: the TopGear data set previously used and a Breast Cancer diagnostic set. Both data sets satisfy the requirements of the DDC method to a great extent, as most variables contain non-categorical values and have more than 3 discrete values in each variable. The TopGear dataset is made up of 297 instances and 33 variables, and contains characteristics of car models. The dataset of breast cancer is made up of 32 variables and contains 569 instances.[1] One variable, the patient id number is removed, as it does not contain any predictive power. The remaining attributes contain characteristics of a patient's breast mass and a diagnosis, which state if cells are malignant or benign.

3.1 Methodology to compare linear regression

In order to investigate the influence of the DDC method on the performance of regressions, the data will be split into a set containing the explanatory variables $\mathbf{X}$ and a set including the dependent variable $\mathbf{Y}$. In the TopGear dataset, the prices of the car models will be the dependent variable. While the diagnosis of the fine needle aspirate test is the dependent variable in the breast cancer dataset. As the diagnosis of the test was either Malignant or Benign, the data is transformed to a 0/1 variable, where malignant is set equal to zero and benign equal to one. Furthermore, a training and a test set are created, using a percentage ratio of 80/20 to evaluate my estimates of the β coefficients. The linear regressions will be computed on the original data and on the imputed data matrix. When applying the DDC method to the training data set, the dependent variable will be included as well. For the test set, the imputed matrix by the DDC method will be executed with-

[1] The breast cancer dataset and explanation of each variable is publicly available at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

out including the dependent variable $\mathbf{Y}$. If the dependent variable would be included, this would influence the estimations of the dependent variables. By not including the dependent variables, the independent relationship between test and training data is maintained.

The DDC algorithm specifies the assumption that datasets are in essence Gaussian distributed, I therefore will apply transformations on non-Gaussian variables, taking the log of variables or use the Box-Cox transformation. If the data appears to be distributed Gaussian, the implementation of the DDC algorithm will be applied on the data to identify the outlying cells.

Rousseeuw and Van den Bossche (2016) did not develop or describe an automated method that would transform non-Gaussian variables in a data set. This could cause a problem if the number of variables in a data set would be too large to inspect manually. Therefore, I have developed a method in this thesis, that transforms non-Gaussian variables automatically by using the p-values of a chosen normality test. The first step in the algorithm is to determine if the original variable is Gaussian distributed. If one has to reject the hypothesis that the variable is Gaussian-distributed, then a transformation will be applied using the Box-Cox transformation or taking the logarithm of the variable. The step hereafter is to compute the p-values of the transformations and the non-transformed variable, using the same normality test. By now comparing all the p-values, the maximum p-value determines which applied transformation to the variable should be kept. In addition, the method has the option to trim the data by choosing a lower and upper percentile. This minimizes the presence of large outliers influencing the results of the normality test.

In my case, the Anderson-Darling test and the Lilliefors test provided the best results as a normality test for the two data sets. To determine the optimal normality test in general, a theoretical study should be carried out. Due to a limit in the amount of time, this was not investigated in this thesis. The new method is implemented in MATLAB and can be found in the file exchange of this thesis.

Besides performing a standard linear regression on the original data, I will also be using robust statistical methods to better fit the data and compare its results with the imputed matrix from the DDC algorithm. If one executes multi-linear regressions in MATLAB, there is the option to use robust statistics to estimate the coefficients of the regressors. The algorithm uses iteratively re-weighted least squares with an user specified weighting function.[1] To estimate the coefficients of the regressions, a M-estimator is used by the implemented method. A M-estimator minimizes the standard residuals from least squares, given in equation (6). The minimization problem is given in equation (7), where the function ρ assigns weights to each residual and σ is the scale estimate. To minimize the equation, the derivative is taken and set equal to zero (see equation (8)). The ψ function hereby denotes the derivative of the ρ function that assigns weights to the observations, where Tukey's bisquare is often chosen as default (Ruckstuhl, 2014). Other weight functions one can specify for the ψ function in MATLAB are: Andrews, Cauchy, Fair, Huber, Logistic, Talwar and Welsch.

$$r_i(\hat{\beta}) = y_i - x_i^T\hat{\beta} \tag{6}$$

$$\sum_{i=1}^{n} \rho\Big(\frac{r_i(\hat{\beta})}{\sigma}\Big) \tag{7}$$

$$\sum_{i=1}^{n} \psi\Big(\frac{r_i(\hat{\beta})}{\sigma}\Big)x_i = 0 \tag{8}$$

$$f(\beta_{MM}) = \frac{1}{n}\sum_{i=1}^{n} \rho\Big(\frac{r_i}{\hat{\sigma}}\Big) \tag{9}$$

Although the M-estimator is a robust estimator in regressions, it is not the most reliable estimator one can achieve. This has to do with the fact that equation (8) can have multiple solutions

[1] More information on robust regression and its weight functions can be found at: https://www.mathworks.com/help/stats/robustfit.html

which differ in efficiency. The S-estimator is equal to the M-estimator, however it computes the minimum of the scale estimates σ to get more reliable estimates. A drawback though, is that the S-estimator is statistical inefficient. A better estimator for robust regressions would be the Modified M-estimator (MM-estimator), which combines the high resistance to outliers from a S-estimator and the high efficiency of a M-estimator (Ruckstuhl, 2004). A possible drawback of MM-estimators could be the increased computation time compared to other estimators, though this is not a problem for the sizes of the data sets used in this thesis. The minimization problem for a MM-estimator is given in equation 9, where f is minimized for β. MM-estimators are not part of the standard implementations of MATLAB, however they are included in the Flexible Statistics and Data Analysis (FSDA) toolbox.[1]

For the robust regressions on both data sets, I used M-estimators as well as MM-estimators. The eventual chosen weight function for the M-estimators was based upon the estimator with the lowest mean absolute prediction error. For the MM-estimator Tukey's bisquare weight function was chosen.

To evaluate the predictive power of each regression, the mean squared prediction error (MSPE) and median absolute prediction error (MAPE) will be computed. To compute the prediction errors, the predicted dependent variable for the test set was compared to the true value of $\mathbf{Y}$ in the test set. By taking the median of the sum of the absolute differences between the predicted and true value of $\mathbf{Y}$, the MAPE is computed. And, by taking the mean of the sum of the squared differences between the predicted and true value of $\mathbf{Y}$, the MSPE is determined.

If any outliers are present in the data sets, the median absolute prediction error should provide us with more reliable results in terms of the errors than the mean squared prediction error. This has to do with the fact that the mean in the MSPE is not a robust statistic, while the median is.

Therefore the trimmed MSPE is computed as well by removing a percentile of the lowest and highest values of the squared differences, using percentile combinations: 10/90, 5/95 and 2.5/97.5. These values are chosen by trial and error. The trimmed MSPE could provide information on the amount of outliers present in the data sets, by having large differences between its values for different percentiles. To compare the different values of the MSPE and MAPE with each other, I will discuss their relative differences. On all the dependent variables for the same data set, either none or all the same transformations are applied, in order to make a fair comparison. Furthermore, the regressions are not optimized in any way to fit the data better, that is, no variable selection, cross terms or other methods were used to increase the performance of the regressions. This does not matter, as one wants to know the relative differences between the methods, not the absolute performance of each regression.

I set up three hypotheses in order to test my predictions, where $\mathbf{X_{org}}$ represents the training set with the original values and the matrix $\mathbf{X_{org}}$ represents the training set with imputed values estimated by the DDC method. The hypotheses are given below:

Hypothesis 1 The MAPE for predicting the same dependent variable is significantly lower for linear regressions using the matrix $\mathbf{X_{imp}}$ as explanatory variables than using the matrix $\mathbf{X_{org}}$ as explanatory variables.

Hypothesis 2 The MAPE for predicting the same dependent variable is significantly lower for linear regressions using the matrix $\mathbf{X_{imp}}$ as explanatory variables than robust regressions using a M-estimator with the matrix $\mathbf{X_{org}}$ as explanatory variables.

Hypothesis 3 The MAPE for predicting the same dependent variable is significantly lower for linear regressions using the matrix $\mathbf{X_{imp}}$ as explanatory variables than robust regressions using a MM-estimator with the matrix $\mathbf{X_{org}}$ as explanatory variables.

[1] The FSDA toolbox can be downloaded at the website: http://www.riani.it/MATLAB.htm. A description of the main documentation page is available at: http://www.riani.it/MATLAB/FSDA/index.html.

3.2 Results

Breast Cancer Dataset

To determine if the dataset Breast Cancer is approximately Gaussian distributed, a histogram and Q-Q plot are taken of each variable. A large part of the data set turned out to be heavily skewed, therefore it was decided to transform multiple variables by taking the logarithm or applying a Box-Cox transformation. The automated method, discussed in the methodology, produced better results than the visual inspection of the variables in terms of median absolute prediction errors. Therefore, the reported results on Breast Cancer are based on the transformations performed by the automated method. In this case, the Anderson-Darling test was specified as the normality test to compute the p-values and no variables were trimmed in the method. Besides these transformations, values equal to zero were replaced by NaN in the 30 explanatory variables, as these values should be impossible to get. This last step was taken before any transformations were applied to the data.

Linear regressions are performed on the dependent variable $\mathbf{Y}$, the diagnosis of a patient, and on the explanatory variables $\mathbf{X}$. The regressions are computed for the test and training set of $\mathbf{X}$ and $\mathbf{Y}$, where the 80/20 ratio is taken. In order to determine the effect of the DDC algorithm on the regressions, a comparison is made between different versions of the explanatory variables. I compared the original values of $\mathbf{X_{org}}$ with the estimated values of $\mathbf{X_{imp}}$, where the latter matrix was computed by the DDC algorithm. Furthermore, robust regressions were carried out on the training and test data using $\mathbf{X_{org}}$, but never on the imputed matrix $\mathbf{X_{imp}}$. The weight function assigned to the robust regressions using M-estimators was Tukey's bisquare, as this gave the lowest MAPE. The MM-estimator used Tukey's bisquare weighting function as well.

In Table (1) the MAPE, the MSPE and the trimmed MSPE, for the percentile combinations 10/90, 5/95 and 2.5/97.5, are given for the different predictions of $\mathbf{Y}$ in the Breast Cancer data set. Here, $\mathbf{Y}$ was computed by multiplying the β coefficients of the training set with the explanatory variables $\mathbf{X}$ of the test set. The variables $\mathbf{X_{robust\text{-}M}}$ and $\mathbf{X_{robust\text{-}MM}}$ represent the robust regressions using the M-estimator and the MM-estimators respectively.

If one looks at the values in the table, one can see that the MAPE of $\mathbf{X_{imp}}$ is significantly lower than the MAPE of $\mathbf{X_{org}}$, with a difference of approximately 19%. This confirms hypothesis 1, which states that the MAPE of linear regressions using $\mathbf{X_{imp}}$ as regressor is lower compared to using $\mathbf{X_{org}}$ as regressor. The MAPE of $\mathbf{X_{imp}}$ is 19% lower than the MAPE of $\mathbf{X_{org,robust\text{-}M}}$ as well, confirming hypothesis 2. The results of the original data and the robust regression are the same, indicating that the use of the robust M-estimator did not increase prediction performance. The MAPE of $\mathbf{X_{org,robust\text{-}MM}}$ is lower than the MAPE of $\mathbf{X_{imp}}$. Therefore hypothesis 3 will be rejected for the Breast Cancer data set. The use of a robust MM-estimator outperforms the estimates of the DDC algorithm by a substantial amount, the MAPE is approximately 81% lower. Based on this data set, one could better use a robust regression with a MM-estimate to predict a dependent variable, if one would have no interest in detecting the outliers. Table (1) furthermore reveals that the MSPE decreases substantially (about 10% to 60%) when the trimmed data for the percentile 2.5/97.5 is compared to the non-trimmed MSPE. This could indicate that large outliers are present in about 5% of the data. To illustrate the outliers in the Breast Cancer dataset, a cellmap of the outliers is given in Figure (5), which is based on the flagged outliers by the DDC algorithm.

TopGear

Rousseeuw and Van den Bossche (2016) specified which transformations one has to apply to the variables in the TopGear data set, in order to get approximately Gaussian distributed variables. They consist of taking the logarithm on the following variables: *price*, *displacement*, *break horsepower*, *torque* and *top speed*. The automated method, earlier discussed in the methodology to transform non-Gaussian variables, produced better results in terms of the median absolute prediction errors than the transformations suggested by the authors did. Therefore, the results in the table for the TopGear data set are acquired with the use of transformed variables by applying the automated method. Instead of using the Anderson-Darling test, the Lilliefors test was specified for the normality test this time, without the use of trimming any variables.

The multi-linear regressions are performed on the dependent variable $\mathbf{Y}$, the price of car models, and on the explanatory variables $\mathbf{X}$, characteristics of the cars. The regressions are computed for the test and training set of $\mathbf{X}$ and $\mathbf{Y}$, where the 80/20 ratio is taken. In order to determine the effect of the DDC algorithm on the regressions, I compare them using different regressors: the matrix $\mathbf{X_{org}}$ and the matrix $\mathbf{X_{imp}}$. The DDC method is applied on both the dependent variable and the independent variables in the training set, but only on the independent variables in the test set. The values of the dependent variable therefore did not influence the independent values in the test data and vice versa. The robust regressions both used Tukey's biweight function, as this again led to the lowest prediction errors.

In Table (2) the MAPE, the MSPE and the trimmed MSPE, for the percentile combinations 10/90, 5/95 and 2.5/97.5, are given for the different predictions of $\mathbf{Y}$ in the TopGear data set. Here, $\mathbf{Y}$ was computed by multiplying the β coefficients of the training set with the explanatory variables $\mathbf{X}$ of the test set. The variables $\mathbf{X_{robust\text{-}M}}$ and $\mathbf{X_{robust\text{-}MM}}$ represent the robust regressions using the M-estimator and the MM-estimators respectively. For all entries in the table, the logarithm was taken to decrease the size of the values.

The MAPE of the regression using $\mathbf{X_{imp}}$ is lower than the MAPE of $\mathbf{X_{org}}$, however the relative difference is smaller than 1%. Therefore, there is no significant evidence to either reject or accept hypothesis 1. The MAPE of $\mathbf{X_{robust\text{-}M}}$ is considerably lower than the MAPE of $\mathbf{X_{imp}}$, roughly 20%. I therefore have to reject hypothesis 2, which states that regressions using $\mathbf{X_{imp}}$ instead of $\mathbf{X_{robust\text{-}M}}$ provide better predictions for the dependent variable. The difference between the MAPE of $\mathbf{X_{robust\text{-}MM}}$ and $\mathbf{X_{imp}}$ is large and significant, approximately 22%. The third hypothesis therefore has to be rejected as well.
The differences between the MSPE and the trimmed MSPE are small, which could indicate that there are only a few outliers with very small/large values in the data set or that more than 20% of the data consists of outliers which causes the trimmed MSPE not to decrease in value.

Conclusion

The results of both data sets contradict each other on certain levels. In the Breast Cancer data set, the first and second hypothesis could be accepted, while the third hypothesis was rejected. In the TopGear data set, the first hypothesis could not be accepted, and the second and third hypothesis were rejected. It is therefore hard to determine any conclusions on the effect of the DDC method on regressions. However, there seems to be evidence that the impute matrix leads to better prediction results than the matrix with the original values. This could be expected, as outliers in the original data can influence the coefficients in linear regressions. The effect was present in the Breast Cancer data set, but it could hardly be measured in the TopGear data set. The effect should therefore be treated with caution. In addition, the robust regression using MM-estimators provided better results than any other regression in both data sets. This implies that using a robust regression with a MM-estimator always leads to better results, if outliers are present in the data set. However, identifying and inspecting outliers can provide a researcher with additional information on his/her data set, which the robust methods cannot do. None of these conclusions can be proven at this point, because the theoretical framework misses and the number of data sets was too small.

Tables of the Results

Table 1: Values of the Mean Squared Prediction Error (MSPE), trimmed MSPE and the Median Absolute Prediction Error (MAPE) for the Breast Cancer data set

	$\mathbf{X}_{\mathbf{org}}$	$\mathbf{X}_{\mathbf{org,\ robust\text{-}M}}$	$\mathbf{X}_{\mathbf{org,\ robust\text{-}MM}}$	$\mathbf{X}_{\mathbf{imputed}}$
MAPE	0.1753	0.1753	0.0269	0.1422
MSPE	0.0595	0.0595	0.1876	0.0465
$\mathbf{MSPE}_{\mathbf{97.5}}$	0.0516	0.0516	0.0786	0.0426
$\mathbf{MSPE}_{\mathbf{95.0}}$	0.0483	0.0483	0.0648	0.0401
$\mathbf{MSPE}_{\mathbf{90.0}}$	0.0430	0.0430	0.0269	0.0364

Table 2: Values of the Mean Squared Prediction Error (MSPE), trimmed MSPE and the Median Absolute Prediction Error (MAPE) for the TopGear data set

	$\mathbf{X}_{\mathbf{org}}$	$\mathbf{X}_{\mathbf{org,\ robust\text{-}M}}$	$\mathbf{X}_{\mathbf{org,\ robust\text{-}MM}}$	$\mathbf{X}_{\mathbf{imputed}}$
MAPE	10.116	8.0137	7.8215	10.044
MSPE	20.907	17.316	17.010	20.346
$\mathbf{MSPE}_{\mathbf{97.5}}$	20.647	16.769	16.267	20.278
$\mathbf{MSPE}_{\mathbf{95.0}}$	20.483	16.649	16.082	20.211
$\mathbf{MSPE}_{\mathbf{90.0}}$	20.258	16.500	15.897	20.147

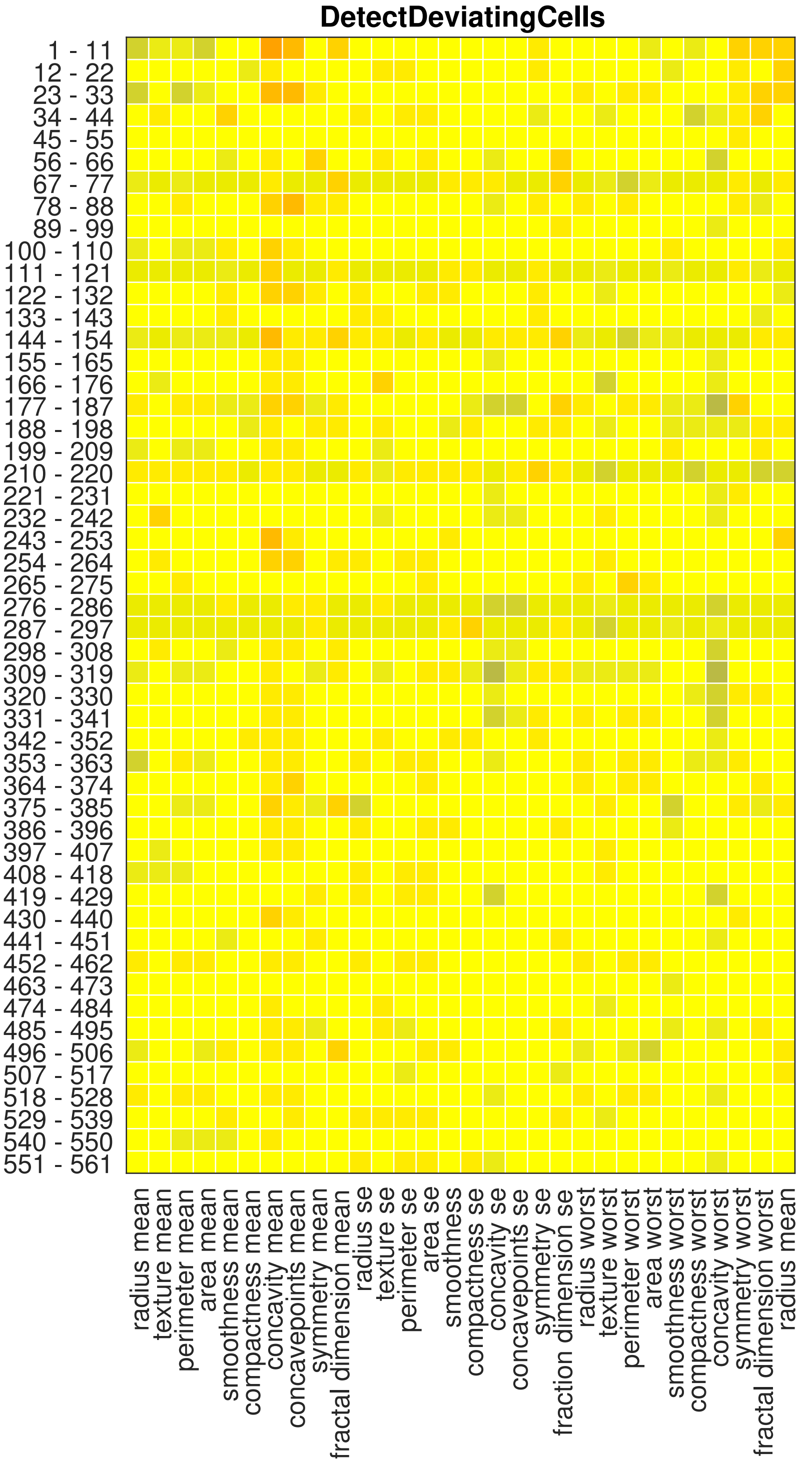


Figure 5: Cellmap of the Breast Cancer dataset

Displays a cellmap for different characteristics of patients' breast cells. The cellmap was created by applying the Detecting Deviating Cells (DDC) method to the Breast Cancer dataset. The cells are grouped in blocks of 11 x 1 for clarity.

References

[1] Agostinelli, C., Leung, A., Yohai, V.J., & Zamar, R.H. (2015), Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination, Test, 24, 441-461.

[2] Alqallaf, F., Van Aelst, S., Yohai, V. J., & Zamar, R. H. (2009). Propagation of outliers in multivariate data. The Annals of Statistics, 311–331.

[3] Danilov, M. (2010), Robust estimation of multivariate scatter in non-affine equivariant scenarios, Ph.D. dissertation, University of British Columbia, Vancouver.

[4] Gnanadesikan, R., & Kettenring, J. R. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. Biometrics, 81–124.

[5] Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial intelligence review, 22(2), 85–126.

[6] Huber, P. J. (2005). Robust statistics (Vol. 579). John Wiley & Sons.

[7] Hubert, M., & Debruyne, M. (2010). Minimum covariance determinant. Wiley interdisciplinary reviews: Computational statistics, 2(1), 36–43.

[8] Morgenthaler, S. (2007). A survey of robust statistics. Statistical Methods & Applications, 15(3), 271–293.

[9] Leung, A., Zhang, H., and Zamar, R. (2016), Robust regression estimation and inference in the presence of cellwise and casewise contamination, Computational Statistics and Data Analysis, 99, 1-11.

[10] Lopuhaä, H.P., & Rousseeuw, P.J. (1991), Breakdown points of affine equivariant estimators of multivariate location and covariance matrices, The Annals of Statistics, 19, 229-248

[11] Öllerer, V., Alfons, A., & Croux, C. (2016), The shooting S-estimator for robust regression, Computational Statistics, to appear.

[12] Rousseeuw, P. J., & Van den Bossche, W. (2017). Detecting deviating data cells. Technometrics, (just-accepted).

[13] Ruckstuhl, A. (2014). Robust Fitting of Parametric Models Based on M-Estimation. Lecture notes.

[14] Van Aelst, S., Vandervieren, E., & Willems, G. (2012), A Stahel-Donoho estimator based on huberized outlyingness, Computational Statistics and Data Analysis, 56, 531–542

Appendix

The DDC method uses the following six steps to filter any data set before the algorithm implementation is performed. The order of the presented steps is important and should be followed consecutively.

Step 1: Remove any categorical columns from the data set.
Step 2: Remove any rows and/or columns containing only the case or variable number from the data set.
Step 3: Count the amount of NaN values in each variable and divide it by the length of that variable. If the ratio is above a predetermined value, the variable should be removed from the data set. The predetermined fraction in Rousseeuw and Van den Bossche (2016) was set equal to 0.5.
Step 4: Count the amount of NaN values in each row and divide it by the width of that row. If the ratio is above a predetermined value, the row should be removed from the data set. The predetermined fraction in Rousseeuw and Van den Bossche (2016) was set equal to 0.5.
Step 5: Remove discrete variables if they contain the same or a lower amount of different values than a predetermined number. Rousseeuw and Van den Bossche (2016) maintained a cut-off value equal to 3.
Step 6: Remove columns where more than 50% of the data contains the same value.