Identifying Outliers in a Regression Framework

STAT 844: Statistical Learning - Function Estimation

Bradley Hedges

Spring 2020

1. Introduction

Data collection often results in outliers. These outliers can be from one-off events or errors in the data collection, but when we are pulling data from some statistical distribution, there is a chance we will naturally encounter outliers. Identifying outliers is decidedly important in regression analysis, due to their potential influence on the resulting model. These points may have unreasonable impact on modelling parameters, precision, and how well the model works in prediction [1]. Outliers also may give a statistician important insights into the data. Perhaps an outlier corresponds to a unit or process that optimizes the response and should be correspondingly rewarded [1].

Consider the toy data set generated in R from the model $y = x + 7$:

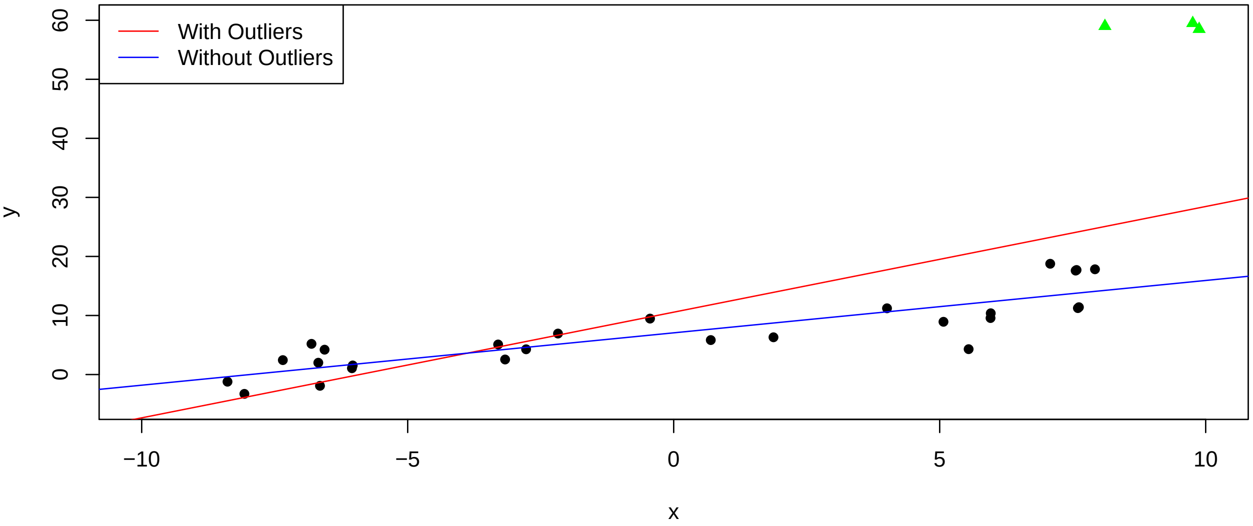


Figure 1: Effect of outliers on ordinary least squares fit

Figure 1 clearly illustrates the influence of outliers (green triangles) on an Ordinary Least Squares (OLS) fit. Without the outliers, the data is fit quite well. Our with-outlier model exaggerates the response for x-values closer to the outliers, which corresponds to higher model error. Though the outliers are easily identified in this toy problem, that might not always be the case, and we often cannot remove the outliers before fitting.

As we have seen throughout the course, classical regression methods (most notably Ordinary Least Squares) are particularly sensitive to outliers and influential points. We have investigated some methods for mitigating outliers, such as changing the loss function to a more robust one, but these methods come with some caveats.

First, in robust regression, the outliers are never actually identified. How do we know if a robust loss function is necessary? Is the eyeball-test rigorous enough to justify complicating our regression techniques? Furthermore, robust regression gives no metric to quantify how significant the outliers are.

Perhaps the most well-known technique for identifying outliers comes in the form of boxplots, which were introduced by Tukey in 1977 [2]. This test, however, should be seen as informal, and used in the preliminary stages of data analysis to better understand the data. Using Tukey's boxplot will flag at least 30% of samples taken from a normal distribution as containing an outlier [2]. Boxplots further do not quantify how significant the outliers are, or what to do with outliers once they are identified. Though a boxplot gives some insights given there is a single categorical variable, it loses meaning when tackling a continuous variable, or multivariate problem.

Thus, identifying outliers is important to mitigate model bias and understand the impact of this data on a model. In most cases, we cannot simply remove or ignore the outliers unless we know they were found with fault data collection techniques. Throughout this paper, we will investigate some classical techniques to identify and quantify outliers, as well as look at some techniques to model the data while accounting for the outliers. The techniques for identifying outliers will include Cook's Distance, DFFITS, and identifying multiple outliers with hierarchical clustering through analyzing standardized residuals and predicted values. To model our data, we will be investigating some choices of loss function compared against the robust regression methods discussed in class.

2. Data

Depending on the complexity of the problem, we will be using a few different data sets. The most simple, which will mainly be used with the purpose of introducing new topics, is the $y = x + 7$ data. There will be two versions of this data set: One with a single outlier, and one with a cluster of outliers. This data includes 50 x-values pulled from a uniform distibution with values in $[-10, 10]$. The corresponding response (y-values) is generated by passing these x-values through the given function. Each y-value is then shifted by a value drawn from a $N(0, 16)$ distribution. Finally, 3 outliers were added to the data set at around $x \approx 9.5$. The data was initialized with $set.seed(123456)$ in R.

The second data set is slightly more complex than the first. It contains multiple clusters of outliers and is taken from the function $y = x^2 + 2x + 7$. The data is manipulated in a very similar form as the previous data set. This data set has three noticeable outlying clusters at around $x = -2, 1.5, 4$. Plotting the data gives a better understanding of the outliers:

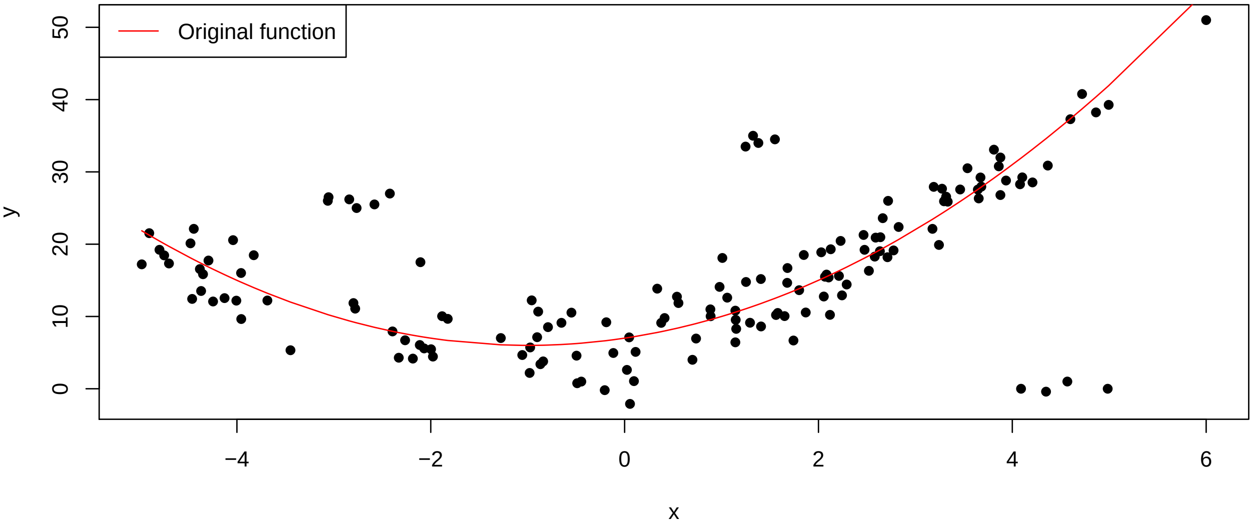


Figure 2: Data drawn from y = xˆ2 + 2x +7

The third data set involved taking the first data set and shifting the first "cluster" upward. This fixed the data into three distinct groups. The purpose of this data set can be found in section 3.4. Because this data is very similar in its construction to that of the first data set, and because a plot of the data is shown later, we will not go into significant detail in this section.

3. Outlier Identification

3.1 Cook's Distance

In 1977, Dennis Cook introduced a new metric for classifying the influence in a data set [3]. Cook's metric not only identifies outliers, but quantifies the relative impact of points on a least squares fit. The metric is fittingly called Cook's distance. Consider then the standard linear regression model:

$$Y = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.1.1}$$

With $\boldsymbol{\beta}$ a $p \times 1$ vector, and X an $n \times p$ matrix of observations. Suppose $\boldsymbol{\varepsilon} \sim N(0, \sigma^2)$ are independent random variables, and define $H = X(X^TX)^{-1}X^T$ to be our hat matrix. In 1975, Huber noted that outliers typically correspond to large values of the H-matrix [3], so our distance metric will involve this H. Cook's distance for the i^{th} point in a data set is defined as follows:

$$D_i = \frac{(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i)^T X^T X (\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i)}{ps^2}. \tag{3.1.2}$$

In equation (3.1.2), $\widehat{\boldsymbol{\beta}}$ is our LS estimate of $\boldsymbol{\beta}$ with all the data, and $\widehat{\boldsymbol{\beta}}_i$ is our estimate of $\boldsymbol{\beta}$ with the i^{th} data point removed. We define $s^2 = R^TR/(n-p)$, where R is the covariance matrix of residuals defined in (3.1). D_i can be compared against a $F_{p,(n-p)}$ distribution to quantify the magnitude of the distance $\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i$.

Under (3.1.2), larger values of D_i correspond to more influential points (outliers). It might be obvious that the more influential a point, the bigger our difference $(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i)$, and so D_i is larger. Notice that X^TX is necessarily positive-semidefinite, so $(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i)^T X^T X (\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i) \geq 0$.

An equivalent form of D_i that relates to the V matrix can be written as:

$$D_i = \frac{t_i^2 w_i}{p}, \tag{3.1.3}$$

with $t_i = r_i/(s\sqrt{1-h_{ii}})$, and $w_i = h_{ii}/(1-h_{ii})$, where h_{ii} is the i^{th} diagonal element of H. It is obvious from (3.3) that D_i is large (corresponding to an influential point), if h_{ii} is large.

Cook acknowledges some downsides to the statistic. In general, influential points near the boundary of the data set are harder to detect than those in the interior. Furthermore, since D_i is a decreasing function of p, it is harder to find influential points in complex models [3].

One might now wonder how to quantify an influential point. A rule-of-thumb cutoff point in literature is the line $4/(n-p)$ [4]. Note that an influential point might not be an outlier, so we would like to see if Cook's distance can differentiate between the two.

Consider the $y = x^2 + 2x + 7$ data discussed earlier. We will implement Cook's distance in R to find any influential points by fitting a quadratic to the data. Note that we could fit any degree polynomial, and use Cross Validation to choose the degree. Since we are not investigating polynomial fit, we will just choose quadratic for convenience.

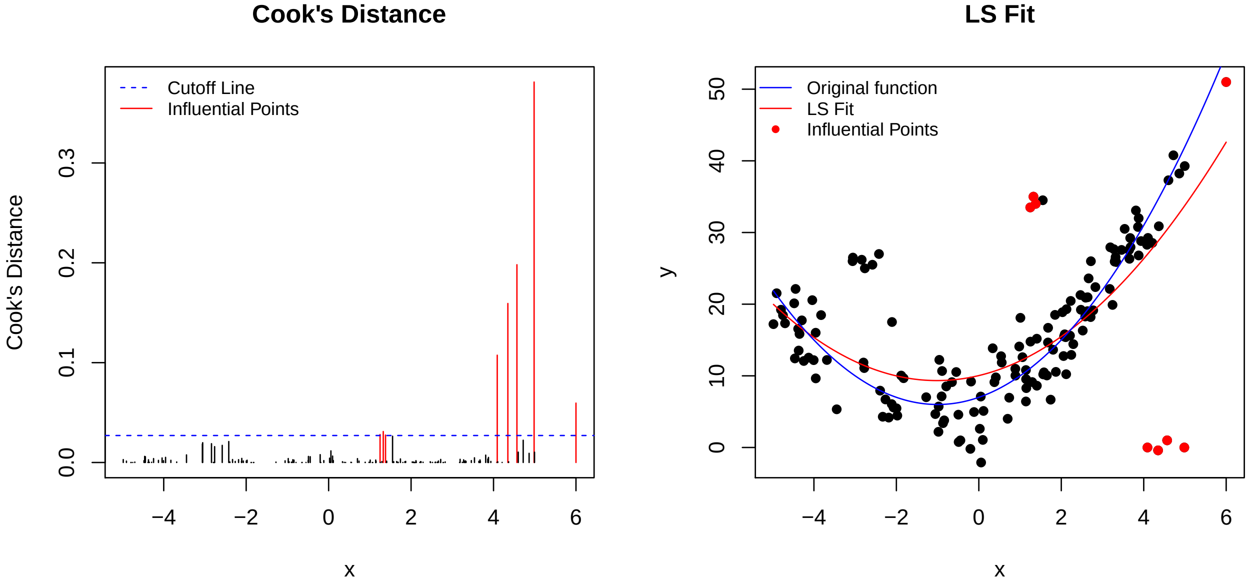


Figure 3: The left plot shows the Cook's distance for each of the points. The right plot shows the corresponding influential points plotted against the LS fit.

We can see from Figure 3 that Cook's distance identifies 8 influential points. These points generally coincide with an eyeball-test of outliers with a few exceptions. We might expect the cluster around $x \approx -2.5$ to be outliers, but they are not according to the $4/(n-p)$ rule-of-thumb. Looking at $x = 1.5$, we see a cluster of what we would again expect to be influential points (particularly outliers). Based on our cutoff criteria, the last point in the cluster is not an influential point, but the other points are. Again demonstrating that this method is imperfect. Finally, we have a point at $x = 6$ that is defined as influential. Notice that it lies very close to the curve $y = x^2 + 2x + 7$, and so we wouldn't call it an outlier. From this we see that Cook's distance only identifies points that strongly influence the LS fit, and not necessarily outliers. As such, once the influential points are identified, choosing the outliers from this is up to the descretion of the investigator.

We might also be interested in how well the LS model fits the data if we remove the influential points chosen by Cook's distance. From a practical standpoint, it likely doesn't make sense to simply remove the data, but for the sake of qualifying the impact of these points, we do so in Figure 4.

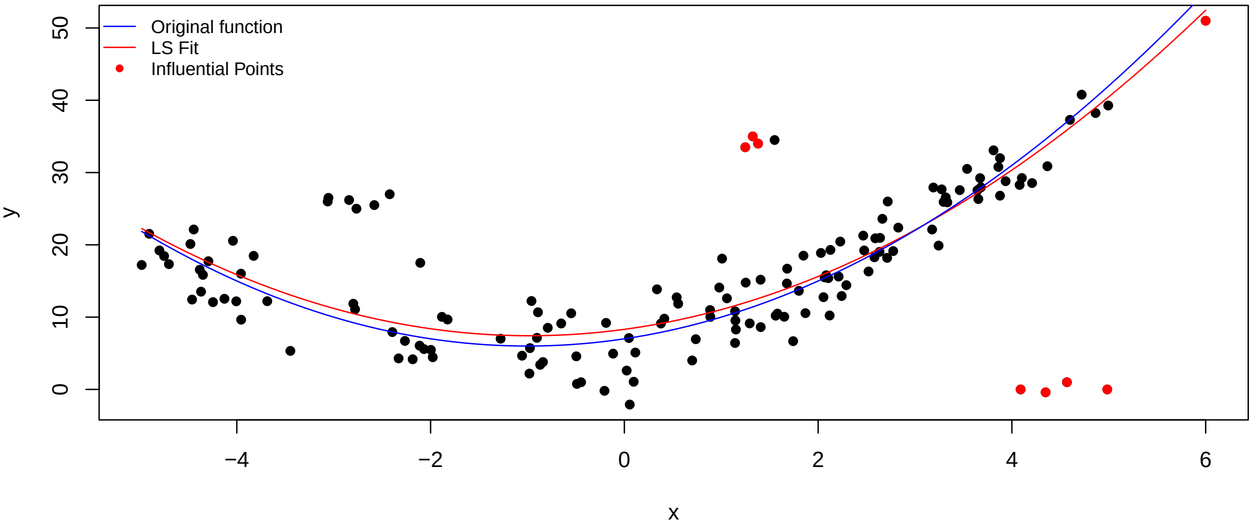


Figure 4: LS Fit with Influential Data Removed

As we expect, Figure 4 shows that removing the influential points, most of which we expect to be outliers, results in a LS fit that looks very similar to the original function. Once again, we cannot indiscriminately remove all (or perhaps any) of our influential points, but we now have a better understanding of how much these points change a fit.

3.2 DFFITS

We cannot simply take Cook's Distance at face-value; we must compare it to other well-defined metrics to see if they give similar results. As such, we will briefly introduce and analyse another metric: DFFITS. Introduced by Belsley et al. in 1980 [5], DFFITS uses a similar technique to Cook's Distance where a point is removed, and the relative perturbation of the regression fit is documented. We define this metric as:

$$\mathrm{DFFITS}_i = \frac{\widehat{y}_i - \widehat{y}_i^{(-i)}}{\widehat{\sigma}_i\sqrt{h_{ii}}}, \tag{3.2.1}$$

where $\widehat{y}_i$ is the predicted response at x_i from the full regression model, $\widehat{y}_i^{(-i)}$ is the predicted response at x_i from a regression model generated from the data set with the i^{th} element removed, h_{ii} is the i^{th} diagonal element of the previously mentioned hat matrix, and $\widehat{\sigma}_i$ is the standard error with the i^{th} element removed. Some manipulation of this equation, and taking

$$t_i = \frac{y_i - x_i^T\widehat{\boldsymbol{\beta}}^{(-i)}}{\widehat{\sigma}_i\sqrt{h_{ii}}},$$

then equation (3.2.1) can be written as

$$\mathrm{DFFITS}_i = \sqrt{\frac{h_{ii}}{1-h_{ii}}}t_i. \tag{3.2.2}$$

Under this form, we can clearly see that for large h_{ii} or large t_i, DFFITS is similarly large. We recall that when h_{ii} is large, it generally corresponds to an influential point. Furthermore, t_i will be large when the model is significantly perturbed by removing a point [5]. Once again, we do not yet have a cutoff for which a point is considered influential. Belsley et al. recommend considering a point influential when $|\mathrm{DFFITS}_i| \geq 2\sqrt{p/n}$ [5], where p and n are defined in (3.1.1).

One now might wonder why we might choose DFFITS instead of Cook's Distance. The two equations seem very similar, and in fact, we can write:

$$\mathrm{CD}_i = \frac{\widehat{\sigma}_i}{k\widehat{\sigma}}\mathrm{DFFITS}_i^2, \qquad (3.2.4)$$

as provided by [5].Despite the similarities between these two metrics, DFFITS is considered the better choice [5] because DFFITS gives more information about the estimated variance of the data, and simultaneously computes the influence of a point on β and variance [5]. We can compare DFFITS to Cook's distance with our toy data set taken from $y = x^2 + 2x + 7$ with the following plots:

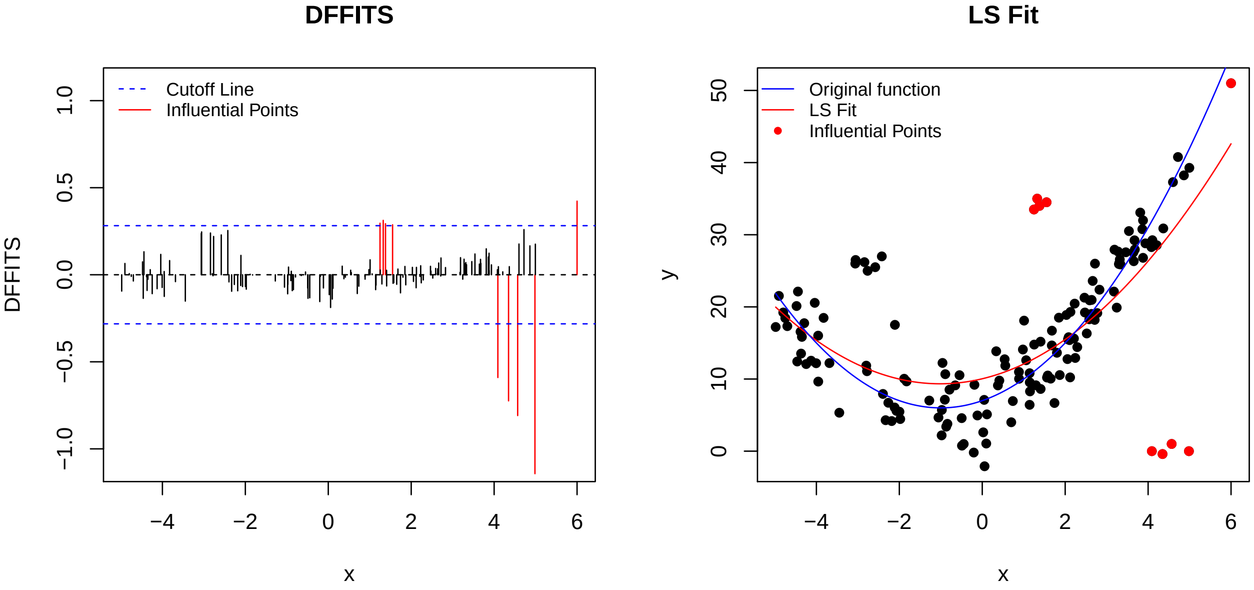


Figure 5: The left plot shows DFFITS with outlier coloured red, and the corresponding outliers are plotted with the LS fit in the right plot

In comparing Figures 3 and 5, we see that Cook's distance and DFFITS choose many of the same points. The only difference comes in the cluster around $x \approx 1.5$. With an eyeball test, we would expect this cluster to be influential. Cook's distance rejects the last point in the cluster as it just below the threshold. With DFFITS, this value is slightly above the threshold.

As with Cook's distance, DFFITS does not discriminate between in-lying influential points and outliers. Furthermore, according to Sebert et. al. [1], Cook's distance was introduced to work well in regression problems with a single outlier. The same is true of DFFITS [5]. Both algorithms work by systematically removing one point at a time, and therefore are not effective when a data set has multiple influential points. This is particularly limiting, as regression data sets often have multiple outliers that cannot easily be identified individually. To mitigate this restriction, we therefore might be interested in a technique to deal with multiple outliers.

3.3 A Clustering Algorithm

The outlier analysis techniques investigated in sections 3.1 and 3.2 are particularly limiting in their inability to deal with multiple outliers. Sebert et al. introduced a clustering algorithm for outlier detection with the intent of solving this problem [1]. This clustering method uses hierarchical clustering with standard euclidean distance d(x_1,x_2). Euclidean distance is an obvious choice because it is easy to interpret as a measure of similarity [1].

This $d(x_1, x_2)$ is used to measure the distance between all observations using the standardized predicted values and standardized residuals that we get from the LS fit [6]. This algorithm uses hierarchical clustering whereby the n data points are initially partitioned into n clusters, and then the clusters are merged until all data is in a single cluster. Single linkage is one type of hierarchical clustering algorithm and is often referred to as “nearest neighbour” clustering because of the one-at-a-time joining of clusters [1].

Any hierarchical linkage algorithm will technically work, and depends on the structure of the data. If they appear to form elongated structures, single-linkage is effective because of its ability to chain data points (a property that some mathematicians say is a negative). If the data typically forms elliptical clusters, we might use a different clustering algorithm, such as complete-linkage [1].

The algorithm then uses a modified form of Mojena’s stopping rule (outlined in appendix) to identify the optimal number of clusters [9]. It then calls the largest cluster the “inliers”, and considers the rest of the data “outliers” [7]. The algorithm according to reference [1] follows:

1. Standardize predicted values and residuals from LS fit. The standardization is done by calculating $z_i = (x_i - \overline{x})/s$, where $\overline{x}$ is the mean of the observations and s is the standard deviation.
2. Compute $d(x_1, x_2)$ for all pairs of standardized predicted values and residuals. Cluster the points according to a hierarchical clustering algorithm.
3. Using the modified Mojena’s Stopping Rule, cut linkage tree at height $h + 1.25s_h$, where h is the mean cluster height, and s_h is the standard deviation of the heights.
4. Find group that has the largest number of data points. This is the set of inliers, all other data are outliers.

Consider the following 30 simulated data points, which look to be split into 3 distinct groups:

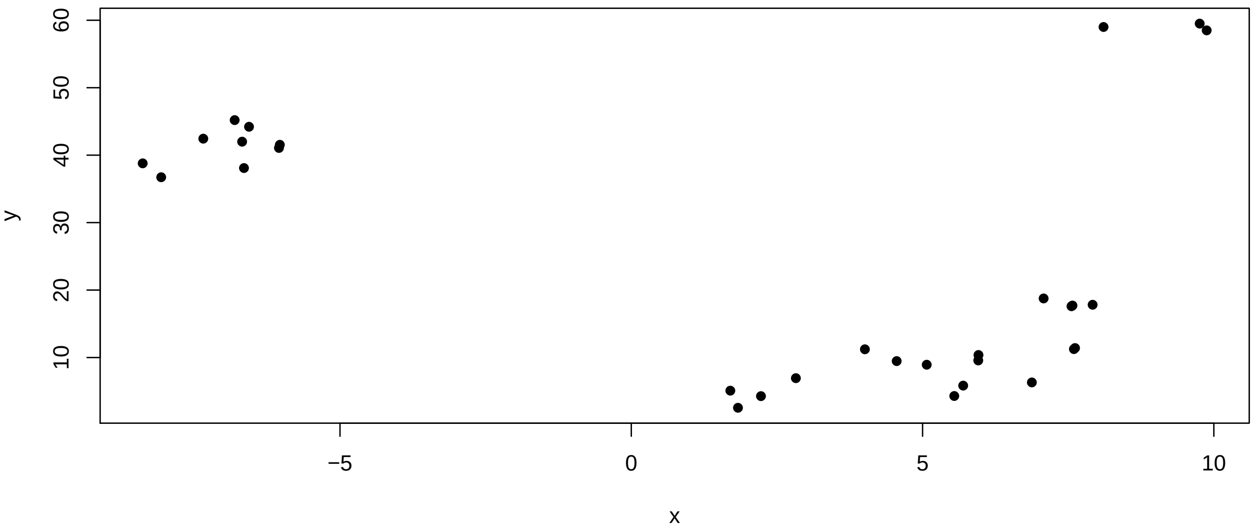


Figure 6: Clustered Data for mutliple outlier analysis

Here, the data seems to form a distinctly elongated, elliptical structure. We will therefore use single-linkage clustering algorithm. We can apply the Sebert et al. clustering algorithm to produce a plot of the corresponding clustering tree:

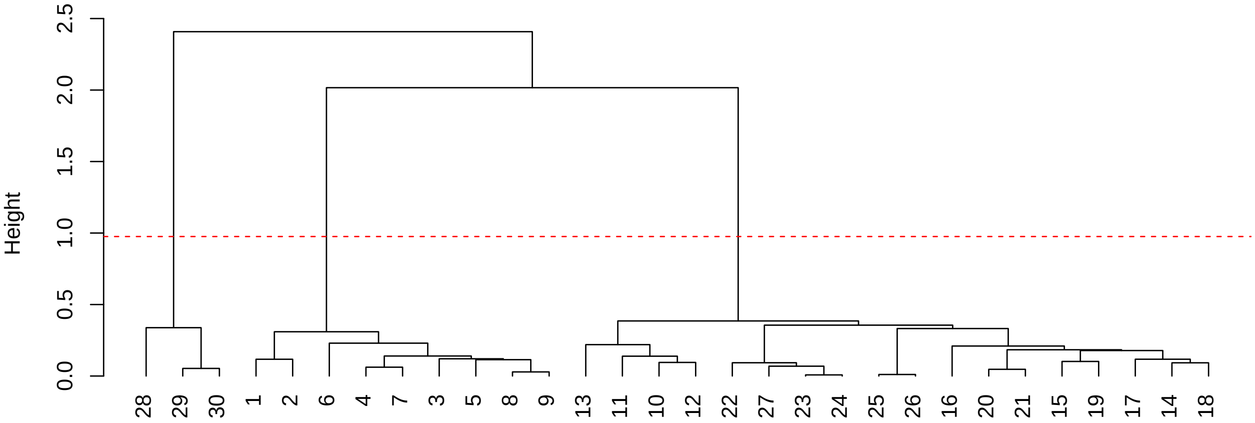


Figure 7: Single-linkage clustering Tree with Mojena Cutoff

Investigating the tree shows us that the algorithm splits the data into three classes: Class 1 has points 1 : 9, class 2 contains points 28 : 30, and class 3 contains points 10 : 27. Class 3 is clearly the largest class, so according to our algorithm, we call that class the "inliers" and all other points are "outliers". The data is therefore split according to Figure 8:

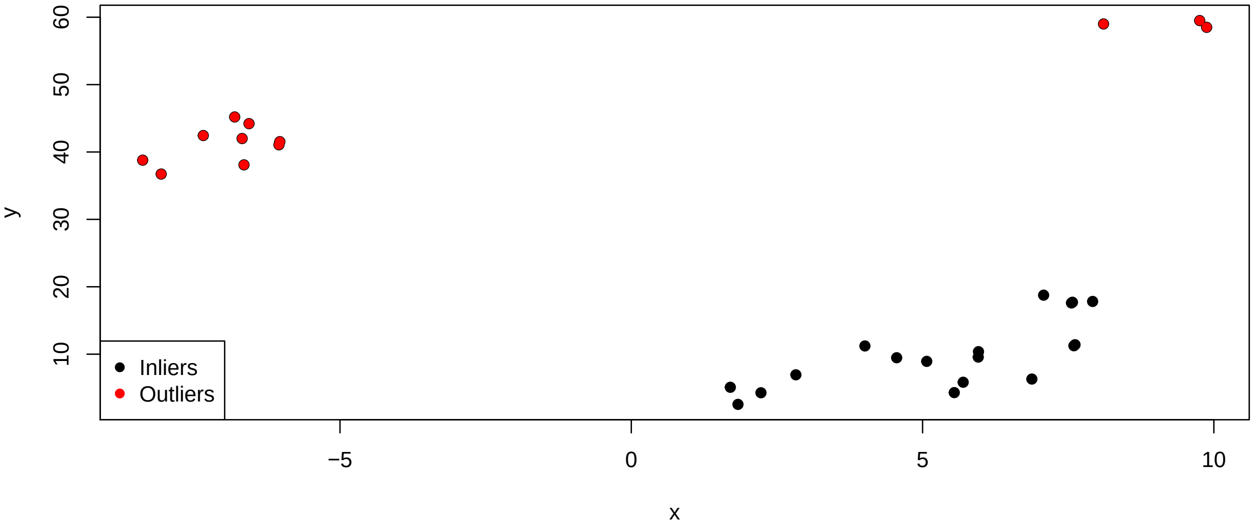


Figure 8: Data partitioned according to Sebert et al. algorithm

Since this algorithm was designed to combat the issues in the Cook's distance metric (namely its inability to correctly identify multiple clusters of outliers), it makes sense to compare the results of the algorithm to that of Cook's distance. For the sake of space, we will not investigate DFFITS on this data, as it would likely give the same results as Cook's distance, and suffers from the same drawbacks as Cook's distance.

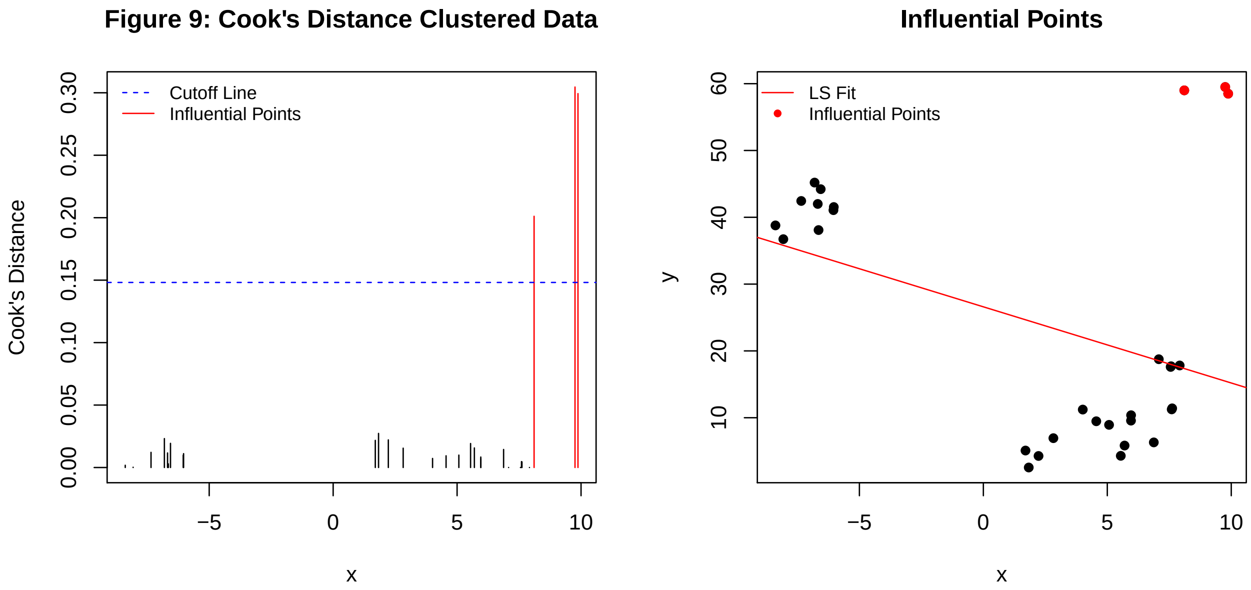


Figure 9: The left plot shows the Cook's distance for the Cluster data, and the right plot shows the corresponding outliers with the data

Figure 9 gives us results similar to that of literature: Cook's distance is a poor metric for identifying multiple outliers, and the clustering algorithm mitigates that problem. Our algorithm identified an entire cluster of outliers that Cook's distance was unable to identify.

3.4 Robust Regression through choice of Loss function

Thus far, we have only discussed some methods to identify influential points and from them extrapolate the outliers. As was previously mentioned, unless we know that the identified outliers correspond to data that was found with faulty data collection, we cannot simply remove that data. The benefit of identifying outliers comes after our models are built, when we draw conclusions and do prediction. There are, however, techniques to mitigate the influence of outliers. In the following section, we will look at choice of loss function through the lens of robust regression. Note that the purpose of this paper is to quantify outliers rather than implement the knowledge in our model. Because of this, and for the sake of space, this section will be brief.

In least squares regression, we are finding a model to minimize square error loss. A significant downside of square error loss is that the error blows up for significant outliers, so these outliers tend to dominate the model parameters. Square error is used so frequently because it has a closed-form solution and is therefore relatively simple to implement. One solution is to choose a different error function that penalizes extreme points. Through our coursework we have investigated some such functions including Huber error and Hampel error. Closed form solutions do not exist for these solutions, so we often use a form of Newton iteration with these loss functions [8]).

We can introduce the Log-Cosh error defined as $\rho(r) = log(cosh(r))$ [8]), which is slightly smoother than Huber error (Huber error is not twice-differentiable everywhere, and Log-Cosh is). We can compare the corresponding linear models of Hampel and Log-Cosh to the model we get from least-squares with our $y = x^2 + 2x + 7$ data. (Note that Huber loss is defined as $\rho(r) = r^2$ near $r = 0$, and $log(cosh(r)) \approx r^2$ for $r \approx 0$, so the resulting linear models look very similar. As such, we instead compare Hampel and Log-Cosh).

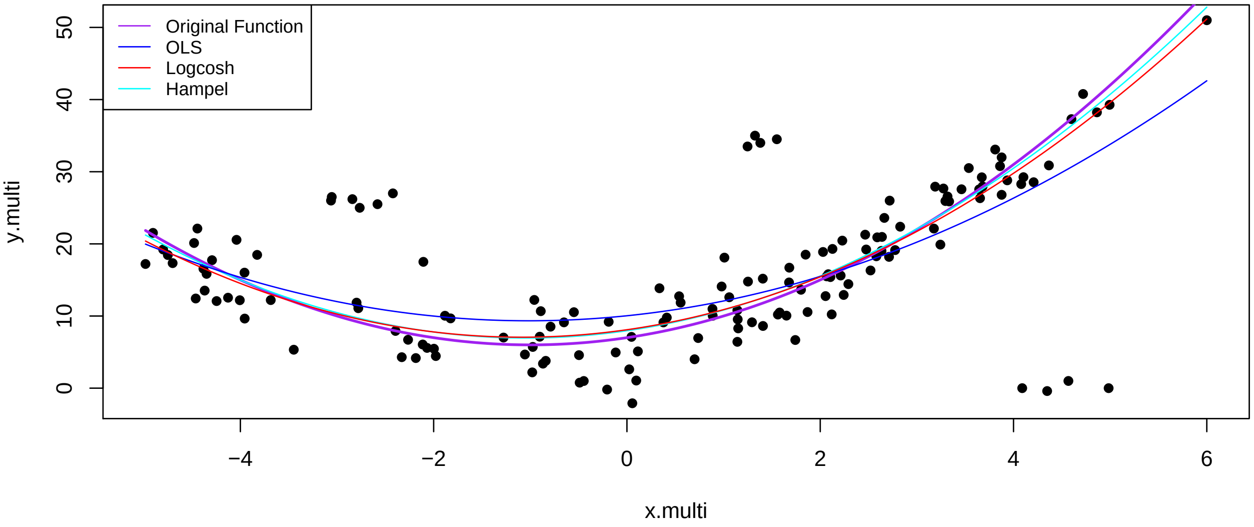


Figure 10: Robust Loss functions

As we can see, the robust linear models with Log-Cosh and Hampel error are quite comparable to the original function $y = x^2 + 2x + 7$, and resemble it much more closely than the LS fit. This therefore indicates that our robust linear models are substantially less influenced by outliers than the LS fit, as we would expect.

4. Conclusions and Future Work

Throughout this paper, we have investigated some methods of identifying and dealing with outliers. The first two methods for identifying outliers were Cook's distance and DFFITS. Both of these techniques have the benefit of being easy to implement and understand, but fall short when the data set contains many outliers. To remedy the multiple-outlier shortcoming, Sebert et al. came up with an algorithm that clusters similar data together. Though much better at quantifying multiple outliers, this algorithm is harder to conceptualize and implement.

We then briefly tackled the question: What do we do once we have identified outliers? There is no simple answer, but solutions could involve using a loss function that is robust against extreme points, or clearly marking outliers when drawing conclusions, and giving insights into why these outliers might exist.

Future work would involve investigating more methods for identifying outliers. This paper only investigated three of very many techniques. I would also look into more complex data sets. For the purposes of illustration, the data sets I made were sufficient. However, small, cherry-picked examples are not the best for getting an effective understanding of how the techniques work. Furthermore, I only tested on two-dimensional data. These techniques should scale relatively well to many-dimensional data sets. Since most real-world problems are many-dimensional, it would be useful to see how implementation works for these larger problems.

We briefly discussed the difference between an influential point and an outlier, but did not go into detail about how to differentiate the two. One algorithm for validating outliers that came up freqently in my research was to remove the identified outliers, and use a t-statistic to test the candidate outliers against the group of inliers [6]. I would have liked to implement this metric, but did not have the space to discuss it.

Due to the page-limit constraints, I was unable to go into significant detail about any of the techniques. In the future I would be interested in diving deeper, investigating the rigor (or lack thereof) that went into their construction.

The work I did could easily branch into other, more advanced problems in robust regression. I could spend more time looking into new cost functions, how to deal with heteroscedasticity, computational intensity, and possible alternatives to M-estimation. I also would have liked to introduce the most classic methods for identifying outliers (residual plots and quantile boxplots), and why they might not be sufficient, especially in multi-dimension frameworks.

Appendix

Data Collection

This report included three different data sets. The first data set was only used once because of its simplicity. It was generated by taking 30 points from a uniform distribution between $[-10, 10]$, and passing it through a function $y = x + 7$. To make the data look more realistic, each point was shifted by a random value pulled from a normal distribution. This data existed only to introduce the reader to the concept of outliers in regression, and show how significantly they effected an OLS fit. This data was generated as follows:

```r
set.seed(123456)
x = sort(runif(30,-10,10))
outliers = c(28,  29,  30)
y = x+7 + rnorm(30,sd=3)
y[outliers] = c(59 , 59.5 , 58.5)
```

The next data set was generated from the function $y = x^2 + 2x + 7$. As in the first data set, 150 points were generated, passed through the function, and shifted by values pulled from a normal distribution. Some points were then coerced into looking like outliers. This data set was used for the bulk of the paper, and the outlier points were specifically shifted so that some of them would barely make the cutoff for Cook's distance, and some would be barely below the cutoff. My intent was to make the two groups look similarly "outlying" to convince the reader that an eye test is not always sufficient for determining outliers. This data was generated as follows:

```r
set.seed(654321)
x.multi = sort(runif(150,-5,5))
y.multi = x.multi^2+ 2*x.multi+ 7 + rnorm(150,sd=3)
outliers = c(140,143,145,149,22,23,24,27,28,29,80,83,84,87)
y.multi[outliers] =
  c(0,-0.4,1,0,26,26.5,26.2,25,25.5,27,33.5,35,34,34.5)
x.multi[151] = 6
y.multi[151] = x.multi[151]^2 + 2*x.multi[151] + 7 - 4
```

For the final data set used in this paper, I took the first data set, and shifted the first 9 data points signficiantly upward, then shifted the next 5 data points to the right to create three distinct "clusters". The cluster data set served the purpose of outlining the efficacy of the sebert et al. algorithm as opposed to Cook's distance when dealing with multiple outliers. The data was generated as follows:

```r
x.clusters = x
y.clusters = y
y.clusters[1:9] = y.clusters[1:9] + 40
x.clusters[10:16] = x.clusters[10:16] + 5
```

Modelling Details

For each data set in sections 3.1-3.3, I used ordinary least squares models to fit the data. Since the purpose of this paper was not to investigate modeling, but instead the work that comes before and after modelling, the chosen models are not as important. The $x + 7$ data was fit with a simple linear model, and the $x^2 + 2x + 7$ data was fit with a quadratic. I could have instead chosen the model complexity with Cross Validation, but the resulting analysis would not have changed significantly. In section 3.4, the models were again fit with quadratics, but the Hampel and Log-Cosh models were fit using *rlm* in R instead of *lm*.

Theory

Section 3.1

- In his paper, Cook briefly discusses how to use his eponymous distance metric in detecting multiple outliers (which we have established the metric is poor at identifying). Cook introduces the independent variable hull (IVH), which is the smallest convex set that contains all data points. He then notes that outliers at the boundary of the IVH are harder to detect.
- Cook further states that since h_{ii} (the diagonal elements of the H matrix) are an increasing function of p (the length of β), it is more difficult to identify outliers the more complex a model gets.

Section 3.3

- Euclidean distance is defined as $d(x_1, x_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$.
- Mojena's stopping rule, mentioned in section 3.3, is a stopping metric for hierarchical grouping methods introduced by R. Mojena in 1977. He suggests a cutoff point when sorting data with hierarchical models to be approximately $\overline{h} + \alpha s_h$, where $\overline{h}$ is the average height of the tree clusters, s_h is the standard deviation of the cluster heights, and α is between $2.75 - 3.5$. In 1985, a more comprehensive study was done, and an α-value of 1.25 was found to be optimal [9].
- We standardize the residuals because the observations that are most variable will dominate the euclidean distance measure. For the residuals and predicted values to be comparable in variation, it is necessary that we standardize them both [1]
- Miller and Cooper in [9] state that the current stopping procedures are "ad-hoc", and mostly based on empirical evidence rather than rigorous mathematics. Despite this, they acknowledge that there is no evidence to suggest that Mojena's rule is any worse than the other stopping mechanisms. It was therefore chosen because of its simplicity.
- Sebert et al. concluded that the clustering algorithm works best with few outliers according to the cases they tested.
- This algorithm tends to produce swamped observations. That is, the algorithm has a tendency to call inliers "outliers" when they are slightly too far from the largest cluster [6].

Section 3.4

- We briefly discussed Hampel and Huber loss. Huber loss is defined as:

$$\rho(r) = \begin{cases} \frac{1}{2}r^2, & |r| \leq c \\ c(|r| - \frac{1}{2}c), & |r| > c \end{cases}$$

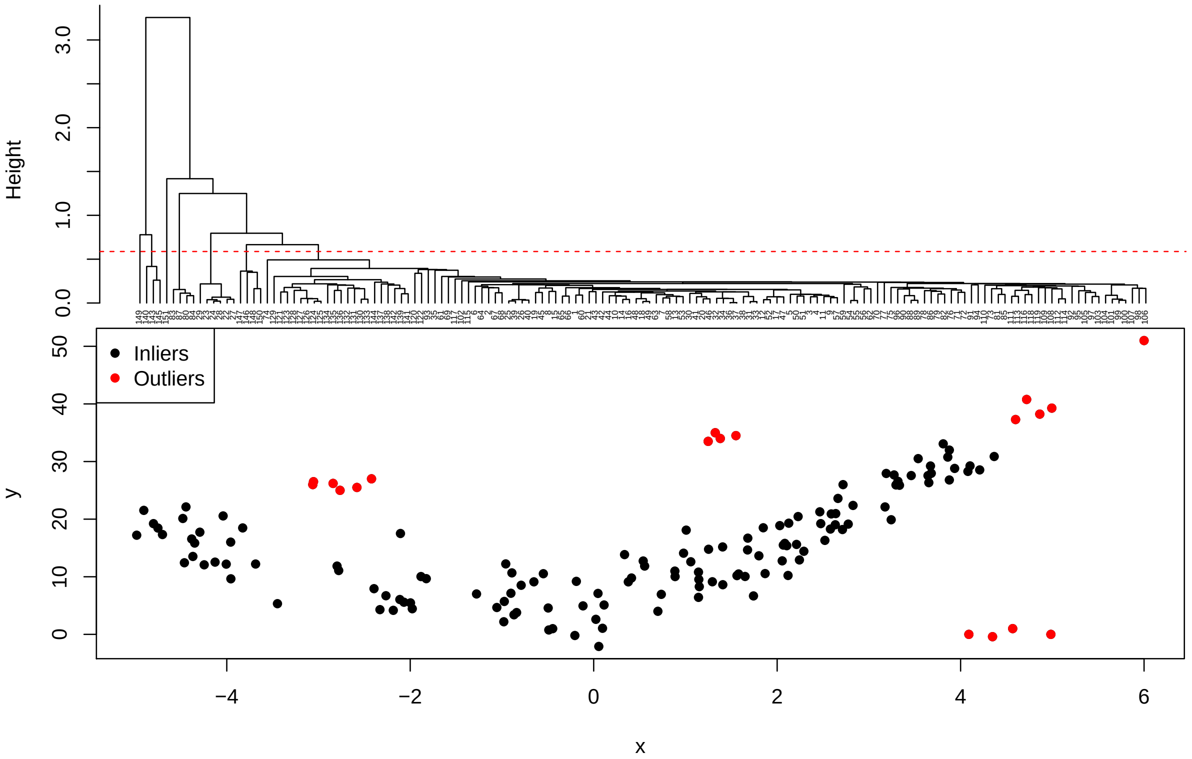
- Hampel loss is defined such that:

$$\begin{aligned} \psi(r) &= \rho'(r) \\ &= \begin{cases} r, & |r| \leq a \\ a \times \text{sign}(r), & a \leq |r| \leq b \\ a \times \frac{c-|r|}{c-b}\text{sign}, & b \leq |r| \leq c \\ 0, & |r| \geq c \end{cases} \end{aligned}$$

- The derivative of $\log(\cosh(r))$ is $tanh(r)$

Plots

- We can run the single-linkage clustering algorithm on the $x^2 + 2x + 7$ data to visualize how well it works on a dataset that doesn't follow a very well-defined clustering rule:



- This plot indicates that the algorithm is flawed when we try to use it on a data set that is not clearly clustered. Though the obvious outliers were chosen, the algorithm also selected the influential point at $x = 6$, as well as another group around $x = 5$ that was not identified by Cook's distance or DFFITS. To remedy the overestimation of outliers, we could use the t-statistic reference in the Conclusions and Future Work section.

References

[1] D. Sebert, D. Mongomery, D. Rollier (1998) A clustering algorithm for identifying multiple outliers in linear regression. *Computational Statistics and Data Analysis*, 27(4),461-484. doi:10.1016/s0167-9473(98)00021-8

[2] R. Dawson (2011) How Significant is a Boxplot Outlier?. *Journal of Statistics Education*, 19(2), doi: 10.1080/10691898.2011.11889610

[3] R. D. Cook (1979) Influential Observations in Linear Regression *Journal of the American Statistical Association.* 74(365), 169-174. doi:10.1080/01621459.1979.10481634

[4] G. D. Jayakumar, A. Sulthan (2014) Exact Distribution Of Cook's Distance And Identification Of Influential Observations. *Hacettepe Journal of Mathematics and Statistics*, 44(8), 1-1. doi: 10.15672/hjms.201487459

[5] A. H. Imon (2005) Identifying multiple influential observations in linear regression. *Journal of Applied Statistics*, 32(9), 929-946. doi:10.1080/02664760500163599

[6] S. Kim, W.J. Krzanowski (2007) Detecting multiple outliers in linear regression using a cluster method combined with graphical visualization. *Computational Statistic*, 22(1), 109-119. doi:10.1007/s00180-007-0026-3

[7] R. Mojena (1977) Hierarchical grouping methods and stopping rules: An evaluation *The Computer Journal*, 20(4), 359-363. doi:10.1093/comjnl/20.4.359

[8] P. W. Holland, R. E. Welsch (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9), 813-827. doi:10.1080/03610927708827533

[9] G. W. Milligan, M. C. Cooper (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2), 159-179. doi: 10.1007/bf02294245