FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Michal Jelínek

Least Weighted Squares: Simulation Studies

Department of Probability and Mathematical Statistics

Supervisor of the bachelor thesis: RNDr. Jan Kalina, Ph.D.
Study programme: General Mathematics

Prague 2025

I declare that I carried out this bachelor thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to gratefully thank my supervisor for all the consultations and time dedicated to me and my thesis. Next, the author of this thesis template (Mareš [1]) deserves my deep gratitude for saving me plenty of time with technical aspects of the work. I thank to developers and administrators of the faculty computational cluster Sněhurka as well, as the cluster enabled me to perform much more simulations in much shorter time.

Title: Least Weighted Squares: Simulation Studies

Author: Michal Jelínek

Department: Department of Probability and Mathematical Statistics

Supervisor: RNDr. Jan Kalina, Ph.D., Department of Probability and Mathematical Statistics

Abstract: This thesis investigates the least weighted squares estimator (LWS) as a robust alternative to more traditional regression methods, which are sometimes sensitive to outliers and heteroscedasticity. Concepts of data-dependent weights and various variance estimators, as the sandwich method or nonparametric bootstrap, are introduced. Simulation studies compare LWS with other robust estimators in terms of accuracy and precision and demonstrate its strength in some of the considered model designs.

Keywords: linear regression, simulations, robust statistics

Název práce: Nejmenší vážené čtverce: simulační studie

Autor: Michal Jelínek

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: RNDr. Jan Kalina, Ph.D., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Tato práce zkoumá odhad nejmenších vážených čtverců (LWS) jako robustní alternativu k tradičnějším regresním metodám, které bývají někdy citlivé na odlehlé hodnoty a heteroskedasticitu. Jsou představeny koncepty vah závislých na datech a různé odhady rozptylu, jako je sendvičová metoda nebo neparametrický bootstrap. Simulační studie porovnávají nejmenší vážené čtverce s jinými robustními odhady z hlediska přesnosti a preciznosti a ukazují jejich přednosti v některých z uvažovaných modelů.

Klíčová slova: lineární regrese, simulace, robustní statistika

Contents

Introduction

Since the beginning of the 19th century, linear regression has held an undisputable place among statistical tools. For many real-life applications, as well as theoretical studies, it has proven as very valuable for modeling assumed dependencies through the linear regression model. The robust linear regression and its versatility makes it a fundamental method in statistical analysis, offering insights and solutions across a broad spectrum of disciplines.

The traditional least squares estimator (LS), while powerful, is sensitive to outliers and heteroscedasticity. This sensitivity can lead to significant biases and inefficiencies in the estimation process. To address these issues, robust statistical techniques have been developed, offering more reliable estimates in the presence of non-homoscedastic data.

In this thesis, we explore several statistical methods, focusing particularly on robust estimators. The primary object of interest is the least weighted squares estimator (LWS). In the first chapter, we compare LWS to their better known colleague, weighted least squares (WLS), and emphasize the differences. Among several variations of LWS, we focus particularly on the setting with data-dependent weights. Then, some statistical methods for evaluating obtained estimates and their variablity are introduced, as well as practical implementation of chosen estimators.

In the second chapter, we explore the performance of LWS through various simulation studies. After discovering optimal values of some of its parameters, we compare LWS to better known robust estimators, such as WLS or MM. We show that none of these estimation procedures outshines the others in all the examined aspects. Moreover, in some model designs the results of LWS end up being the best and hence, we call for further studying and application of this strong statistical technique into practice.

1 Theory

The first goal of this thesis is to introduce the least weighted squares (LWS) estimator. For this purpose, we need to present the basics regarding the linear regression model and the least squares estimator at first.

1.1 Basic definitions

On our way towards the least weighted squares, we should clarify some underlying terms.

Definition 1 (Linear regression model)**.** *Let n and p be natural numbers, $n > p$. Let $\{(Y_i, X_{i1}, \dots, X_{ip})^\top,\ i \in \{1, \dots, n\}\}$ be a set of independent and identically distributed random vectors (we say* observations *or* data rows*) and $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^\top$ (we say* error terms*) a vector of independent random variables. Denote $\boldsymbol{Y} = (Y_1, \dots, Y_n)^\top$ (say* response *or* independent variable*), $\boldsymbol{X}_i = (X_{i1}, \dots, X_{ip})^\top$ (*regressors*,* independent *or* explanatory variable*), $i \in \{1, \dots, n\}$ and*

$$\mathbf{X} = \begin{pmatrix} \boldsymbol{X}_1^\top \\ \vdots \\ \boldsymbol{X}_n^\top \end{pmatrix}.$$

We say that $(\boldsymbol{Y}, \mathbf{X})$ follows a linear regression model (LRM)*, if*

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{1.1}$$

for some $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top \in \mathbb{R}^p$ and the following conditions hold for $i \in \{1, \dots, n\}$:

$$\mathsf{E}[\varepsilon_i | \boldsymbol{X}_i] = 0, \tag{1.2}$$

$$\mathsf{var}[\varepsilon_i | \boldsymbol{X}_i] = \sigma^2(\boldsymbol{X}_i), \tag{1.3}$$

where $\sigma^2(\cdot)$ is a function on $\mathbb{R}^p$ such that $\sigma^2(\boldsymbol{X}_i)$ is an almost surely finite, not constant zero and non-negative random variable.

Reader may notice that we consider the regressors in the form of random variables. The second possible approach with fixed regressors, applied e.g. by Víšek [2], will not be examined throughout the paper.

In this thesis we consider the intercept to be always present among the explanatory variables. That means we lay down $\boldsymbol{X}_1 = (X_{1,1}, \dots, X_{1,p})^\top = (1, \dots, 1)^\top$. Even though the opposite variant of not including the intercept is also possible, it requires a special setting to make practical sense, which is not the case in our models.

The assumption (1.2) is quite reasonable in the common sense. It only requires the model not to be shifted in such a way, that the expected error given $\boldsymbol{X}_i$ is positive or negative. We want it to remain zero.

The formulation of (1.3) in Definition 1 refers to the general definition of a linear regression model, i.e. either for homoscedastic and for heteroscedastic data. However, we could obtain the narrower definition for homoscedastic data only by placing $\sigma^2(\cdot) \equiv \sigma_e^2$ for some constant $\sigma_e^2 \in (0, \infty)$.

Heteroscedasticity, as an assumption in regression analysis, necessitates the use of diagnostic tools to detect and assess its presence; in the literature, attention has also been paid to testing heteroscedasticity for robust regression, as in Kalina [3].

Remark 1. *The assumption* (1.3) *could be equivalently written as*

$$\mathsf{var}[Y_i|\boldsymbol{X}_i] = \sigma^2(\boldsymbol{X}_i)$$

for $\sigma^2(\boldsymbol{X}_i)$ almost surely finite non-negative random variable, such that $\sigma^2(\boldsymbol{X}_i) \not\equiv 0$ (our intention of this last requirement is to avoid degenerated cases). For homoscedastic data,

$$\mathsf{var}[Y_i|\boldsymbol{X}_i] = \sigma_e^2 \in (0, \infty).$$

Naturally, having defined the linear regression model, a question occurs, how to estimate the model parameters. The estimator which very often comes to mind at first in the terms of linear regression is the least squares estimator.

Definition 2 (Least squares estimator)**.** *Under the conditions of Definition 1 the* least squares estimator (LS) *is defined as*

$$\hat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{Y}. \tag{1.4}$$

There are plenty of theoretical results being done regarding the least squares estimator. A big part of them, however, requires homoscedastic data to be true, i.e. the condition $\sigma^2(\cdot) \equiv \sigma_e^2 \in (0, \infty)$. It is known (as e.g. Víšek in [4] points out) that even under heteroscedasticity, the least squares estimator is unbiased, consistent and asymptotically normal. Many users of the linear regression might then think that the heteroscedasticity is not a serious concern. However, when it comes to building a model and testing the significance of individual parameters, without homoscedasticity the regression error estimate $\hat{\sigma}_e^2$ does not make any sense.

Hence, the test statistics with $\hat{\sigma}_e$ in the denominator (see e.g. Anděl [5]) has to be modified to suit the heteroscedasticity. Otherwise, the resulting p-values are wrong and we might end up with a model with many insignificant covariates. Then we are usually not able to explain the effects of dependent variables on the response correctly.

In some cases, however, the solution for heteroscedasticity can be quite simple.

Example. If there are m groups counting $n_1, \ldots, n_m$ observations, but only the group means are available, assuming $n_i \neq n_j$ for some $i, j \in \{1, \ldots, m\}$ (to ensure that heteroscedastic design), the heteroscedasticity is present in the data. In mathematical terms we assume $Y_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Z_{ij}$ where for $j \in \{1, \ldots, n_i\}$ it holds that $\mathsf{E}[Z_{ij}|\boldsymbol{X}_i] = 0$ and

$$\mathsf{var}[Z_{ij}|\boldsymbol{X}_i] = \sigma_e^2 \in (0, \infty).$$

In other words, data $\{(Z_{ij}, \boldsymbol{X}_i)^\top, j \in \{1, \ldots, n_i\}, i \in \{1, \ldots, m\}\}$ are homoscedastic, i.e. they suit well for a homoscedastic LRM. Then,

$$\mathsf{var}[Y_i|\boldsymbol{X}_i] = \frac{1}{n_i} \sigma_e^2 \tag{1.5}$$

is not constant, as we would expect it to be for homoscedastic data according Remark 1 (it differs for some i, j in $\{1, \ldots, m\}$ for which $n_i \neq n_j$).

In this setting, however, we make use of the weighted least squares estimator in the Definition 5 below. Nevertheless, firstly we need to define some terminology.

Definition 3 (Fitted values, residuals, sorted residuals, ranks)**.** *Under the conditions of the Definition 1 and having an arbitrary estimator $\hat{\boldsymbol{\beta}}$, the ith* fitted value *$\hat{Y}_i$ is defined as $\hat{Y}_i = \boldsymbol{X}_i^\top \hat{\boldsymbol{\beta}}$. Next, as the ith* residual *$u_i(\hat{\boldsymbol{\beta}})$ is considered the quantity $u_i(\hat{\boldsymbol{\beta}}) = Y_i - \boldsymbol{X}_i^\top \hat{\boldsymbol{\beta}}$. Denote $\boldsymbol{u}(\hat{\boldsymbol{\beta}}) = (u_1(\hat{\boldsymbol{\beta}}), \dots, u_n(\hat{\boldsymbol{\beta}}))$ the vector of residuals. The ith* sorted residual *$u_{(i)}(\hat{\boldsymbol{\beta}})$ is the ith order statistics of $\boldsymbol{u}(\hat{\boldsymbol{\beta}})$. The* rank *of ith residual r_i is the index from $\{1, \dots, n\}$ such that $u_i(\hat{\boldsymbol{\beta}}) = u_{(r_i)}(\hat{\boldsymbol{\beta}})$.*

In situations when $u_i(\hat{\boldsymbol{\beta}}) = u_j(\hat{\boldsymbol{\beta}})$ for some $i \neq j$, we sort these two (or more) residuals at random. Even though the theoretical probability of this event is zero, in praxis these events occur e.g. by means of rounding.

Remark 2. *With the residuals defined, the least squares estimator could be equivalently written as*

$$\hat{\boldsymbol{\beta}}_{LS} = \underset{\hat{\beta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} (u_i(\hat{\boldsymbol{\beta}}))^2. \tag{1.6}$$

In other words, by performing the least squares we are looking for a solution $\hat{\boldsymbol{\beta}}_{LS}$ of $\boldsymbol{Y} = \mathbf{X}\hat{\boldsymbol{\beta}}$ in $\hat{\boldsymbol{\beta}}$ which minimizes the sum of squares of all the residuals. We will refer to this sum as to the *mean square error* (if divided by n) or *residual sum of squares.*

Definition 4. Mean square error (MSE) *is defined by an expression*

$$\mathsf{MSE}(\hat{\boldsymbol{\beta}}) = \frac{1}{n} \sum_{i=1}^{n} (u_i(\hat{\boldsymbol{\beta}}))^2.$$

This is the basic characteristics of how (un)well the fitted values of the model correspond to the observed response $\boldsymbol{Y}$. However, it does not suit to the methods of robust statistics quite well, as is described below in the section 1.6.2. Now, however, we proceed with the definition of the weighted least squares estimator.

1.2 Weighted least squares estimator

Throughout the thesis, the author is trying his best both in terms of clear defining and also distinguishing between the weighted least squares and the least weighted squares, whose names are only the permutations of each other.

Definition 5 (Weighted least squares estimator)**.** *Suppose we have a vector of non-negative weights $\boldsymbol{w} = (w_1, \dots, w_n)^\top$ corresponding to n observations. Then the* weighted least squares estimator (WLS) *is defined as*

$$\hat{\boldsymbol{\beta}}_{WLS} = \underset{\hat{\beta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} w_i (u_i(\hat{\boldsymbol{\beta}}))^2. \tag{1.7}$$

Remark 3. *The weighted least squares estimator could be equivalently defined as*

$$\hat{\boldsymbol{\beta}}_{WLS} = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \boldsymbol{Y}, \tag{1.8}$$

where

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$$

is a diagonal matrix with weights $\boldsymbol{w}$ on its diagonal.

Example (continuation). Returning to the example from the previous section 1.1, we use the weighted least squares estimator with weights

$$\mathbf{W} = \begin{pmatrix} n_1 & 0 & \cdots & 0 \\ 0 & n_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n_m \end{pmatrix}$$

to solve the problem. The expression (1.5) could be then rewritten as

$$\mathsf{var}[\boldsymbol{Y}|\mathbf{X}] = \sigma_e^2 \cdot \mathbf{W}^{-1}.$$

Theoretically, the weighted least squares estimator (as in (1.8)) leads back to the properties of the ordinary least squares. To see this, we consider a transformation

$$\boldsymbol{Y}' = \mathbf{W}^{1/2}\boldsymbol{Y}, \quad \mathbf{X}' = \mathbf{W}^{1/2}\mathbf{X} \quad \text{and} \quad \boldsymbol{\varepsilon}' = \mathbf{W}^{1/2}\boldsymbol{\varepsilon}.$$

Then it holds that

$$\mathsf{E}[\boldsymbol{\varepsilon}'|\mathbf{X}'] = \mathbf{0} \quad \text{and} \quad \mathsf{var}[\boldsymbol{Y}'|\mathbf{X}'] = \sigma_e^2 \cdot \mathbf{W}^{1/2}\mathbf{W}^{-1}\mathbf{W}^{1/2} = \sigma_e^2.$$

Hence, the requirements for the homoscedastic LRM

$$\boldsymbol{Y}' = \mathbf{X}'\boldsymbol{\beta}' + \boldsymbol{\varepsilon}'$$

are fulfilled and the LS estimator $\hat{\boldsymbol{\beta}}'_{LS}$ can be expressed as

$$\hat{\boldsymbol{\beta}}'_{LS} = \left((\mathbf{X}')^\top \mathbf{X}'\right)^{-1} (\mathbf{X}')^\top \boldsymbol{Y}' = \left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}\boldsymbol{Y},$$

which is exactly how the estimator $\hat{\boldsymbol{\beta}}_{WLS}$ looks like, considering the Remark 3.

The weighted least squares will also be useful in due course when constructing an estimator based on dependent weights. We will return to the WLS in the end of section 1.5.

1.3 Least weighted squares estimator

Returning to the least squares, another issue with them lies in their non-robustness. That means that even a single observation with a biased dependent variable (*vertical outlier*), or both dependent and independent variable (*leverage point*), may shift the model far from what we would call a sensible estimate of parameters. Such an *outlier* can be caused in numerous ways - for example, by a (gross) error in measurement or just by presence of a relatively exceptional observation in the dataset.

In other words, non-robust estimators are sensitive to outliers.

Therefore, in robust statistics we need to address the task of estimating the coefficients in a linear regression model differently. One would suggest using the WLS from the previous section 1.2, but an unsolvable problem with a straightforward application occurs when no weights are known beforehand. More generally, it is needed to evaluate (at least roughly) at first, which observations fit well to the assumed model and oppositely, which are (probably) the outliers. This is how we get at two-step regression estimate procedures.

Now we aim for defining the least weighted squares estimator, which is an estimator based on weighting the residuals according to their ranks. The following definition of a weight function is helpful.

1.3.1 Weight function

Definition 6 (Weight function)**.** *By a* weight function *we mean non-negative function on* $[0, 1]$*, which is non-increasing.*

We explore the definition of weight function a little deeper. By the first condition on a weight function $w(\cdot)$ we want to ensure it assigns only non-negative weights. Next, having n observations and hence n residuals, we divide the interval $[0, 1]$ into n uniform subintervals and map the residuals to the center points of the subintervals. Therefore, we need to assign the weights to values $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$, hence the need to define a weight function on $[0, 1]$.

Based on this approach of n equidistant points, note that the values of the weight function on $[0, 1] \cap \left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}^C$ are often not important for our purposes. Hence, when referring to the weight function in the next, only the values on $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$ are usually considered.

Often, we work with sorted residuals. As usual in the robust statistics, we want to assign the largest weights to observations with the smallest residuals, i.e. to observations with the smallest ranks of residuals. Vice versa, observations with the highest ranks of residuals (e.g. outliers) are being assigned with the smallest weights - with these close to zero or even zero. From there is follows the requirement on the weight function of being non-increasing.

Weights vs. a weight function

This is the first suitable place to emphasize the difference between a *weight function* presented here and *weights* of individual observations covered by the previous section. For applying the latter, we must have some preliminary information about the individual rows of data. This might be obtained from the nature of collection of the data themselves (as was the case in the example, where only the group means were available) or by means of fitting a preliminary model. In both cases, we cannot control for the exact values of these weights very much (except of own designing the data collection beforehand - by choosing the group sizes before the data themselves are sampled).

In opposition, the term *weight function* so far refers to a fixed (non-ascending) function, whose shape and values are known independently of the data.

However, often we gain weights out of a weight function design: we evaluate the weight function on a given set of n points (as mentioned above already, in this thesis for this purpose we use almost exclusively the set $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$) and assign these values to n observations by means of a permutation. The searching for this particular permutation is directed by minimizing of some sort of residual sum of squares in theory and by the corresponding iteration algorithm in practice.

Vice versa, for gaining values of some weight function $w(\cdot)$ on the set $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$, we only have to sort the elements of $\boldsymbol{w}$ in the non-increasing order - if $\boldsymbol{w}$ is a vector of weights corresponding to a model based on n observations.

Fixed weight functions designs

Now we introduce some fixed weight functions designs. They stand as the base for the following, most commonly known version of the least weighted squares estimator. Hence, we denote them with acronym LWS. In further we explore, which of them yield the most promising results. Regarding the weight functions from LWS-1 to LWS-4, we were directly inspired by J. Kalina's article [6] and we assume them without any modification. The weight function LWS-5 is even further simplification of the weight function LWS-3. It is closely related to the least trimmed squares, for definition of which see the section 1.4 below.

LWS-1 **Linear weights**

$$w(t) = 1 - t, \quad t \in [0, 1].$$

LWS-2 **Weights generated by the logistic function**

$$w(t) = \frac{1 + \exp\left\{-\frac{s}{2}\right\}}{1 + \exp\left\{s\left(t - \frac{1}{2}\right)\right\}}, \quad t \in [0, 1],$$

for a given constant $s > 0$, while in further we consider only $s = 10$.

LWS-3 **Trimmed linear weights generated for $\tau \in [1/2, 1)$ by**

$$w(t) = \left(1 - \frac{t}{\tau}\right) \cdot \mathbb{I}[t < \tau], \quad t \in [0, 1],$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. τ is related to the amount of observations that are trimmed out.

LWS-4 **Weights generated by the so-called error function**

$$w(t) = 1 - \frac{2}{\sqrt{\pi}} \int_0^t \exp\{-x^2\}\, dx. \tag{1.9}$$

LWS-5 **Trimmed constant weights generated for $\tau \in [1/2, 1)$ by**

$$w(t) = \mathbb{I}[t < \tau], \quad t \in [0, 1]. \tag{1.10}$$

In the literature, sometimes the designs of weight functions LWS-3 and LWS-5 are combined with the data-adaptive trimming constant τ. For now, we keep these weight functions independent of the data by choosing fixed $\tau = 0.75$.

The LWS estimator combined with all of these data-independent weight functions (i.e., 5 versions of LWS) are to be explored numerically in the second chapter. But firstly, we define the LWS and see how the term *weight function* is applied in there.

1.3.2 Least weighted squares definition

Definition 7 (Least weighted squares estimator)**.** *Under the conditions of the Definition 1, keeping the notation from the Definition 3 and for a given weight function $w(\cdot)$, the* least weighted squares estimator (LWS) *is defined as*

$$\hat{\boldsymbol{\beta}}_{LWS} = \underset{\hat{\boldsymbol{\beta}} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} w\left(\frac{i - 1/2}{n}\right) (u_{(i)}(\hat{\boldsymbol{\beta}}))^2.$$

For a given explicitly defined weight function, e.g. for these five mentioned above, LWS is a one-step procedure. In further, we introduce an idea from Čížek [7], who devised an algorithm to obtain a particular type of weights that are data-dependent. Firstly, however, we define other estimators, which will perform as the opponents mainly to the least weighted squares. This order was chosen due to their close relationship with LWS and also weight functions defined right above.

1.4 Other robust estimators

In the simulation section we compare the properties of LWS estimation with some other common robust estimators in linear regression. Specifically, we explore estimators' accuracy, precision and efficiency. For more details regarding evaluating an estimator, see the section 1.6.

1.4.1 Least trimmed squares and least median squares

We have already defined the least squares estimator in the Definition 2. However, as we have referred above, for a dataset contaminated with outliers, we expect this estimator to turn out as insufficient.

We expect that the least trimmed squares estimator will behave much better on such data. It is one of the simplest robust estimators. Its weights (again, a vector of constants corresponding to rows of data) are obtained just by applying the weight function LWS-5 from 1.3 with a given trimming constant τ, according the way that was described in the subsection 1.3.1. Hence, it consist of zeros and ones only, while for $\tau \in [1/2, 1)$ there are exactly $\lfloor \tau \cdot n \rfloor$ observations with assigned weight 1 and $n - \lfloor \tau \cdot n \rfloor$ observations are trimmed out (i.e., their assigned weight is zero).

Definition 8 (Least trimmed squares estimator)**.** *Under the conditions of the Definition 1 and keeping the notation from the Definition 3, for* $\tau \in [1/2, 1)$ *the* least trimmed squares estimator (LTS) *is defined as*

$$\hat{\boldsymbol{\beta}}_{LTS} = \underset{\hat{\boldsymbol{\beta}} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{\lfloor \tau \cdot n \rfloor} (u_{(i)}(\hat{\boldsymbol{\beta}}))^2.$$

Obviously, the smaller the τ, the more robust the estimator is. However, we set a lower border $\tau = 1/2$. The reasoning behind is that for $\tau < 1/2$ the estimator would classify more that $1/2$ of observations as outliers, which does not make any good sense. In further, while referring to the LTS, we set $\tau = \lfloor \frac{n}{2} \rfloor + \lfloor \frac{p+1}{2} \rfloor$, as proposed by e.g. Rousseeuw and Leroy [8].

Usage of this data-adaptive trimming constant τ distinguishes LTS from the above-mentioned LWS combined with weight function LWS-5 (which could be formally classified as least trimmed squares as well; nevertheless, for clarity especially in the simulations section, we stick to the distinguishing just mentioned).

The one inconvenience with LTS (but as well as with all the estimators in this section) is the inability to express them explicitly, as was the case with the LS in (1.4) and the WLS in (1.8). Hence, we need to compute them by means of iterations. The algorithms used are described in the section 1.7.

Another highly robust, but yet still quite simple estimator we introduce is the least median squares estimator. This estimator is similar to LS in a way, that instead of minimizing the sum of squares (see (1.6)), which is equivalent to minimizing the mean of squares, here we minimize the median of squares. This exchange of the mean for the median is quite common when it comes down to robustification of estimators.

Definition 9 (Least median squares estimator)**.** *Under the conditions of the Definition 1 and keeping the notation from the Definition 3, the* least median squares estimator (LMS) *is defined as*

$$\hat{\boldsymbol{\beta}}_{LMS} = \arg\min_{\hat{\beta}\in\mathbb{R}^p} \mathsf{med}\left((u_i(\hat{\boldsymbol{\beta}}))^2, i \in \{1,\dots,n\}\right).$$

Still highly robust, but a little more complicated class of estimators follows.

1.4.2 S-estimators

S-estimators are based on minimizing values of some loss function $\rho(\cdot)$. The letter S stands for *scale*.

Usually, the following conditions on the loss function $\rho : \mathbb{R} \to \mathbb{R}$ are required (for details see Hampel et al. [9]):

- $\rho(\cdot)$ is symmetric and continuously differentiable,
- $\rho(0) = 0$,
- for some $c > 0$ $\rho(\cdot)$ is strictly increasing on $[c, \infty)$.

However, we do not analyse S-estimators very thoroughly in this thesis. Therefore, these requirements serve only as a hint for the reader to imagine what hides itself behind the term *loss function*. As we use optimized loss functions of the built-in methods in R, we do not need to explain them here any further.

Definition 10. *Denote $K = \int \rho(x)d\Phi(x)$, where $\Phi(x)$ stands for the distribution function of $N(0,1)$ - normal distribution with zero mean and unit variance.*

So, K is the expected value of $\rho(x)$ under the standard normal distribution.

We define the scale statistics $s = s(u_1(\hat{\boldsymbol{\beta}}),\dots,u_n(\hat{\boldsymbol{\beta}}))$ *as the solution of*

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{u_i(\hat{\boldsymbol{\beta}})}{s}\right) = K. \tag{1.11}$$

Assuming some solution to (1.11) *exists, the* S-estimator *is then defined as the argument of the smallest solution of* (1.11)*, i.e.*

$$\hat{\boldsymbol{\beta}}_S = \arg\min_{\hat{\beta}\in\mathbb{R}^p} s(u_1(\hat{\boldsymbol{\beta}}),\dots,u_n(\hat{\boldsymbol{\beta}})).$$

To get some insight into S-estimators, it might be useful to assume that model residuals $u_i(\boldsymbol{\beta})$ come from the normal distribution with a fixed variance $\sigma^2 \in (0,\infty), i \in \{1,\dots,n\}$. Then, as the empirical mean is a consistent estimator of the expected value (details e.g. in Anděl [5]), for n large enough there is a unique solution to (1.11), which takes the form $s = s(u_1(\hat{\boldsymbol{\beta}}), \ldots, u_n(\hat{\boldsymbol{\beta}})) \approx \sigma$, as the distribution of the scaled residuals $\frac{u_i(\hat{\beta})}{s}$ gets closer to $N(0,1)$ as $n \to \infty$ (this is a shorter description of the convergence to the random variable with $N(0,1)$ distribution and it is commonly used throughout the rest of the thesis). Vaguely saying, this idea is then somehow generalized among other distributions of residuals as well.

The high level of robustness is a valuable property, for which S-estimators are often employed. Hence, as well as the least trimmed squares or the least median squares, they are legitimately regarded as a very suitable initial estimators (not only) for the two-step least weighted squares algorithm. See the section 1.5.

1.4.3 MM-estimators

As another opponent we pick the MM-estimator, which is widely recognized as an efficient and reliable tool in robust regression and it still deserves high attention (see e.g. Fishbone and Mili [10]). As their definition is rather complicated and outside of the scope of this thesis, we just refer to their basic principle they are based on. For details see Yohai [11].

In the first step, initial S-estimator is used to compute a scale estimate $\hat{s}$. Provided some assumptions of technical character, the MM-estimator $\hat{\boldsymbol{\beta}}_{MM}$ is then defined as the solution of

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{u_i(\hat{\boldsymbol{\beta}})}{\hat{s}}\right) Y_i = 0$$

in $\hat{\boldsymbol{\beta}}$.

In comparison with S-estimators, MM-estimators possess esp. a higher efficiency, i.e., they usually need less observations to estimate the model parameters more reliably (as e.g. Yohai in [11] points out). As a matter of categorization, MM-estimators belong to the category of two-step estimators, i.e. the first step consists of an initial estimation and only then the final MM-estimator of coefficients results from the second step. Now we stay in this field of two-step estimators for a little while, as we return to the LWS estimator and its further variants.

1.5 Data-dependent weights

Until this point we were accompanied only by weight functions that were fixed, or at least, that did not depend on data more than by means of data-dependent trimming constant. Now we explore a realm of data-adaptive weight functions, i.e. of these, whose values depends on the data.

The two-step approach studied e.g. by Čížek [7] goes in the following way: in the first step we use some highly robust initial estimator to estimate $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}^0$ and get the residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0)$. In this thesis we consider four different initial estimators - LTS, LMS, S and LS estimators. We test their reliability on a given simulation model in the Chapter 2.

Then, weights $\boldsymbol{w}(\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0))$ that depend on $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0)$, are computed. As in 1.3.1, by sorting elements of $\boldsymbol{w}(\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0))$ we get the values of the corresponding weight function on $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$. These obtained values of the weight function are then being treated as fixed and for example the least weighted squares estimator from the Definition 7 is performed.

Now we take a closer look at how such weights might be computed. We aim our attention to the Čížek's design - the purpose of these weights is to bring generally distributed residuals closer to normality.

Firstly, we know that in a linear regression model with error terms following conditionally $N(0,1)$ distribution (i.e., $\varepsilon_i|\boldsymbol{X}_i \sim N(0,1), i \in \{1,\ldots,n\}$; in further we omit the word *conditionally*, as in the linear regression setting it is usual to be interested in the conditional in opposition to the unconditional distribution), the ordinary least squares $\hat{\boldsymbol{\beta}}_{LS}$ is the best unbiased (i.e., $\mathsf{E}[\hat{\boldsymbol{\beta}}_{LS}] = \boldsymbol{\beta}$) linear estimator of $\boldsymbol{\beta}$ (Gauss and Dieterich [12]). The adjective *best* refers to the lowest mean square error among all linear unbiased estimators of $\boldsymbol{\beta}$. Next, if the residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}})$ are normally distributed (in LRM, that occurs if and only if the error terms $\boldsymbol{\varepsilon}$ are normally distributed), then for every $i \in \{1,\ldots,n\}$, $(u_i(\hat{\boldsymbol{\beta}}))^2$ follows the χ^2_1 distribution, i.e. chi-square distribution with 1 degree of freedom.

The idea behind Čížek's approach lies in transforming the squares of generally distributed residuals to χ^2_1-distributed random variables by means of suitable weights. By this step we should "get closer" to the situation with normally distributed residuals, which is optimal for applying the least squares. The data-dependent weights in the following definition serve exactly to this "normalization" purpose. Let $\hat{\boldsymbol{\beta}}_n$ refer to the estimator of $\boldsymbol{\beta}$ calculated from n observations.

Definition 11 (Data-dependent weights). *Denoting $(G^0_n)^{-1}$ the empirical quantile function of the model squared residuals $(u_i(\hat{\boldsymbol{\beta}}^0_n))^2$, $i \in \{1,\ldots,n\}$ and F^{-1}_χ the quantile function of χ^2_1 distribution, weights defined as*

$$\hat{w}_n(t) = \frac{F^{-1}_\chi(\max\{t, a_n\})}{(G^0_n)^{-1}(\max\{t, a_n\})}, \tag{1.12}$$

where $t \in \left\{\frac{i-1/2}{n}, i \in \{1,\ldots,n\}\right\}$ and $a_n = \min\{m/n : u^2_{(m)}(\hat{\boldsymbol{\beta}}^0_n) > 0,\, m \in \mathbb{N}\}$, will be in further called data-dependent*.*

The technical formulation $(\max\{t, a_n\})$ instead of just (t) is included to avoid dividing by zero. However, there are some concerns appearing when it comes down to the algorithm implementation: due to the only finite computer accuracy and the inevitable rounding, the sharp inequality in the definition of a_n obviously cannot be proceeded to software in this exact form. In praxis we conduct the following: we sort the squares of residuals and all of them smaller than some constant $\eta > 0$ are replaced by η.

The question is, however, how small this η should be. On one hand, we do not want to weaken the main idea of the weights too much by setting η too high. On the other hand, the case of choosing too small η might lead to a division by too small number in the denominator of (1.12). This could yield very high values of some of the weights, destabilizing the overall procedure.

The final choice $\eta = 10^{-9}$ is discussed more in detail in the appendix A.2.

Remark 4. *Under several assumptions and suitable regularity conditions (see Čížek [7, sec. 5]) $\hat{w}_n(t)$ converge to $w(t) = F^{-1}_\chi(t)/(G^0_\beta)^{-1}(t)$ in probability as $n \to \infty$, where $(G^0_\beta)^{-1}(t)$ stands for the quantile function of the squared model residuals under the true but unknown parameter $\boldsymbol{\beta}$.*

To put it more simply, in the literature the idea above is justified mathematically as well.

1.5.1 Data-dependent least weighted squares estimator

Čížek uses these data-dependent weights in the setting of the least weighted squares estimator.

Definition 12 (Data-dependent least weighted squares estimator)**.** *For a weight function $w(\cdot)$ obtained by sorting the data-dependent weights from the Definition 11, the* data-dependent least weighted squares (dd-LWS) *estimator is defined as*

$$\hat{\boldsymbol{\beta}}_{dd-LWS} = \underset{\hat{\beta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} w\left(\frac{i-1/2}{n}\right)(u_{(i)}(\hat{\boldsymbol{\beta}}))^2.$$

Here, the first step lies in using a (preferably) highly robust estimator to obtain the weights. Then, after sorting the weights, standard LWS routine is accomplished (for how to approach the theoretical formula from the Definition 7 numerically, see the section 1.7 below).

Nevertheless, LWS is not the only estimator with which we connect the data-dependent weights from the Definition 11.

1.5.2 Data-dependent weighted least squares estimator

It makes sense to handle with the weights obtained from the Definition 11 as with weights for the WLS estimator as well. It means not to sort them and not to iterate in this case.

Definition 13 (Data-dependent weighted least squares estimator)**.** *By* data-dependent weighted least squares estimator (dd-WLS) *we mean the weighted least squares estimator from the Definition 5 with the data-dependent weights obtained as described in the Definition 11.*

In Čížek [7], this approach seems to be missing. Čížek introduced a similar *robust and efficient weighted least squares estimator* (REWLS), which is based on 0-1 weights only with a data-dependent trimming constant. Hence, it is closely related to the least trimmed squares, but from the LTS from the Definition 8 it differs by using slightly more complicated form of a trimming constant τ. Details are provided in his article [7].

When Čížek refers to the data-dependent weights from the Definition 11, he does so only in connection with the LWS, not with the WLS. At the same time, author of this thesis holds the combination of data-dependent weights with WLS quite intuitive, as by permuting the weights we lose some of the benefits of that one permutation suited exactly for the given data. Simulations challenging this idea numerically are presented in the second chapter.

1.6 Robust metrics of an estimator

After performing simulations we obtain estimates of $\boldsymbol{\beta}$ and the corresponding residuals. In this section we define some metrics that enable us to evaluate and to quantify the quality of obtained estimates.

In the beginning we mention three terms related to how well the given estimator $\hat{\boldsymbol{\beta}}_n$ based on n observations estimates the coefficients $\boldsymbol{\beta}$ themselves. We are interested in the *accuracy* of the estimator, i.e. whether its expected value of coefficients given the data is in some sense close to the true model parameters. The *precision* is closely related to the variance of the estimator - the lower the variance, the more precise the estimator is. Finally, *efficient* estimator is the one that offers relatively exact estimation of the model parameters provided only smaller amount of data.

1.6.1 Mean square error of coefficients

Having K datasets and hence K realizations of an estimator $(\hat{\boldsymbol{\beta}}_n^1, \ldots, \hat{\boldsymbol{\beta}}_n^K)$, we may study the discrepancies between the estimated and the true coefficients for a given n, hence exploring the estimators' accuracy and precision at the same time. To author's knowledge, the name and the abbreviation of the following metric are not much unified across the literature. Hence, we present our own labeling of that quantity.

Definition 14 (Mean square error of coefficients). *Given $n \in \mathbb{N}$ and K realizations of an estimator $(\hat{\boldsymbol{\beta}}_n^1, \ldots, \hat{\boldsymbol{\beta}}_n^K)$, we define* mean square error of coefficients (MSEoC) *as*

$$\mathsf{MSEoC}(\hat{\boldsymbol{\beta}}_n) = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{p} (\hat{\beta}_{n,j}^i - \beta_j)^2.$$

Obviously, we may use the MSEoC only when coefficients $\boldsymbol{\beta}$ are known beforehand. However, as we use only simulated data through the rest of the thesis, there is no problem in adding this metric to our standard evaluating tools.

Having two or more different estimators and an increasing sequence of natural numbers $(n_1, \ldots, n_m)$, we may compare estimators' mean square errors of coefficients for increasing m. This approach offers us good options to compare efficiency of two or more estimators, even though we do not define the term *efficiency* rigorously. The efficient estimator will show lower values of MSEoC than the inefficient estimator.

Next, we introduce some measures of the regression errors, i.e. how well or badly the fitted values correspond to the data. Naturally, these metrics are based on model residuals.

1.6.2 Residual sums of squares

The most basic sum of squares of residuals is the mean square error, which has been already covered in the Definition 4.

Obviously, the optimal estimators in minimizing the MSE comes from the non-robust family. The least squares are a good example. The other robust estimators do not achieve such a relatively low value of MSE - their basic principle is trimming away some observations, or at least assigning them smaller weights compared to other observations. The squared residuals of these trimmed observations are usually by far the leading summands of the resulting MSE sum, as MSE includes all the squares of observations' residuals with the same weight.

In all of our simulations later, some form of heteroscedasticity is present (e.g. by creating outliers). Next, our main focus is aimed at robust estimators. In these situations, MSE is not an optimal metric. As a consequence of that, we consider another three robustly-adjusted residual sum of squares presented right below.

Trimmed mean square error

Definition 15. *Let* $\alpha \in [1/2, 1)$ *be a given constant. Then the* trimmed mean square error (TMSE) *is calculated by*

$$\mathsf{TMSE}(\hat{\boldsymbol{\beta}}_n) = \frac{1}{\lfloor \alpha \cdot n \rfloor} \sum_{i=1}^{\lfloor \alpha \cdot n \rfloor} (u_{(i)}(\hat{\boldsymbol{\beta}}_n))^2.$$

In comparison with the MSE, now we sum the squares only of the first $\lfloor \alpha \cdot n \rfloor$ smallest residuals. In further, we assume $\alpha = 0.75$ for evaluating initial estimators and $\alpha \in \{0.7, 0.8, 0.9\}$ elsewhere.

Weighted mean square error

Similarly to TMSE, also in terms of weighted mean square error we assign non-zero weights only to $\lfloor \alpha \cdot n \rfloor$ smallest residuals. However, in comparison with TMSE, the rest of residuals is weighted by means of an error function, which takes exactly the form as we have already seen in the expression (1.9).

Definition 16. *For a given constant* $\alpha \in [1/2, 1)$, the weighted mean square error (WMSE) *is defined by an expression*

$$\mathsf{WMSE}(\hat{\boldsymbol{\beta}}_n) = \frac{1}{\lfloor \alpha \cdot n \rfloor} \sum_{i=1}^{\lfloor \alpha \cdot n \rfloor} w\left(\frac{i - 1/2}{n}\right) (u_{(i)}(\hat{\boldsymbol{\beta}}_n))^2, . \tag{1.13}$$

where $w(\cdot)$ *is the error function from* (1.9).

Reader may notice that similarly to MSE and TMSE, the sum from (1.13) is divided by $\frac{1}{\lfloor \alpha \cdot n \rfloor}$, even though the sum of weights $\sum_{i=1}^{\lfloor \alpha \cdot n \rfloor} w\left(\frac{i-1/2}{n}\right)$ does not equal to $\lfloor \alpha \cdot n \rfloor$. There is no issue with this, as we use these metrics later exclusively to compare two or more estimators - it is not then as important whether their value is divided by some constant or not.

In the simulations chapter, we proceed with $\alpha = 0.75$ or $\alpha \in \{0.7, 0.8, 0.9\}$ again.

1.7 Computational aspects of estimators

As we have already mentioned in the section 1.2, we know the explicit formulas only for LS and WLS. Other estimates are computed by means of iterations. The formal description of the so-called FAST Algorithm for LWS can be found in the Algorithm 1, which have been proposed e.g. by Kalina et al. in [13]. We modified it slightly, as we take a subsample of size $p + 1$ instead of p and as we do not divide the weighted residual sum of squares by n. That, however, should not pose any significant difference.

Algorithm 1 The FAST Algorithm for the LWS estimator of $\boldsymbol{\beta}$ in the linear regression model

Input: Data rows $(Y_i, X_{i1}, \ldots, X_{ip})$, $\quad i \in \{1, \ldots, n\}$
Input: Weight function $w(\cdot)$ or the corresponding vector of weights $\boldsymbol{w} = (w_1, \ldots, w_n)$
Input: $J = 10000$
Input: $\epsilon = 10^{-6}$
Input: $k_{max} = 30$
Output: Optimal permutation of $w_1, \ldots, w_n$ denoted as $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_n)^T$
Output: $\hat{\boldsymbol{\beta}}_{LWS}$

for $j = 1$ to J **do**
  $m^{j0} := \infty$
  Select randomly $p+1$ observations
  Estimate $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}^{j0}_{LS} = (\hat{\beta}^{j0}_1, \ldots, \hat{\beta}^{j0}_p)^T$ obtained as the least squares estimator in the linear regression model using exactly the selected $p+1$ observations
  $k := 0$
  repeat
    $u_i(\hat{\boldsymbol{\beta}}^{jk}) := Y_i - \hat{\beta}^{jk}_1 X_{i1} - \ldots - \hat{\beta}^{jk}_p X_{ip}, \quad i \in \{1, \ldots, n\}$
    $o_1, \ldots, o_n$:= vector of order statistics computed from $u_1^2(\hat{\boldsymbol{\beta}}^{jk}), \ldots, u_n^2(\hat{\boldsymbol{\beta}}^{jk})$
    $\boldsymbol{w}^{j,k+1} := (w^{j,k+1}_1, \ldots, w^{j,k+1}_n)^T$ with

$$w^{j,k+1}_i := w\left(\frac{o_i - 1/2}{n}\right), \quad i \in \{1, \ldots, n\}$$

    Estimate $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}^{j,k+1} = (\hat{\beta}^{j,k+1}_1, \ldots, \hat{\beta}^{j,k+1}_p)^T$ obtained as the WLS estimator with weights $\boldsymbol{w}^{j,k+1}$
    $m^{j,k+1} := \sum_{i=1}^n w^{j,k+1}_i (u_i(\hat{\boldsymbol{\beta}}^{j,k+1}))^2$
    $k := k+1$
  until $m^{jk} > m^{j,k-1} - \epsilon$ **or** $k > k_{max}$
  $\tilde{\boldsymbol{\beta}}^j = (\tilde{\beta}^j_1, \ldots, \tilde{\beta}^j_p)^T := \hat{\boldsymbol{\beta}}^{j,k-1}$
  $\tilde{\boldsymbol{w}}^j = (\tilde{w}^j_1, \ldots, \tilde{w}^j_n)^T := \boldsymbol{w}^{j,k-1}$
end for
$j^* := \arg\min_j \sum_{i=1}^n \tilde{w}^j_i (u_i(\tilde{\boldsymbol{\beta}}^j))^2$
$\tilde{\boldsymbol{w}} := (\tilde{w}_1, \ldots, \tilde{w}_n)^T := (\tilde{w}^{j^*}_1, \ldots, \tilde{w}^{j^*}_n)^T$
$\hat{\boldsymbol{\beta}}_{LWS} := \tilde{\boldsymbol{\beta}}^{j^*}$, the weighted least squares estimator in the linear regression model with weights $\tilde{w}_1, \ldots, \tilde{w}_n$

1.7.1 The FAST algorithm

For simplicity it might be useful to condiser firstly only the 0-1 weights for some trimming constant $\tau \in [1/2, 1)$, as in (1.10). Our goal here is then to find the right permutation of them, which will determine the vector of weights. We start with a random subsample counting $p+1$ rows out of n and calculate the LS estimator $\hat{\boldsymbol{\beta}}^0$ using merely these $p+1$ observations. By applying estimated $\hat{\boldsymbol{\beta}}^0$ on the whole sample of size n we obtain a vector of residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0)$ of the length n. After sorting the residuals, we assign weight 1 to the first $\lfloor \alpha \cdot n \rfloor$ corresponding observations. So we obtained the permuted weight vector $\boldsymbol{w}^0$.

Then we plug-in this vector $\boldsymbol{w}^0$ of 0-1 weights into the weighted least squares estimator, which gives us another estimate $\hat{\boldsymbol{\beta}}^1$. In the same fashion as in the previous paragraph we obtain $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^1)$ and $\boldsymbol{w}^1$. We proceed in this estimator-residuals-weights pattern until we reach a set value of repetitions k_{max}, i.e. until we obtain $\hat{\boldsymbol{\beta}}^{k_{max}}$, or until we are not getting better enough from the step k to the step $k+1$. In all of our simulations, $k_{max} = 30$.

For the purpose of "not getting better" we define error constant ϵ and stop the cycle when the weighted residual sum of squares

$$\sum_{i=1}^{n} w_i^k (u_i(\hat{\boldsymbol{\beta}}^k))^2.$$

in the $(k+1)$st step is not smaller than the weighted residual sum of squares in the kth step by more than this ϵ. In mathematical terms, we break the cycle when

$$\sum_{i=1}^{n} w_i^{k+1} (u_i(\hat{\boldsymbol{\beta}}^{k+1}))^2 > \sum_{i=1}^{n} w_i^k (u_i(\hat{\boldsymbol{\beta}}^k))^2 - \epsilon.$$

By default we take $\epsilon = 10^{-6}$.

In this setting, the quality of the resulting estimate of $\boldsymbol{\beta}$ depends on how (un)well we have chosen the initial $p+1$ observations out of n. We do not know how the weighted residual sum of squares behaves as a function of $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{w}$, i.e. whether and where does it reach its local or global minima. In order not to converge only to some local minimum, which might be far from the global optimum, we repeat the procedure in the previous paragraphs J times. As the resulting LTS estimator we pick the one with the lowest weighted residual sum of squares over all J realizations. Unless stated otherwise, we proceed with $J = 10000$.

The generalization to all the other estimators is quite straightforward. While 0-1 weights are a special case of some general weight function, it suffices to exchange them for the values of a given weight function $w(\cdot)$ on $\left\{\frac{i-1/2}{n}, i \in \{1, \ldots, n\}\right\}$ (or for any other vector of non-negative weights $\boldsymbol{w}$).

Then in the algorithm, after sorting the residuals we assign the weight $w\left(\frac{i-1/2}{n}\right)$ (or the ith largest element of the vector of weights $\boldsymbol{w}$, respectively) to the observation with the ith smallest residual. The estimator-residuals-weights pattern then follows in the same way as above.

1.8 Variability of estimators

Now we focus on the precision aspect of an estimator, i.e. how variable its estimates are as random variables. In this thesis, we apply and compare two different methods of estimating the variance of a given estimator $\hat{\boldsymbol{\beta}}$. Firstly, we explain the White's sandwich estimator of the covariance matrix $\mathsf{var}\,\hat{\boldsymbol{\beta}}$. However, this approach is theoretically sound only for LS and WLS estimators. Nevertheless, as our intention in this paper is to explore some properties primarily of the LWS estimator, we include one "guess" of a decent LWS variance estimator belonging to the family of sandwich methods.

Secondly, the nonparametric bootstrap algorithm is introduced. An advantage of this quite universal procedure is that it can be applied to every considered estimator. Later on, the simulations compare both of these approaches (where it is possible).

1.8.1 Sandwich estimator for the least squares

Denote $\mathbf{V}_X = \mathsf{E}\boldsymbol{X}_i\boldsymbol{X}_i^\top$ the covariance matrix of $\boldsymbol{X}_i$ and $\mathbf{S}_X = \mathsf{E}\sigma^2(\boldsymbol{X}_i)\boldsymbol{X}_i\boldsymbol{X}_i^\top$ (now there is no conditioning). Assume both $\mathbf{V}_X$ and $\mathbf{S}_X$ are finite (i.e., all of their elements are finite) and positive semidefinite.

Then in the linear regression model it holds that

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{LS} - \boldsymbol{\beta}) \xrightarrow{D} \mathcal{N}_p(\mathbf{0}, \mathbf{V}_X^{-1}\mathbf{S}_X\mathbf{V}_X^{-1}) \tag{1.14}$$

by White [14], for $\mathbf{0} = (0, \dots, 0)^\top \in \mathbb{R}^p$ and $\mathcal{N}_p$ denoting the p-dimensional normal distribution. Again, by this notation we mean that the left-hand side converges in distribution to the random variable with a distribution given by the right-hand side of the expression.

For $\sigma^2(\cdot) \equiv \sigma_e^2$ we can simplify (1.14) to

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{LS} - \boldsymbol{\beta}) \xrightarrow{D} \mathcal{N}_p(\mathbf{0}, \sigma_e^2\,\mathbf{V}_X^{-1}). \tag{1.15}$$

However, in further we work under the more general assumption, where $\sigma^2(\boldsymbol{X}_i)$ is a function of the regressors, with requirements as in the Definition 1.

With a finite data sample, the true distribution of $\boldsymbol{X}_i$ is unknown and hence we do not know how the matrices $\mathbf{V}_X$ and $\mathbf{S}_X$ look like. Nevertheless, we can estimate them sufficiently. Defining

$$\widehat{\mathbf{V}}_X = \frac{1}{n}\left(\mathbf{X}^\top\mathbf{X}\right) = \frac{1}{n}\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^\top,$$

then from the assumption of finite $\mathbf{V}_X$ and element-wise application of the Weak Law of Large Numbers (Dupač and Hušková [15, Theorem 4.3]) we get $\widehat{\mathbf{V}}_X \xrightarrow{P} \mathbf{V}_X$. In other words, we get the weak consistency of $\widehat{\mathbf{V}}_X$.

Regarding a weakly consistent estimator of $\mathbf{S}_X$, denote

$$\mathbf{U} = \mathsf{diag}((u_1(\hat{\boldsymbol{\beta}}_{LS}))^2, \dots, (u_n(\hat{\boldsymbol{\beta}}_{LS}))^2) \tag{1.16}$$

and let

$$\widehat{\mathbf{S}}_X = \frac{1}{n}\left(\mathbf{X}^\top\mathbf{U}\mathbf{X}\right) = \frac{1}{n}\sum_{i=1}^n (u_i(\hat{\boldsymbol{\beta}}_{LS}))^2\boldsymbol{X}_i\boldsymbol{X}_i^\top.$$

Then, we reach our targeted result by means of the following theorem (for proof see White [14]):

Theorem 1 (Weak consistency of $\widehat{\mathbf{S}}_X$)**.** *Let for each component X_{ij}, $j \in \{1, \ldots, p\}$ of the vector $\boldsymbol{X}_i$ hold $\mathsf{E} X_{ij}^4 < \infty$ and*

$$\mathsf{E}\,\sigma^2(\boldsymbol{X}_i)\left|X_{ij}^3\right| < \infty.$$

Then, $\widehat{\boldsymbol{S}}_X \xrightarrow{P} \boldsymbol{S}_X$.

Corollary. The matrix $\widehat{\mathbf{V}}_X^{-1}\widehat{\mathbf{S}}_X\widehat{\mathbf{V}}_X^{-1}$ is a weakly consistent estimator of the covariance matrix $\mathbf{V}_X^{-1}\mathbf{S}_X\mathbf{V}_X^{-1}$ (from the right-hand side of (1.14)).

Proof. The statement of the corollary follows from the Continuous Mapping Theorem (Mann and Wald [16]). ☐

In terms of the least squares estimator $\hat{\boldsymbol{\beta}}_{LS}$, from (1.14) for large enough n we can write

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{LS} - \boldsymbol{\beta}) \dot{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{V}_X^{-1}\mathbf{S}_X\mathbf{V}_X^{-1})$$

From here, by the way of equivalent adjustments we arrive at

$$\hat{\boldsymbol{\beta}}_{LS} \dot{\sim} \mathcal{N}_p(\boldsymbol{\beta}, \frac{1}{n}\mathbf{V}_X^{-1}\mathbf{S}_X\mathbf{V}_X^{-1}). \tag{1.17}$$

The matrix product on the right hand side can be weakly consistently estimated by the statement from the corollary. Altogether,

$$\widehat{\mathsf{var}}(\hat{\boldsymbol{\beta}}_{LS}) = \frac{1}{n}\widehat{\mathbf{V}}_X^{-1}\widehat{\mathbf{S}}_X\widehat{\mathbf{V}}_X^{-1} = \frac{1}{n}\left[\frac{1}{n}\left(\mathbf{X}^\top\mathbf{X}\right)\right]^{-1}\left[\frac{1}{n}\left(\mathbf{X}^\top\mathbf{U}\mathbf{X}\right)\right]\left[\frac{1}{n}\left(\mathbf{X}^\top\mathbf{X}\right)\right]^{-1} =$$
$$= \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\left(\mathbf{X}^\top\mathbf{U}\mathbf{X}\right)\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \tag{1.18}$$

is a weakly consistent estimator of $\mathsf{var}(\hat{\boldsymbol{\beta}}_{LS})$. It is usually referred to as the White's Sandwich Estimator.

1.8.2 Asymptotic distribution of the weighted least squares

Now we try to get some insight into the asymptotic distribution of the weighted least squares. Here we assume that by means of the data-dependent weights, we find ourselves in the exactly same situation as in the example from the beginning of this chapter, i.e. the weights of the individual observations are known (for now, regardless the precision). Next, for a vector of data-dependent weights $\boldsymbol{w} = (w_1, \ldots, w_n)$ resulting from the initial estimation, we might define a matrix

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$$

in the same way as we did it in that mentioned example. Returning to the transformation from that section 1.2 (multiplying the original regressors, response and error terms by the matrix $\mathbf{W}^{1/2}$ from the left) and keeping the notation from there, we know that

$$\boldsymbol{Y}' = \mathbf{X}'\boldsymbol{\beta}' + \boldsymbol{\varepsilon}' \tag{1.19}$$

satisfies a definition of a homoscedastic LRM - which is in accordance with the main idea of the data-dependent weights explained in 1.5 as well.

Then, plugging into (1.15) we get

$$\sqrt{n}(\hat{\boldsymbol{\beta}}'_{LS} - \boldsymbol{\beta}') \xrightarrow{D} \mathcal{N}_p(\mathbf{0}, \sigma_e^2\, \mathbf{V}_{X'}^{-1}). \tag{1.20}$$

In a similar fashion as above,

$$\widehat{\mathbf{V}}_{X'} = \frac{1}{n}\left((\mathbf{X}')^\top \mathbf{X}'\right) = \frac{1}{n}\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right) = \frac{1}{n}\sum_{i=1}^{n} w_i \cdot \boldsymbol{X}_i \boldsymbol{X}_i^\top$$

is again a weakly consistent estimator of $\mathbf{V}_{X'}$. Next, similarly as in (1.17), we can rewrite (1.20) as

$$\hat{\boldsymbol{\beta}}_{WLS} = \hat{\boldsymbol{\beta}}'_{LS} \dot{\sim} \mathcal{N}_p(\boldsymbol{\beta}', \frac{1}{n}\sigma_e^2 \mathbf{V}_{X'}^{-1}).$$

Hence, under our assumptions,

$$\widehat{\mathsf{var}}(\hat{\boldsymbol{\beta}}_{WLS}) = \frac{1}{n}\hat{\sigma}_e^2 \widehat{\mathbf{V}}_{X'}^{-1} = \frac{1}{n}\hat{\sigma}_e^2 \left[\frac{1}{n}\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)\right]^{-1} = \hat{\sigma}_e^2 \left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{-1} \tag{1.21}$$

is a consistent estimator of $\mathsf{var}(\hat{\boldsymbol{\beta}}_{WLS})$, if $\hat{\sigma}_e^2$ is a consistent estimator of σ_e^2. As such an estimator $\hat{\sigma}_e^2$ with this property we can take the basic $\hat{\sigma}_e^2 = \frac{SS'_e}{n-p}$, where SS'_e is the residual sum of squares in the model (1.19). It is not hard to derive that

$$SS'_e = \sum_{i=1}^{n}(Y'_i - (\boldsymbol{X}'_i)^\top \hat{\boldsymbol{\beta}}_{WLS})^2 = \sum_{i=1}^{n}(w_i^{1/2} \cdot (Y_i - \boldsymbol{X}_i^\top \hat{\boldsymbol{\beta}}_{WLS}))^2 = \sum_{i=1}^{n} w_i \cdot (u_i(\hat{\boldsymbol{\beta}}_{WLS}))^2,$$

which is nothing but the weighted residual sum of squares in the original model. Altogether, the covariance matrix estimator for WLS we are going to test numerically in the second chapter takes the form

$$\widehat{\mathsf{var}}(\hat{\boldsymbol{\beta}}_{WLS}) = \frac{1}{n-p}\left(\sum_{i=1}^{n} w_i \cdot (u_i(\hat{\boldsymbol{\beta}}_{WLS}))^2\right)\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{-1}. \tag{1.22}$$

1.8.3 Sandwich estimator for the least weighted squares

While researching for this thesis, an article from Víšek [4] regarding the sandwich variance estimator for LWS was found. However, his formula seemed not to work well on the simulated data. Instead, we modified it slightly to the form which follows.

Keeping the notation from the subsection 1.8.1, firstly we exchanged the LS estimator in the definition of the matrix $\mathbf{U}$ for LWS, i.e.

$$\mathbf{U}_{LWS} = \mathsf{diag}((u_1(\hat{\boldsymbol{\beta}}_{LWS}))^2, \ldots, (u_n(\hat{\boldsymbol{\beta}}_{LWS}))^2).$$

Then, the estimate of the matrix $\mathbf{S}_X^{LWS}$ was defined as

$$\widehat{\mathbf{S}}_X^{LWS} = \frac{1}{n}\left(\mathbf{X}^\top \mathbf{U}_{LWS}\mathbf{W}^f \mathbf{X}\right) = \frac{1}{n}\left(\mathbf{X}^\top \mathbf{W}^f \mathbf{U}_{LWS}\mathbf{X}\right) = \frac{1}{n}\sum_{i=1}^{n} w_i^f \cdot (u_i(\hat{\boldsymbol{\beta}}_{LWS}))^2 \boldsymbol{X}_i \boldsymbol{X}_i^\top, \tag{1.23}$$

where $\mathbf{W}^f$ is again a diagonal matrix with weights $\boldsymbol{w}^f = (w_1^f, \ldots, w_n^f)^\top$ on its diagonal, but this time they are ordered according to the optimal permutation in terms of the lowest residual sum of squares (in praxis, by applying the FAST algorithm we might not stumble over the globally optimal permutation; but for sufficiently large constants J and k_{max} we usually get at least close enough; for details regarding the algorithm see the subsection 1.7.1).

The letter f in the superscript refers to the adjective "final", i.e. to the permutation of weights that has been evaluated as the best among all considered. In the second equality we used that both of the matrices $\mathbf{U}_{LWS}$ and $\mathbf{W}^f$ are diagonal.

Finally, we set the sandwich variance estimator for LWS very similarly to the one for LS, namely

$$\widehat{\mathsf{var}}(\hat{\boldsymbol{\beta}}_{LWS}) = \frac{1}{n}\widehat{\mathbf{V}}_X^{-1}\hat{\mathbf{S}}_X^{LWS}\widehat{\mathbf{V}}_X^{-1} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\left(\mathbf{X}^\top\mathbf{U}_{LWS}\mathbf{W}^f\mathbf{X}\right)\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \quad (1.24)$$

is our "guess" estimator of $\mathsf{var}(\hat{\boldsymbol{\beta}}_{LWS})$.

As already noted before, this variance estimator is not supported by the theory (at least to the author's knowledge), neither for restricting ourselves to fixed weights designs only. We do not know, whether it estimates $\mathsf{var}(\hat{\boldsymbol{\beta}}_{LWS})$ consistently.

However, to our hypothesis, it seems to fit better to the general theory of weighted estimators from the literature than if the term $\mathbf{W}^f$ from the expression (1.23) was omitted. In favor of this claim, the resulting variance estimates (see simulations chapter) are quite close to the ones obtained from the bootstrap algorithm, which we introduce right now.

1.8.4 Nonparametric bootstrap

The nonparametric bootstrap technique serves as a common way in statistics how to estimate (except others) the variability of a certain random variable. The algorithm proceeds in a way that is captured in the Algorithm 2. We slightly reformulated some details of the Algorithm 2 presented by Kalina et al. [13], but the core remains unchanged. In our computations we proceed with $R = 100$ bootstrap replicates.

The chosen value of R is definitely at the lower end of the range of values, for which are bootstrap estimates reliable (Alipour et al. [17]). Nevertheless, as the FAST algorithm is computationally quite expensive, we conducted the following trade-off: for to be able to perform at least these 100 replicates in a sensible time domain, we lowered the number of iterations in the FAST algorithm J from 1.7.1 from 10000 to 1000 (only when simulating via the bootstrap algorithm).

From the mere empirical experience, in our model settings $R = 100$ is sufficient. Even though we do not state this claim in any way rigorously, across all the simulations the results with $R = 100$ were quite consistent.

The output from the bootstrap algorithm is the empirical covariance matrix. The wanted vector of variance estimates $(\widehat{\mathsf{var}}(\hat{\beta}_1), \ldots, \widehat{\mathsf{var}}(\hat{\beta}_p))$ is found on its diagonal.

Algorithm 2 Nonparametric bootstrap

Input: Data rows $(Y_i, X_{i1}, \ldots, X_{ip}), \quad i \in \{1, \ldots, n\}$
Input: $R > 0$
Output: Empirical covariance matrix $\hat{\boldsymbol{\kappa}}$ computed from the individual estimates of the model covariance matrix $\boldsymbol{\kappa}$

1: Compute the estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ in the linear regression model
2: **for** $r = 1$ to R **do**
3: Generate n new bootstrap data rows

$$({}_rY_j^*, {}_rX_{j1}^*, \ldots, {}_rX_{jp}^*), \quad j \in \{1, \ldots, n\},$$

by sampling with replacement from the data rows $(Y_i, X_{i1}, \ldots, X_{ip}), \quad i \in \{1, \ldots, n\}$
4: Consider a linear regression model

$${}_rY_j^* = {}_r\beta_{1r}X_{j1}^* + \ldots + {}_r\beta_{pr}X_{jp}^* + {}_rv_j, \quad j \in \{1, \ldots, n\},$$

with random errors ${}_rv_j, \quad j \in \{1, \ldots, n\}$
5: Estimate ${}_r\boldsymbol{\beta} = ({}_r\beta_1, \ldots, {}_r\beta_p)^T$ by ${}_r\hat{\boldsymbol{\beta}}$ and store the value
6: **end for**
7: Compute the empirical covariance matrix $\hat{\boldsymbol{\kappa}}$ from values ${}_r\hat{\boldsymbol{\beta}}, \quad r \in \{1, \ldots, R\}$

One further note regarding using bootstrap in our simulations can be found in A.1. We tried to introduce another coefficient estimator based on these R bootstrap replicates, but we did not see any interesting results.

1.9 Using R libraries

LWS and LTS estimators were implemented in R manually by the author of this thesis, using only functions from the R base library [18] (except some initial estimators). They can be found as attachments to this thesis and their possibly updated versions will appear in [19].

This author's own implementation of LTS is however of much higher computational demands compared to the LTS estimator loaded from the library `MASS` [20] via the function `lqs`. Except of this issue, the self-implemented estimator performed almost identically as the LTS from `lqs` in all of the simulations. For the purpose of optimizing the initial estimator even from the time perspective, only the LTS from the library `MASS` was used.

The function `lqs` from `MASS` occurred in the underlying codes for the simulations two more times. Namely, when computing LMS and S-estimators. As was already covered in the section 1.7, the application of these two estimator requires iterations as well. In both cases only the number of iterations was modified during the optimization of the initial estimator. In all other cases, it was proceeded with default settings.

Regarding the REWLS and MM estimators, the library `robust` [21] was used. As there are many parameters to modify, we simplified our task by letting the parameters as they were set by default in the actual library version.

2 Simulations

In the second part of the thesis, we run several simulations. We search for optimal values of some parameters regarding computation of the data-dependent weight from the section 1.5. Then, more importantly, we conduct two larger studies, each one comparing two aspects of considered estimators: accuracy (by observing the estimated coefficients $\hat{\boldsymbol{\beta}}$) and precision (by computing the variance estimation of each coefficient estimator). However, firstly we need to describe the basic model used.

2.1 Model

As the reference sample size of the linear regression model in this thesis we set $n = 200$. For $i \in \{1, \dots, n\}$, the intercept was included and we generated a random variable $X_{i,2}$ from the standard normal distribution and a random variable $X_{i,3}$ from the uniform distribution on $(0, 2)$. The response was calculated as

$$Y_i = 1 + 2 \cdot X_{i,2} - 3 \cdot X_{i,3} + \varepsilon_i,$$

where ε_i follows the standard normal $N(0, 1)$ distribution as well.

The data were usually contaminated in the following way: we randomly selected 20% of observations and in these cases the error term ε_i came from the uniform distribution on $(-50, -10) \cup (10, 50)$.

2.2 Setting the parameters for data-dependent weights

The data-dependent weights are one of the main objects of interest in this thesis. Therefore, we need to explore some of their parameters settings at first. After this section, when we will have found the optimal choices for the individual parameters, we proceed to comparing all the estimators mentioned in the first chapter and to exploring their accuracy and variability.

2.2.1 Initial estimator and the number of iterations for its computation

Here we try to evaluate the best initial estimator out of the four mentioned in the section 1.5 (LTS, LMS, S and LS estimators), as well as the optimal number of the iterations performed for its computation (why do we need to iterate was examined in the section 1.7). The results go in parallel for the data-dependent LWS and the data-dependent WLS.

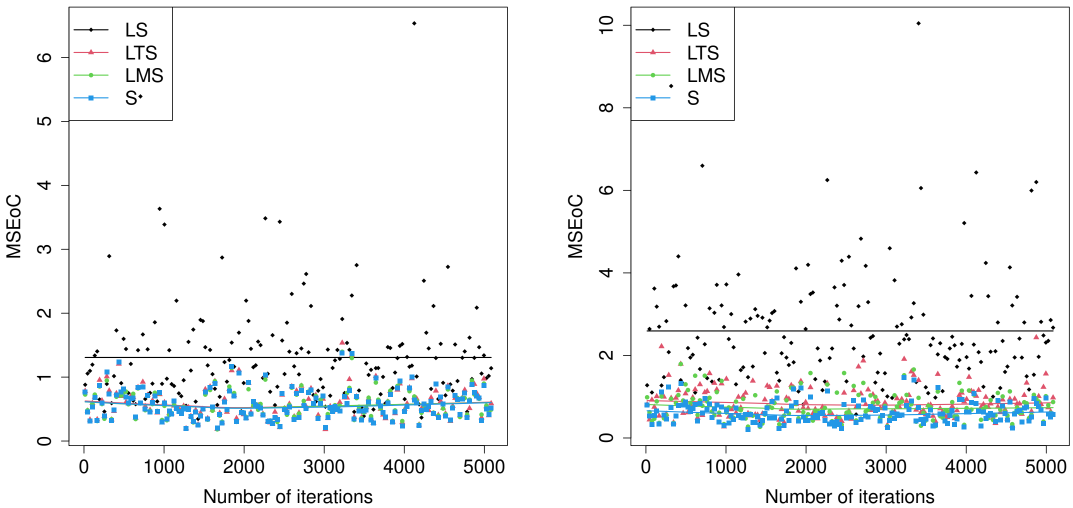
The simulations proceeded in the following way: for a fixed number of iterations m, we generated 100 datasets, each containing 200 data rows according to the design in the previous section 2.1. Based on these data we estimated the coefficients $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}_0$ by means of LTS, LMS and S estimators computed with m iterations and got the corresponding residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0)$. From these initial residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^0)$, we obtained the data-dependent weights $\hat{w}_n(t)$ according to the Definition 11 and used these in the data-dependent LWS estimator from the Definition 7 and in the data-dependent WLS estimator from the Definition 13. In all the data-dependent weights generation, the constant η from the section 1.5 was now set to 10^{-6}.

By means of this approach we got the two-step estimates $\hat{\boldsymbol{\beta}}^1_{dd-LWS}$ and $\hat{\boldsymbol{\beta}}^1_{dd-WLS}$ of $\boldsymbol{\beta}$ and the residuals $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^1_{dd-LWS})$ and $\boldsymbol{u}(\hat{\boldsymbol{\beta}}^1_{dd-WLS})$. Obtaining these quantities enabled us to calculate $\mathsf{MSEoC}(\hat{\boldsymbol{\beta}}^1)$, $\mathsf{WMSE}(\hat{\boldsymbol{\beta}}^1)$ and $\mathsf{TMSE}(\hat{\boldsymbol{\beta}}^1)$ as well (for both $\hat{\boldsymbol{\beta}}^1$ equal to $\hat{\boldsymbol{\beta}}^1_{dd-LWS}$ or $\hat{\boldsymbol{\beta}}^1_{dd-WLS}$).

The least squares estimator was approached slightly differently. As there are no iterations in the LS algorithm, we simply calculated LS estimation for every m from its formula (1.6) (independently on this particular value of m). The residuals, MSEoC and weighted residual sums of squares could be then evaluated in the same way as for other initial estimators.

For $m \in M = \{15 + 30 \cdot k, \quad k \in \{0, 1, \ldots, 169\}\}$, i.e. for the sequence from 15 to 5085 with a difference 30, we present the mean averages (out of 100 datasets) of the corresponding errors in Figures 2.1, 2.2 and 2.3. The lines are outputs of the function `lowess` from the R Stats package [18]}. Their purpose is to roughly show the trend in the data.

For the LS, as there is no dependency between m and MSEoC (or TMSE or WMSE for $\alpha = 0.75$, respectively), we plotted a simple horizontal line depicting the mean value of MSEoC (or TMSE or WMSE).



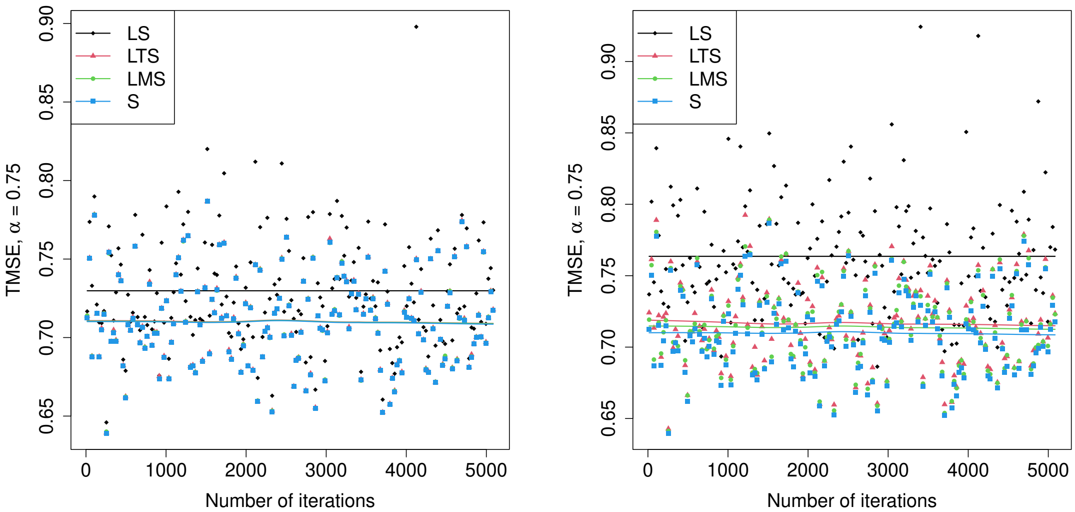
(a) MSEoC for the dd-LWS. **(b)** MSEoC for the dd-WLS.

Figure 2.1 Mean square error of coefficients with lines depicting trends in the data.

In both of the pictures, there is clearly notifiable that the least squares are not a suitable initial estimator. For both dd-LWS and especially for dd-WLS, MSEoCs are many times higher when the given estimator was combined with LS. But even when not paying attention to the LS, dd-LWS possess noticeably better results than dd-WLS in terms of MSEoC. Its values for S, LMS and LTS estimators are approximately two times lower for dd-LWS, as is visible from the x-axes of the graphs.

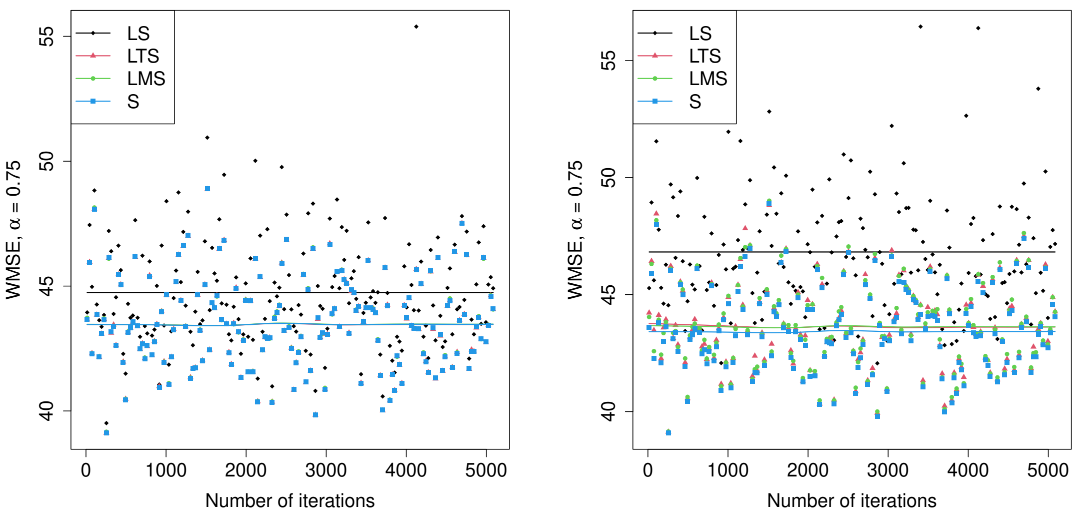
Next, regarding the other three initial estimators (excluding LS), we observe no much difference between them. Even though, as the best one it turns out to be the S-estimator, as is observable in the picture on the right. In the left for dd-LWS, S-estimator, LMS and LTS yield more or less exactly the same results.

Lastly, there is hardly any difference between all the considered numbers of iterations for computing initial estimators. All of the lowess lines seem quite well parallel with the horizontal axis.



(a) TMSE for $\alpha = 0.75$ for dd-LWS. **(b)** TMSE for $\alpha = 0.75$ for dd-WLS.

Figure 2.2 Trimmed mean square error for $\alpha = 0.75$ with with lines depicting trends in the data.



(a) WMSE for $\alpha = 0.75$ for dd-LWS. **(b)** WMSE for $\alpha = 0.75$ for dd-WLS.

Figure 2.3 Weighted mean square error for $\alpha = 0.75$ with lines depicting trends in the data.

In the Figures 2.2 and 2.3, in opposition to MSEoC we observe there is no significant discrepancy in the values on the vertical axis for dd-LWS and dd-WLS. That means that both of the estimators are roughly equally effective in terms of TMSE or WMSE for $\alpha = 0.75$.

Except of that, for Figures 2.2 and 2.3 it could have been written more or less the same as for the first Figure 2.1. We observe S, LMS and LTS are performing quite similarly, again with the notion that S-estimator seems to be a slightly better initial estimator for dd-WLS. Hence, in further we always combine dd-LWS and dd-LWS with the S-estimator.

In all of the pictures above, no significant trend of the lowess lines is visible. In other words, it seems like the number of iterations does not have any effect on the considered errors at all. Hence, we stay at the lower end and proceed with the number $m = 555$ for both of the estimators.

2.3 Main simulations

Having answered the questions from the first chapter about several parameters for generating data-dependent weights, we proceed to main simulations. One of the biggest goals of this thesis is to compare various coefficient estimators and their variance.

2.3.1 Basic model design

Now, we generated 200 datasets according to our design from the section 2.1 and for each one, we employed all of the estimators introduced in the Chapter 1.

Firstly we focus on the estimated coefficients. In the same manner as in the previous section, we compare MSEoC and TMSE and WMSE for $\alpha \in \{0.7, 0.8, 0.9\}$ of various estimators of $\boldsymbol{\beta}$. For each single data sample, the given error metric was computed and the mean averages of these 200 error values were captured in the Table 2.1.

In all of the tables below, by the short notation LWS-X for X a natural number from 1 to 5, we mean combination of LWS with the weight function LWS-X, as was described in the first chapter in the section 1.3.

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	9.588	2.608	4.204	51.991	1.181	1.579	12.092
LWS-1	0.766	0.687	1.215	**50.149**	0.305	0.441	**11.262**
LWS-2	0.165	0.562	1.016	50.438	0.248	0.365	11.299
LWS-3	0.134	0.551	1.004	50.604	**0.242**	0.360	11.334
LWS-4	3.105	1.183	2.017	50.568	0.529	0.741	11.464
LWS-5	0.073	0.547	0.983	50.595	0.244	0.356	11.333
LTS	0.277	0.577	1.056	50.653	0.252	0.376	11.349
REWLS	**0.050**	0.554	**0.973**	50.597	0.247	**0.354**	11.334
dd-LWS	0.058	0.548	0.977	50.576	0.244	**0.354**	11.328
dd-WLS	0.064	0.547	0.979	50.583	0.243	**0.354**	11.329
MM	**0.050**	0.553	**0.973**	50.598	0.247	**0.354**	11.334
LMS	0.240	0.580	1.055	50.664	0.256	0.379	11.354
S	0.146	**0.546**	0.992	50.598	**0.242**	0.357	11.333

Table 2.1 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design.

In the Table 2.1 we might observe that in terms of MSEoC, the MM-estimator and REWLS performed the best. However, the two estimators of our focus, dd-LWS and dd-WLS, did not end up much behind.

From the other end of the spectrum, the least squares performed poorly in all of the captured metrics. As they were the least robust estimator among all considered, this behavior came as no surprise. However, the LWS combined with weight function LWS-4 (weights generated by error function from the section 1.3) scored unexpectedly badly - not only in terms of MSEoC, but in terms of TMSE and WMSE for $\alpha \in \{0.7, 0.8\}$ as well. Up till now, we have no explanation for that phenomenon.

When it comes down to TMSE and WMSE for various α, the difference between adaptive weighting schemes (REWLS, dd-LWS, dd-WLS, MM) and other estimators is better observable for lower values of α. Estimators with data-dependent weights usually perform better then. In other words, these adaptive estimators score better in terms of more robust metrics (as for lower values of α we trim out more observations and hence get more robust). This was the behavior that was observed throughout most of other simulation designs as well - the tables with further results can be found in A.3.

Even though only very marginally, both in TMSE and WMSE for $\alpha = 0.7$, dd-LWS and dd-WLS performed better than MM-estimator or than REWLS. Here we make a notion also about the S-estimator, which did much better in terms of metrics based on residual sums of squares (TMSE and WMSE) than in MSEoC.

Moving forward to the next Table 2.2, firstly, we emphasize the different approaches, by which were the quantities in the upper (Sandwich) part of the table obtained. The sandwich LS estimator was calculated as described in the section 1.8.1, the dd-WLS followed the derivation in 1.8.2 and finally, the variance estimators of all the LWS variants result from the "guess" estimation, as was admitted in the section 1.8.3.

Nevertheless, comparing the corresponding values in both parts of the Table 2.2, we might evaluate the procedure from 1.8.3 as quite successful (at least at the considered simulation designs). In most of the cases, the obtained variance estimates do not differ much from those resulting from the bootstrap-based approach, which should be mathematically more reliable.

Regarding the estimates of variance themselves, in the upper part of the Table 2.2 exceptional results were achieved by LWS combined with the weight functions LWS-3 and LWS-5. According to the author there is no obvious explanation of this successful behavior (except of the fact, that the derivation still might not be theoretically sound). Nevertheless, in terms of the bootstrap estimation (the bottom part of the Table 2.2), even better than these LWS-3 and LWS-5 estimators did REWLS, MM, dd-LWS and dd-WLS, in this order and in all of the coordinates.

2.3.2 Heteroscedastic model design

The second model design employed in this thesis is very similar to the first one in the section 2.1, only with the following adjustment of the distribution of error terms. Now,

$$\varepsilon_i \sim e^{X_{i,1}+X_{i,2}},$$

Estimator	$\widehat{\mathsf{var}}(\hat{\beta}_1)$	$\widehat{\mathsf{var}}(\hat{\beta}_2)$	$\widehat{\mathsf{var}}(\hat{\beta}_3)$
Sandwich			
LS	4.118	1.051	3.113
LWS-1	0.261	0.066	0.198
LWS-2	0.062	0.016	0.047
LWS-3	**0.002**	**0.001**	**0.002**
LWS-4	0.773	0.198	0.585
LWS-5	0.010	0.003	0.008
dd-LWS	0.032	0.008	0.024
dd-WLS	0.033	0.008	0.025
Bootstrap			
LS	4.177	1.059	3.182
LWS-1	0.437	0.124	0.341
LWS-2	0.096	0.026	0.072
LWS-3	0.071	0.020	0.053
LWS-4	1.429	0.409	1.116
LWS-5	0.047	0.012	0.035
LTS	0.157	0.043	0.114
REWLS	**0.025**	**0.007**	**0.019**
dd-LWS	0.033	0.008	0.025
dd-WLS	0.037	0.010	0.028
MM	0.026	0.007	0.020
LMS	0.165	0.042	0.122
S	0.081	0.021	0.061

Table 2.2 Comparison of variance estimators of each of the coefficients in the basic model design.

which is one of the heteroscedastic model formulas. Note that here, even the assumption 1.2 from the Definition 1 is violated. Nevertheless, we proceed and see what are the results.

Our aim is to observe, whether there are any significant differences in the behavior of the estimators, when we switch from the model with outliers to this one. Here, the heteroscedasticity is present in different form than it was in the first model (where the variance of error terms of those 20% selected outlying observations was different from the unit variance of the standard normal distribution, hence the heteroscedasticity).

Tables 2.3 and 2.4 were generated in the same way as Tables 2.1 and 2.2. In terms of TMSE and WMSE for $\alpha = 0.7$ in the Table 2.3, the best results were achieved by estimators LWS-3 and S, while LWS-5 and dd-WLS ending behind very close. Interestingly, other "winners" stand from these metrics for $\alpha \in \{0.8, 0.9\}$, namely MM and LWS-1 estimators, respectively.

Here, however, the weights generated by the logistic function (weight function LWS-2; again, we refer to the enumeration of weight functions in 1.3) deserve attention as well. Not only because in terms of residual sums of squares they reached quite decent results and even beat some data-adaptive estimators in some

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	5.241	3.295	5.760	12.003	1.450	2.088	3.402
LWS-1	0.389	2.248	4.473	**10.377**	0.944	1.539	**2.803**
LWS-2	**0.311**	2.221	4.495	10.510	0.925	1.536	2.826
LWS-3	0.361	**2.207**	4.529	10.615	**0.918**	1.543	2.850
LWS-4	2.007	2.597	4.941	11.060	1.113	1.734	3.037
LWS-5	0.568	2.209	4.451	10.559	0.933	1.531	2.842
LTS	0.528	2.305	4.667	10.810	0.953	1.588	2.907
REWLS	0.463	2.228	4.448	10.468	0.937	1.531	2.821
dd-LWS	0.335	2.230	4.477	10.420	0.933	1.535	2.809
dd-WLS	0.341	2.209	4.498	10.514	0.922	1.536	2.827
MM	0.621	2.268	**4.405**	10.443	0.959	**1.528**	2.822
LMS	0.984	2.341	4.724	10.924	0.979	1.618	2.947
S	0.465	**2.207**	4.549	10.651	0.919	1.549	2.860

Table 2.3 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the heteroscedastic model design.

aspects, but more importantly, because they scored the best regarding MSEoC. In other words, their coefficient estimates are the most exact ones as far as the mean square metric from Definition 14 is concerned.

Hence in applications, where the accuracy of estimated coefficients matters and non-negligible heteroscedasticity might be present, applying LWS combined with the weight function LWS-2 might be a good idea.

On the other hand, for example the MM-estimator, very common tool in robust regression, scored almost twice as badly in MSEoC compared to LWS-2.

In the last Table 2.4 we focus on the bootstrap-derived estimates of variance of the individual coefficients. Here, the best performance was achieved by dd-LWS, with LWS-2 and dd-WLS closely behind.

Even under this kind of heteroscedasticity the least squares performed unsurprisingly the worst and again, the weights generated by the error function somehow did not score as well as almost all the other estimators. Perhaps more simulations and analyses are needed to better understand the “bad” behavior of LWS-4.

Estimator	$\widehat{\mathsf{var}}(\hat{\beta}_1)$	$\widehat{\mathsf{var}}(\hat{\beta}_2)$	$\widehat{\mathsf{var}}(\hat{\beta}_3)$
Sandwich			
LS	0.898	1.730	1.777
LWS-1	0.044	0.025	0.054
LWS-2	0.021	0.017	0.026
LWS-3	**0.007**	**0.001**	**0.005**
LWS-4	0.165	0.290	0.310
LWS-5	0.039	0.007	0.031
dd-LWS	0.072	0.041	0.088
dd-WLS	0.073	0.042	0.090
Bootstrap			
LS	0.903	1.660	1.720
LWS-1	0.104	0.121	0.200
LWS-2	0.096	0.107	0.176
LWS-3	0.119	0.150	0.233
LWS-4	0.293	0.747	0.767
LWS-5	0.157	0.233	0.351
LTS	0.192	0.193	0.344
REWLS	0.130	0.167	0.279
dd-LWS	**0.092**	**0.093**	**0.164**
dd-WLS	0.101	0.120	0.194
MM	0.159	0.220	0.350
LMS	0.325	0.305	0.557
S	0.167	0.179	0.303

Table 2.4 Comparison of variance estimators of each of the coefficients in the heteroscedastic model design.

Conclusion

In this thesis, the least squares, the weighted least squares and the least weighted squares estimators in linear regression were introduced. Next, the theory behind a special type of data-dependent weights was presented and asymptotic distributions of the estimators of our interest were derived (or at least, some ideas were provided where no theory was available). In the rest of the first chapter, preparing the field for the upcoming simulations was finished by introducing robust metrics of errors, as well as nonparametric bootstrap.

In the second chapter, by means of simulations we evaluated the optimal initial estimator for considered data-adaptive weights. It turned out to be the S-estimator, but the number of iterations seemed not to play any role in quality of the resulting two-step least weighted squares estimation.

Finally, main simulations were conducted on two models. In both of them, least weighted squares showed promising results either in combination with adaptive and with fixed weighting schemes. Similarly, newly proposed weighted least squares with the special type of data-dependent weights beat in some aspects other considered, more commonly used estimators.

Nevertheless, the space for further analyses and simulations by modifying other parameters is still quite big. In this thesis only one sample size was analyzed, which did not enable us aim our attention much on the efficiency of presented estimators. Similarly, in further work much more designs of dataset contamination (e.g. by leverage points) and other distributions of error terms should be included, in order to compare the estimators more thoroughly and better understand some of their behavior. Lastly, larger number of regressors in the model could be considered in the future work as well.

Bibliography

1. Mareš, Martin (ed.). *Šablona závěrečné práce na MFF UK v LaTeXu* [online]. [visited on 2024-03-02]. Available from: `https://gitlab.mff.cuni.cz/teaching/thesis-templates/thesis-cs`.
2. Víšek, Jan Ámos. Robust error-term-scale estimate. *IMS Collections*. 2010, vol. 7, no. 1, pp. 254–267.
3. Kalina, Jan. Testing heteroscedasticity in robust regression. *Research Journal of Economics Business and ICT*. 2011, vol. 4, pp. 25–28.
4. Víšek, Jan Ámos. Heteroscedasticity Resistant Robust Covariance Matrix Estimator. *Bulletin of the Czech Econometric Society*. 2010, vol. 17, no. 27, pp. 33–49.
5. Anděl, J. *Základy matematické statistiky*. Praha: Matfyzpress, 2011. Vydání třetí. isbn 978-80-7378-162-0.
6. Kalina, Jan; Tichavský, Jan. On Robust Estimation of Error Variance in (Highly) Robust Regression. *Measurement Science Review*. 2020, vol. 20, no. 1, pp. 6–14.
7. Čížek, P. Semiparametrically weighted robust estimation of regression models. *Computational Statistics and Data Analysis*. 2011, vol. 55, no. 282, pp. 774–788.
8. Rousseeuw, Peter; Leroy, Annick. Robust Regression and Outlier Detection. *Journal of Educational Statistics*. 1987, vol. 13, pp. 358–364.
9. Hampel, Frank; Ronchetti, Elvezio; Rousseeuw, Peter; Stahel, Werner. *Robust Statistics: The Approach Based on Influence Functions*. 1986. isbn 9780471735779.
10. Fishbone, Justin; Mili, Lamine. New highly efficient high-breakdown estimator of multivariate scatter and location for elliptical distributions. *Canadian Journal of Statistics*. 2024, vol. 52, no. 2, pp. 437–460. Available also from: `https://onlinelibrary.wiley.com/doi/abs/10.1002/cjs.11770`.
11. Yohai, Victor. High Breakdown-Point and High Efficiency Robust Estimates for Regression. *The Annals of Statistics*. 1987, vol. 15.
12. Gauss, C. F.; Dieterich, H. *Theoria combinationis observationum erroribus minimis obnoxiae*. Apud Henricus Dieterich, 1823.
13. Kalina, Jan; Tobišková, Nicole; Tichavský, Jan. *37th International Conference on Mathematical Methods in Economics 2019*. A Nonparametric Bootstrap Comparison of Variability of Robust Regression Estimators. České Budějovice, Czech Republic: University of South Bohemia, 2019.
14. White, H. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*. 1980, vol. 48, no. 4, pp. 817–838.
15. Dupač, V.; Hušková, M. *Pravděpodobnost a matematická statistika*. Praha: Karolinum, 2013. Druhé upravané vydání. isbn 978-80-246-2208-8.

16. Mann, H. B.; Wald, A. On Stochastic Limit and Order Relationships. *The Annals of Mathematical Statistics*. 1943, vol. 14, no. 3, pp. 217–226.

17. Alipour, Masoud; Stamatakis, Alexandros; Pattengale, N.; Moret, Bernard; Bininda-Emonds, Olaf. How Many Bootstrap Replicates are Necessary? In: 2010, vol. 17, pp. 337–354. isbn 9783642020070. Available from doi: `10.1007/978-3-642-02008-7_13`.

18. R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2022. Available also from: `https://www.R-project.org/`.

19. Jelínek, Michal. *Updated Versions of Codes Underlying My Bachelor Thesis*. 2025. Available also from: `https://github.com/mikij29/bachelor_thesis_codes`.

20. Venables, W. N.; Ripley, B. D. *Modern Applied Statistics with S*. Fourth. New York: Springer, 2002. Available also from: `https://www.stats.ox.ac.uk/pub/MASS4/`. ISBN 0-387-95457-0.

21. Wang, Jiahui; Zamar, Ruben; Marazzi, Alfio; Yohai, Victor; Salibian-Barrera, Matias; Maronna, Ricardo; Zivot, Eric; Rocke, David; Martin, Doug; Maechler, Martin; Konis., Kjell. *robust: Port of the S+ "Robust Library"*. 2024. Available also from: `https://CRAN.R-project.org/package=robust`. R package version 0.7-4.

List of Figures

List of Tables

A Attachments

A.1 Bootstrap mean estimator

Reader may have noticed that per one replication of the nonparametric bootstrap algorithm, as described in the subsection 1.8.4, we obtained R estimates of the regression coefficients as well. In order to gain some information from these quantities (and apply the metric MSEoC from the Definition 14 later), we might have created an estimator of $\boldsymbol{\beta}$ based on these R vectors of length p.

Definition 17. *We define the* bootstrap mean estimator *based on R samples as*

$$ {}_R\hat{\boldsymbol{\beta}}^{\mathsf{mean}} = ({}_R\hat{\beta}_1^{\mathsf{mean}}, \dots, {}_R\hat{\beta}_p^{\mathsf{mean}})^T, \tag{A.1} $$

where for $i \in \{1, \dots, p\}$ $\quad {}_R\hat{\beta}_i^{\mathsf{mean}} = \frac{1}{R}\sum_{i=1}^{R} {}_r\hat{\beta}_i$.

Note that this *bootstrap mean estimator* is an alternative approach to estimate model coefficients by means of any estimator. Commonly, we get a coefficient estimator by one given expression or an algorithm. The point is that here we simulate $R = 100$ such an estimators, even though every time on a possibly slightly "degenerated" data sample (a single data row might be present multiple times in the sampled dataset for a given bootstrap replication). Then we take mean values in every single coordinate of the vector of estimated coefficients.

Of course, as already stated in 1.8.4, in the case of applying bootstrap technique on the LWS estimator, we obtain slightly worse coefficient estimates as we lowered the number of iterations J in the FAST algorithm. Nevertheless, we tested this idea numerically - however, all of the results were very similar to those of regular estimators. Therefore, we decided not to present this bootstrap mean estimator in the main thesis text. The same holds for analogously defined *bootstrap median estimator*.

A.2 Choice of constant η for data-dependent weights

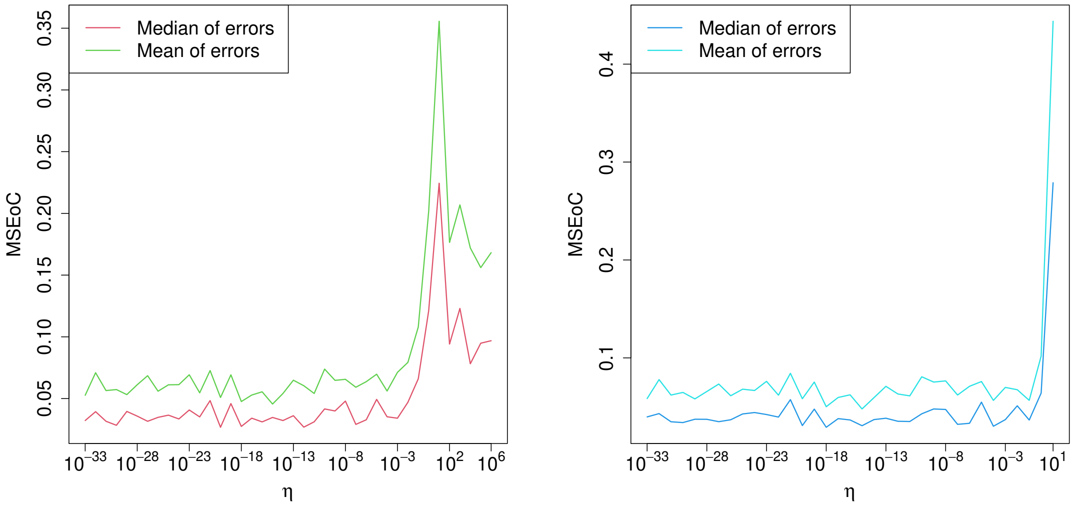
In this attachment we reveal how the choice of the constant η for software implementation of the data-dependent weights (see the note behind the Definition 11) was established. Again, as in 2.2, the simulations proceed both for the data-dependent LWS and the data-dependent WLS.

Here, the simulation design was the following: for each order of η from 10^{-33} to 10^{6} we generated 100 datasets according to the desing in the section 2.1. As was concluded in the section 2.2 - for the role of the initial estimator, the S-estimator with 555 iterations was chosen.

In the Figures A.1 and A.2 we explore, how the given metrics of errors (MSEoC and WMSE for $\alpha = 0.75$; TMSE was excluded from there, as the results are practically the same as for WMSE) change for different values of η. In particular, how does the median and the mean of errors from these 100 samples depend on η.

In the case of the dd-WLS, the x-axes in the graphs were shortened from the right end (only to $\eta = 10^1$), as the values of errors (either mean and median) explode quite quickly beyond this bound. It goes hand in hand with the intuition that for larger values of η the weights are getting further away from their purpose of bringing residuals closer to normal distribution.

Interestingly, for dd-LWS no such behavior occurred. Instead, we observed a slight peak in errors for $\eta = 10^1$. As for right now, we cannot offer any other explanation than that randomness of the whole process might have played its role.



(a) MSEoC for the dd-LWS.

(b) MSEoC for the dd-WLS.

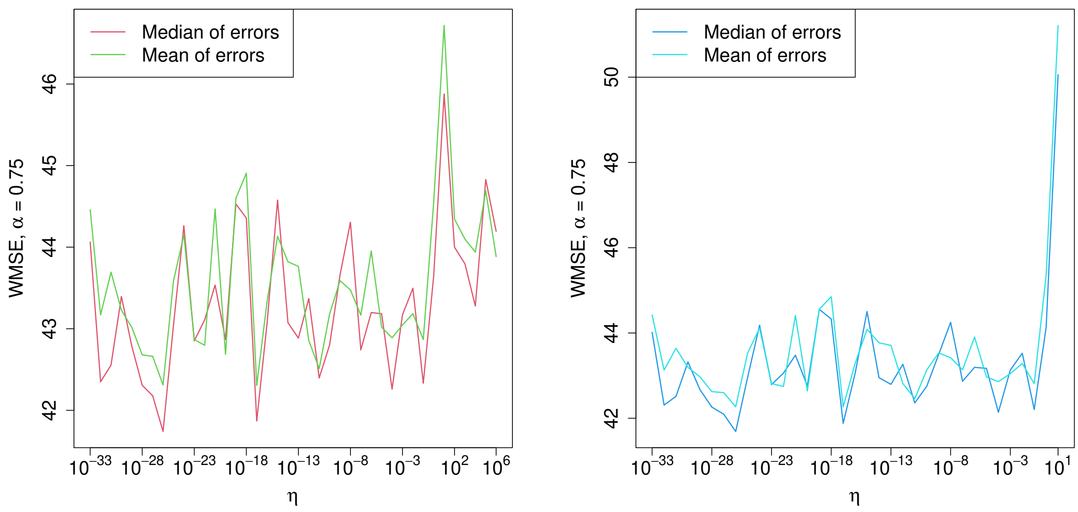
Figure A.1 Mean square error of coefficients for different values of η. The terms median and mean refer to the quantities obtained from 100 values of MSEoC, each one corresponding to one data sample.

As a result, we see our worries from the section 1.5 (about setting too low value of η and therefore destabilizing the whole process by using too small number in the denominator of (1.12)) were not realized. It seems there is no problem in setting η in levels as low as 10^{-32}. However, in the thesis we chose the value $\eta = 10^{-9}$, as we did not want to risk unnecessary numerical issues, which could have arisen in different model settings.

A.3 More simulations results

In this final attachment, tables with results from some other conducted simulations with some parameters modified are included.

The result that should have deserved more attention is the performance of LWS estimator with weight function LWS-1 in the "leverage" simulation design described under the table A.5. While it hugely lost to most of other considered estimators in terms of MSEoC and TMSE and WMSE for $\alpha \in \{0.7, 0.8\}$, for $\alpha = 0.9$ it outperformed them all significantly.



(a) WMSE for $\alpha = 0.75$ for dd-LWS. **(b)** WMSE for $\alpha = 0.75$ for dd-WLS.

Figure A.2 Weighted mean square error for $\alpha = 0.75$ for different values of η. The terms median and mean refer to the quantities obtained from 100 values of WMSE, each one corresponding to one data sample.

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	17.121	4.601	7.172	54.799	2.099	2.732	13.164
LWS-1	1.517	0.860	1.488	**51.429**	0.382	0.543	**11.589**
LWS-2	0.387	0.579	1.047	52.109	0.254	0.375	11.681
LWS-3	0.300	0.550	1.022	52.540	0.241	0.363	11.773
LWS-4	5.746	1.821	3.033	52.234	0.814	1.121	11.979
LWS-5	0.152	0.543	0.975	52.531	0.243	0.354	11.772
LTS	0.554	0.594	1.106	52.596	0.256	0.390	11.792
REWLS	**0.107**	0.556	**0.957**	52.524	0.249	**0.351**	11.772
dd-LWS	0.137	0.546	0.965	52.471	0.244	**0.351**	11.757
dd-WLS	0.154	0.543	0.971	52.486	0.242	0.352	11.761
MM	0.108	0.556	**0.957**	52.521	0.249	**0.351**	11.771
LMS	0.566	0.595	1.110	52.636	0.261	0.396	11.805
S	0.269	**0.540**	0.995	52.521	**0.239**	0.357	11.768

Table A.1 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design with sample size n set to 100.

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	5.939	1.768	2.928	50.390	0.797	1.088	11.545
LWS-1	0.541	0.661	1.163	**49.451**	0.294	0.423	**11.095**
LWS-2	0.117	0.570	1.018	49.696	0.252	0.367	11.130
LWS-3	0.096	0.562	1.010	49.836	**0.248**	0.364	11.160
LWS-4	2.024	0.949	1.641	49.646	0.424	0.600	11.202
LWS-5	0.047	0.559	0.996	49.822	0.249	0.361	11.157
LTS	0.218	0.583	1.049	49.870	0.256	0.376	11.171
REWLS	**0.030**	0.563	**0.990**	49.815	0.251	**0.360**	11.156
dd-LWS	0.038	0.560	0.992	49.802	0.249	**0.360**	11.152
dd-WLS	0.042	0.559	0.993	49.808	0.249	**0.360**	11.154
MM	0.031	0.563	**0.990**	49.814	0.251	**0.360**	11.156
LMS	0.186	0.583	1.051	49.863	0.259	0.379	11.172
S	0.128	**0.558**	1.002	49.828	**0.248**	0.362	11.158

Table A.2 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design with sample size n set to 300.

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	15.548	37.931	111.134	231.824	14.135	34.591	60.974
LWS-1	5.611	34.985	**108.645**	**231.432**	12.604	**33.273**	**60.212**
LWS-2	0.825	34.470	109.606	234.005	12.224	33.344	60.671
LWS-3	0.633	**34.394**	110.005	234.768	**12.189**	33.446	60.855
LWS-4	7.129	35.639	109.386	231.582	12.905	33.586	60.378
LWS-5	7.688	34.706	108.782	233.736	12.606	33.357	60.785
LTS	0.154	34.956	110.689	235.392	12.372	33.663	61.052
REWLS	**0.068**	34.949	110.656	235.370	12.369	33.654	61.046
dd-LWS	0.084	34.916	110.582	235.243	12.357	33.630	61.011
dd-WLS	0.080	34.927	110.610	235.294	12.361	33.639	61.024
MM	**0.068**	34.949	110.656	235.370	12.369	33.654	61.046
LMS	0.265	34.982	110.699	235.401	12.389	33.675	61.063
S	0.161	34.946	110.653	235.370	12.367	33.652	61.045

Table A.3 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design with randomly contaminated 40% of observations (instead of 20%).

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	323.123	50.966	79.723	159.877	23.540	30.583	47.224
LWS-1	1.080	3.086	17.395	**84.734**	1.247	5.179	**20.109**
LWS-2	0.409	2.889	17.401	85.370	1.147	5.142	20.218
LWS-3	**0.127**	2.809	17.424	85.610	**1.109**	5.134	20.261
LWS-4	94.938	17.715	36.453	106.767	7.893	12.806	28.066
LWS-5	0.343	2.850	**17.361**	85.533	1.137	**5.128**	20.252
LTS	0.205	2.861	17.488	85.674	1.128	5.156	20.282
REWLS	0.221	2.803	17.386	85.566	1.113	**5.128**	20.253
dd-LWS	0.132	2.812	17.402	85.538	1.110	**5.128**	20.244
dd-WLS	0.131	2.815	17.419	85.576	1.111	5.133	20.253
MM	0.224	**2.797**	17.396	85.577	1.111	5.131	20.256
LMS	0.468	2.926	17.537	85.734	1.163	5.184	20.311
S	0.206	2.821	17.447	85.638	1.113	5.141	20.269

Table A.4 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design with uncontaminated error terms ε_i following the t distribution with one degree of freedom. Selected 20% of observations came from the same contamination distribution as in the basic model design, i.e. they were uniformly distributed on $(-50, -10) \cup (10, 50)$.

	MSEoC	TMSE $\alpha = 0.7$	TMSE $\alpha = 0.8$	TMSE $\alpha = 0.9$	WMSE $\alpha = 0.7$	WMSE $\alpha = 0.8$	WMSE $\alpha = 0.9$
LS	7.374	1.619	2.644	47.401	0.728	0.985	10.831
LWS-1	3.801	2.252	3.675	**38.801**	1.007	1.366	**8.986**
LWS-2	0.488	0.730	1.296	47.134	0.323	0.468	10.583
LWS-3	0.143	0.561	1.010	50.171	0.248	0.364	11.239
LWS-4	3.019	1.200	2.038	46.164	0.536	0.749	10.459
LWS-5	0.091	0.558	0.989	50.162	0.249	0.360	11.238
LTS	0.299	0.591	1.066	50.156	0.258	0.381	11.240
REWLS	**0.056**	0.563	**0.980**	50.161	0.252	**0.358**	11.238
dd-LWS	0.070	0.561	0.987	49.882	0.250	0.359	11.174
dd-WLS	0.077	0.558	0.987	49.996	0.249	0.359	11.199
MM	**0.056**	0.563	**0.980**	50.158	0.252	**0.358**	11.237
LMS	0.297	0.589	1.062	50.195	0.261	0.383	11.250
S	0.161	**0.556**	0.998	50.174	**0.247**	0.361	11.239

Table A.5 Comparison of various estimators in terms of MSEoC, TMSE and WMSE for several values of α in the basic model design with the following so-called leverage contamination of regressors: for the regressors of selected 20% of observations with contaminated response, the independent variable $X_{i,2}$ followed normal distribution with the mean value 6 and the unit variance, i.e. $X_{i,2} \sim N(6, 1)$ in the 20% of cases with contaminated Y_i.