Machine Learning
https://doi.org/10.1007/s10994-021-05960-5

Transforming variables to central normality

Jakob Raymaekers[1] · Peter J. Rousseeuw[1]

Received: 16 May 2020 / Revised: 28 November 2020 / Accepted: 12 February 2021

Abstract

Many real data sets contain numerical features (variables) whose distribution is far from normal (Gaussian). Instead, their distribution is often skewed. In order to handle such data it is customary to preprocess the variables to make them more normal. The Box–Cox and Yeo–Johnson transformations are well-known tools for this. However, the standard maximum likelihood estimator of their transformation parameter is highly sensitive to outliers, and will often try to move outliers inward at the expense of the normality of the central part of the data. We propose a modification of these transformations as well as an estimator of the transformation parameter that is robust to outliers, so the transformed data can be approximately normal in the center and a few outliers may deviate from it. It compares favorably to existing techniques in an extensive simulation study and on real data.

Keywords Anomaly detection · Data preprocessing · Feature transformation · Outliers · Symmetrization

1 Introduction

In machine learning and statistics, some numerical data features may be very nonnormal (nonGaussian) and asymmetric (skewed) which often complicates the next steps of the analysis. Therefore it is customary to preprocess the data by transforming such features in order to bring them closer to normality, after which it typically becomes easier to fit a model or to make predictions. To be useful in practice, it must be possible to automate this preprocessing step.

In order to transform a positive variable to give it a more normal distribution one often resorts to a power transformation (see e.g. Tukey 1957). The most often used function is the Box–Cox (BC) power transform g_λ studied by Box and Cox (1964), given by

Editors: Tim Verdonck, Bart Baesens, María Óskarsdóttir and Seppe vanden Broucke.

✉ Peter J. Rousseeuw
peter@rousseeuw.net

Jakob Raymaekers
jakob.raymaekers@kuleuven.be

[1] KU Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium

Published online: 21 March 2021

$$g_\lambda(x) = \begin{cases} (x^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0. \end{cases} \tag{1}$$

Here x stands for the observed feature, which is transformed to $g_\lambda(x)$ using a parameter λ. A limitation of the family of BC transformations is that they are only applicable to positive data. To remedy this, Yeo and Johnson (2000) proposed an alternative family of transformations that can deal with both positive and negative data. These Yeo–Johnson (YJ) transformations h_λ are given by

$$h_\lambda(x) = \begin{cases} ((1+x)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \text{ and } x \geq 0 \\ \log(1+x) & \text{if } \lambda = 0 \text{ and } x \geq 0 \\ -((1-x)^{2-\lambda} - 1)/(2-\lambda) & \text{if } \lambda \neq 2 \text{ and } x < 0 \\ -\log(1-x) & \text{if } \lambda = 2 \text{ and } x < 0 \end{cases} \tag{2}$$

and are also characterized by a parameter λ. Figure 1 shows both of these transformations for a range of λ values. In both families $\lambda = 1$ yields a linear relation. Transformations with $\lambda < 1$ compress the right tail of the distribution while expanding the left tail, making them suitable for transforming right-skewed distributions towards symmetry. Similarly, transformations with $\lambda > 1$ are designed to make left-skewed distributions more symmetrical.

Estimating the parameter λ for the BC or YJ transformation is typically done using maximum likelihood, under the assumption that the transformed variable follows a normal distribution. However, it is well known that maximum likelihood estimation is very sensitive to outliers in the data, to the extent that a single outlier can have an arbitrarily large effect on the estimate. In the setting of transformation to normality, outliers can yield transformations for which the bulk of the transformed data follows a very skewed distribution, so no normality is attained at all. In situations with outliers one would prefer to make the non-outliers approximately normally distributed, while the outliers may stay outlying. So, our goal is to achieve *central normality*, where the transformed data look roughly normal in the center and a few outliers may deviate from it. Fitting such a transformation is not easy, because a point that is outlying in the original data may not be outlying in the transformed

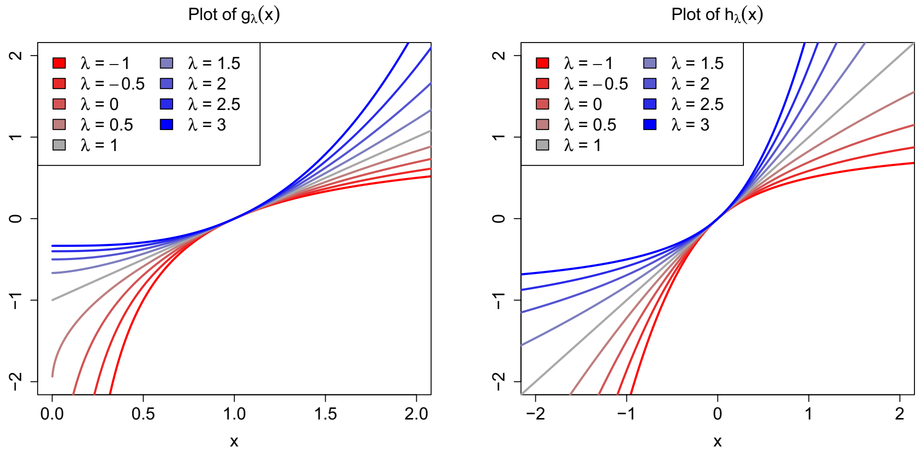

Fig. 1 The Box–Cox (left) and Yeo–Johnson (right) transformations for several parameters λ

data, and vice versa. The problem is that we do not know beforehand which points may turn out to be outliers in the optimally transformed data.

Some proposals exist in the literature to make the estimation of the parameter λ in BC more robust against outliers, mainly in the context of transforming the response variable in a regression (Carroll 1980; Marazzi et al. 2009; Riani 2008), but here we are not in that setting. For the YJ transformation very few robust methods are available. In (Van der Veeken 2010) a trimmed maximum likelihood approach was explored, in which the objective is a trimmed sum of log likelihoods in which the lowest terms are discarded. We will study this method in more detail later.

Note that both the BC and YJ transformations suffer from the complication that their range depends on the parameter λ. In particular, for the BC transformation we have

$$g_\lambda(\mathbb{R}_0^+) = \begin{cases} (-1/|\lambda|, \infty) & \text{if } \lambda > 0 \\ \mathbb{R} & \text{if } \lambda = 0 \\ (-\infty, 1/|\lambda|) & \text{if } \lambda < 0 \end{cases} \tag{3}$$

whereas for the YJ transformation we have

$$h_\lambda(\mathbb{R}) = \begin{cases} (-1/|\lambda - 2|, \infty) & \text{if } \lambda > 2 \\ \mathbb{R} & \text{if } 0 \le \lambda \le 2 \\ (-\infty, 1/|\lambda|) & \text{if } \lambda < 0. \end{cases} \tag{4}$$

So, for certain values of λ the range of the transformation is a half line. This is not without consequences. First, most well-known symmetric distributions are supported on the entire line, so a perfect match is impossible. More importantly, we argue that this can make outlier detection more difficult. Consider for instance the BC transformation with $\lambda = -1$ which has the range $g_{-1}(\mathbb{R}_0^+) = (-\infty, 1)$. Suppose we transform a data set $(x_1, \dots, x_n)$ to $(g_{-1}(x_1), \dots, g_{-1}(x_n))$. If we let $x_n \to \infty$ making it an extremely clear outlier in the original space, then $g_{-1}(x_n) \to 1$ in the transformed space. So a transformed outlier can be much closer to the bulk of the transformed data than the original outlier was in the original data. This is undesirable, since the outlier will be much harder to detect this way. This effect is magnified if λ is estimated by maximum likelihood, since this estimator will try to accommodate all observations, including the outliers.

We illustrate this point using the TopGear dataset (Alfons 2019) which contains information on 297 cars, scraped from the website of the British television show Top Gear. We fit a Box–Cox transformation to the variable `miles per gallon` (`MPG`) which is strictly positive. The left panel of Fig. 2 shows the normal QQ-plot of the `MPG` variable before transformation. (That is, the horizontal axis contains as many quantiles from the standard normal distribution as there are sorted data values on the vertical axis.) In this plot the majority of the observations seem to roughly follow a normal distribution, that is, many points in the QQ-plot lie close to a straight line. There are also three far outliers at the top, which correspond to the Chevrolet Volt and Vauxhall Ampera (both with 235 MPG) and the BMW i3 (with 470 MPG). These cars are unusual because they derive most of their power from a plug-in electric battery, whereas the majority of the cars in the data set are gas-powered. The right panel of Fig. 2 shows the Box–Cox transformed data using the maximum likelihood (ML) estimate $\hat{\lambda} = -0.11$, indicating that the BC transformation is fairly close to the log transform. We see that this transformation does not improve the normality of the MPG variable. Instead it tries to bring the three outliers into the fold, at

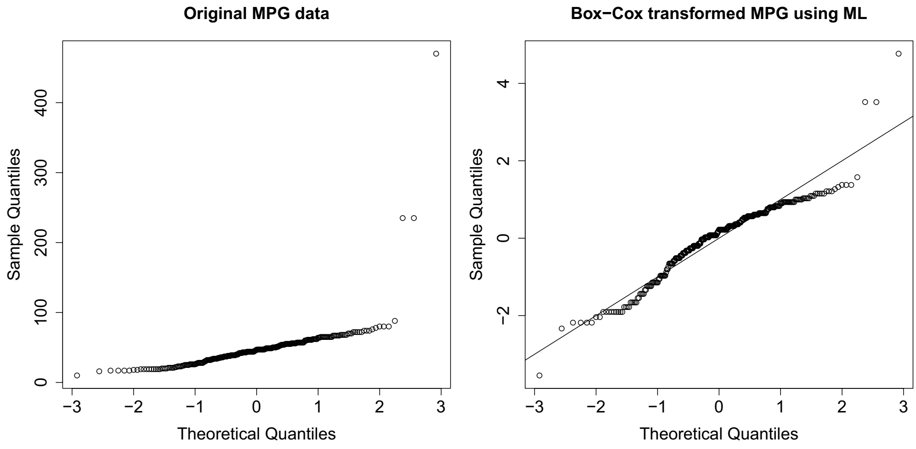

Fig. 2 Normal QQ-plot of the variable `MPG` in the Top Gear dataset (left) and the Box–Cox transformed variable using the maximum likelihood estimate of λ (right). The ML estimate is heavily affected by the three outliers at the top, causing it to create skewness in the central part of the transformed data

the expense of causing skewness in the central part of the transformed data and creating an artificial outlier at the bottom.

The variable `Weight` shown in Fig. 3 illustrates a different effect. The original variable has one extreme and 4 intermediate outliers at the bottom. The extreme outlier is the Peugeot 107, whose weight was erroneously listed as 210 kg, and the next outlier is the tiny Renault Twizy (410 kg). Unlike the MPG variable the central part of these data is not very normal, as those points in the QQ-plot do not line up so well. A transform that would make the central part more straight would expose the outliers at the bottom more. But instead the ML estimate is $\hat{\lambda} = 0.83$ hence close to $\lambda = 1$ which would correspond to not transforming

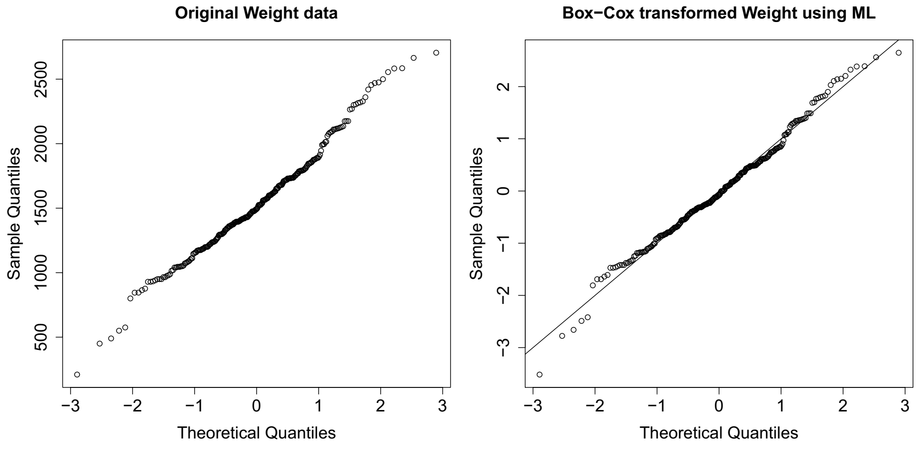

Fig. 3 Normal QQ-plot of the variable `Weight` in the Top Gear dataset (left) and the transformed variable using the ML estimate of λ (right). The transformation does not make the five outliers at the bottom stand out

the variable at all. Whereas the `MPG` variable should not be transformed much but is, the `Weight` variable should be transformed but almost isn't.

In Sect. 2 we propose a new robust estimator for the parameter λ, and compare its sensitivity curve to those of other methods. Section 3 presents a simulation to study the performance of several estimators on clean and contaminated data. Section 4 illustrates the proposed method on real data examples, and Sect. 6 concludes.

2 Methodology

2.1 Fitting a transformation by minimizing a robust criterion

The most popular way of estimating the λ of the BC and YJ transformations is to use maximum likelihood (ML) under the assumption that the transformed variable follows a normal distribution, as will briefly be summarized in Sect. 2.3. However, it is well known that ML estimation is very sensitive to outliers in the data and other deviations from the assumed model. We therefore propose a different way of estimating the transformation parameter of a transformation.

Consider an ordered sample of univariate observations $X = (x_{(1)}, \dots, x_{(n)})$. Suppose we want to estimate the parameter λ of a nonlinear function g_λ such that $g_\lambda(x_{(1)}), \dots, g_\lambda(x_{(n)})$ come close to quantiles from the standard normal cumulative distribution function $\boldsymbol{\Phi}$. We propose to estimate λ as:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{g_\lambda(x_{(i)}) - \hat{\mu}_{\mathrm{M}}}{\hat{\sigma}_{\mathrm{M}}} - \boldsymbol{\Phi}^{-1}(p_i)\right) . \tag{5}$$

Here $\hat{\mu}_{\mathrm{M}}$ is the Huber M-estimate of location of the $g_\lambda(x_{(i)})$, and $\hat{\sigma}_{\mathrm{M}}$ is their Huber M-estimate of scale. Both are standard robust univariate estimators (see Huber 1981). The $p_i = (i - 1/3)/(n + 1/3)$ are the usual equispaced probabilities that also yield the quantiles in the QQ-plot (see, e.g., page 225 in Hoaglin et al. (1983)). The function ρ needs to be positive, even and continuously differentiable. In least squares methods $\rho(t) = t^2$, but in our situation there can be large absolute residuals $|\frac{g_\lambda(x_{(i)})-\hat{\mu}}{\hat{\sigma}} - \boldsymbol{\Phi}^{-1}(p_i)|$ caused by outlying values of $g_\lambda(x_{(i)})$. In order to obtain a robust method we need a bounded ρ function. We propose to use the well-known Tukey bisquare ρ-function given by

$$\rho_{\mathrm{bw}}(x) = \begin{cases} 1 - (1 - (x/c)^2)^3 & \text{if } |x| \le c \\ 1 & \text{if } |x| > c. \end{cases} \tag{6}$$

The constant c is a tuning parameter, which we set to 0.5 by default here. See section A of the supplementary material for a motivation of this choice.

To calculate $\hat{\lambda}$ numerically, we use the `R` function `optimize()` which relies on a combination of golden section search and successive parabolic interpolation to minimize the objective of (5).

2.2 Rectified Box–Cox and Yeo–Johnson transformations

In this section we propose a modification of the classical BC and YJ transformations, called the *rectified* BC and YJ transformations. They make a continuously differentiable switch to a linear transformation in the tails of the BC and YJ functions. The purpose of these modified transformations is to remedy two issues. First, the range of the classical BC and YJ transformations depends on λ and is often only a half line. And second, as argued in the introduction, the classical transformations often push outliers closer to the majority of the data, which makes the outliers harder to detect. Instead the range of the proposed modified transformations is always the entire real line, and it becomes less likely that outliers are masked by the transformation.

For $\lambda < 1$, the BC transformation is designed to make right-skewed distributions more symmetrical, and is bounded from above. In this case we define the rectified BC transformation as follows. Consider an upper constant $C_u > 1$. The rectified BC transformation $\mathring{g}_\lambda$ is defined as

$$\mathring{g}_\lambda(x) = \begin{cases} g_\lambda(x) & \text{if } x \leq C_u \\ g_\lambda(C_u) + (x - C_u)g'_\lambda(C_u) & \text{if } x > C_u. \end{cases} \tag{7}$$

Similarly, for $\lambda > 1$ and a positive lower constant $C_\ell < 1$ we put

$$\mathring{g}_\lambda(x) = \begin{cases} g_\lambda(C_\ell) + (x - C_\ell)g'_\lambda(C_\ell) & \text{if } x < C_\ell \\ g_\lambda(x) & \text{if } x \geq C_\ell. \end{cases} \tag{8}$$

For the YJ transformation we construct rectified counterparts in a similar fashion. For $\lambda < 1$ and a value $C_u > 0$ we define the rectified YJ transformation $\mathring{h}_\lambda(x)$ as in (7) with g_λ replaced by h_λ:

$$\mathring{h}_\lambda(x) = \begin{cases} h_\lambda(x) & \text{if } x \leq C_u \\ h_\lambda(C_u) + (x - C_u)h'_\lambda(C_u) & \text{if } x > C_u. \end{cases} \tag{9}$$

Analogously, for $\lambda > 1$ and $C_\ell < 0$ we define $\mathring{h}_\lambda(x)$ as in (8):

$$\mathring{h}_\lambda(x) = \begin{cases} h_\lambda(C_\ell) + (x - C_\ell)h'_\lambda(C_\ell) & \text{if } x < C_\ell \\ h_\lambda(x) & \text{if } x \geq C_\ell. \end{cases} \tag{10}$$

Figure 4 shows such rectified BC and YJ transformations.

What are good choices of C_ℓ and C_u? Since the original data is often asymmetric, we cannot just use a center (like the median) plus or minus a fixed number of (robust) standard deviations. Instead we set C_ℓ equal to the first quartile of the original data, and for C_u we take the third quartile. Other choices could be used, but more extreme quantiles would yield a higher sensitivity to outliers.

2.3 Reweighted maximum likelihood

We now describe a reweighting scheme to increase the accuracy of the estimated $\hat{\lambda}$ while preserving its robustness. For a data set $x_1, \ldots, x_n$ the classical maximum likelihood estimator for the Yeo–Johnson transformation parameter λ is given by the λ which maximizes the normal loglikelihood. After removing constant terms this can be written as:

$$\hat{\lambda}^{\text{YJ}}_{\text{ML}} = \underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^{n} -\frac{1}{2}\log(\hat{\sigma}^2_{\text{ML},\lambda}) + (\lambda - 1)\operatorname{sign}(x_i)\log(|x_i| + 1) \tag{11}$$

where $\hat{\sigma}^2_{\text{ML},\lambda}$ is the maximum likelihood scale of the transformed data given by

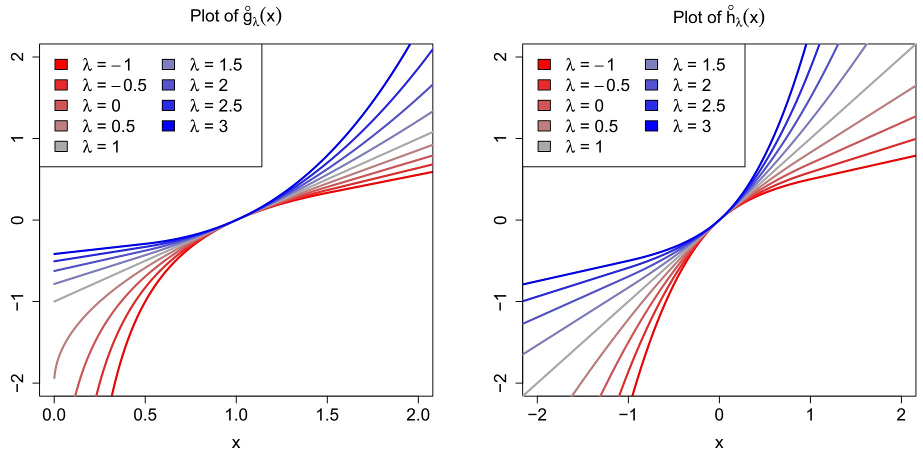

Fig. 4 The rectified Box–Cox (left) and Yeo–Johnson (right) transformations for a range of parameters λ. They look quite similar to the original transformations in Fig. 1 but contract less on the right when $\lambda < 1$, and contract less on the left when $\lambda > 1$

$$\hat{\sigma}^2_{\mathrm{ML},\lambda} = \frac{1}{n}\sum_{i=1}^{n}(h_\lambda(x_i) - \hat{\mu}_{\mathrm{ML},\lambda})^2 \quad \text{where} \quad \hat{\mu}_{\mathrm{ML},\lambda} = \frac{1}{n}\sum_{i=1}^{n} h_\lambda(x_i)\,. \tag{12}$$

The last term in (11) comes from the derivative of the YJ transformation. Criterion (11) is sensitive to outliers since it depends on a classical variance and the unbounded term $\log(1 + |x_i|)$. This can be remedied by using weights. Given a set of weights $W = (w_1, \dots, w_n)$ we define a weighted maximum likelihood (WML) estimator by

$$\hat{\lambda}^{\mathrm{YJ}}_{\mathrm{WML}} = \underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^{n} w_i\left[-\frac{1}{2}\log(\hat{\sigma}^2_{\mathrm{W},\lambda}) + (\lambda - 1)\,\mathrm{sign}\,(x_i)\log(1 + |x_i|)\right] \tag{13}$$

where $\hat{\sigma}^2_{\mathrm{W},\lambda}$ now denotes the weighted variance of the transformed data:

$$\hat{\sigma}^2_{\mathrm{W},\lambda} = \frac{\sum_{i=1}^{n} w_i(h_\lambda(x_i) - \hat{\mu}_{\mathrm{W},\lambda})^2}{\sum_{i=1}^{n} w_i} \quad \text{where} \quad \hat{\mu}_{\mathrm{W},\lambda} = \frac{\sum_{i=1}^{n} w_i\, h_\lambda(x_i)}{\sum_{i=1}^{n} w_i}\,. \tag{14}$$

If the weights appropriately downweight the outliers in the data, the WML criterion yields a more robust estimate of the transformation parameter.

For the BC transform the reasoning is analogous, the only change being the final term that comes from the derivative of the BC transform. This yields

$$\hat{\lambda}^{\mathrm{BC}}_{\mathrm{WML}} = \underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^{n} w_i\left[-\frac{1}{2}\log(\hat{\sigma}^2_{\mathrm{W},\lambda}) + (\lambda - 1)\log(x_i)\right]\,. \tag{15}$$

In general, finding robust data weights is not an easy task. The problem is that the observed data $X = (x_1, \dots, x_n)$ can have a (very) skewed distribution and there is no straightforward way to know which points will be outliers in the transformed data when λ is unknown. But suppose that we have a rough initial estimate λ_0 of λ. We can then transform the data with λ_0 yielding $h_{\lambda_0}(X) = (h_{\lambda_0}(x_1), \dots, h_{\lambda_0}(x_n))$, which should be a lot more symmetric than the original data. We can therefore compute weights on $h_{\lambda_0}(X)$ using a classical weight function. Here we will use the "hard rejection rule" given by

$$w_i = \begin{cases} 1 & \text{if } |h_{\lambda_0}(x_i) - \hat{\mu}| \leqslant \Phi^{-1}(0.995)\,\hat{\sigma} \\ 0 & \text{if } |h_{\lambda_0}(x_i) - \hat{\mu}| > \Phi^{-1}(0.995)\,\hat{\sigma} \end{cases} \tag{16}$$

with $\hat{\mu}$ and $\hat{\sigma}$ as in (5). With these weights we can compute a reweighted estimate $\hat{\lambda}_1$ by the WML estimator in (13). Of course, the robustness of the reweighted estimator will depend strongly on the robustness of the initial estimate λ_0.

Note that the reweighting step can be iterated, yielding a multistep weighted ML estimator. In simulation studies we found that more than 2 reweighting steps provided no further improvement in terms of accuracy (these results are not shown for brevity). We will always use two reweighting steps from here onward.

2.4 The proposed method

Combining the above ideas, our proposed reweighted maximum likelihood (RewML) method consists of the following steps:

- *Step 1* Compute the initial estimate λ_0 by maximizing the robust criterion (5). When fitting a Box–Cox transformation, plug in the rectified function $\mathring{g}_\lambda$. When fitting a Yeo–Johnson transformation, use the rectified function $\mathring{h}_\lambda$. Note that the rectified transforms are only used in this first step.
- *Step 2* Using λ_0 as starting value, compute the reweighted ML estimate from (15) when fitting the unrectified Box–Cox transform g_λ, and from (13) when fitting the unrectified Yeo–Johnson transform h_λ.
- *Step 3* Repeat step 2 once and stop.

2.5 Other estimators of λ

We will compare our proposal with two existing robust methods.

The first is the robustified ML estimator proposed by Carroll (1980). The idea was to replace the variance $\hat{\sigma}^2_{\mathrm{ML},\lambda}$ in the ML formula (11) by a robust variance estimate of the transformed data. Carroll's method was proposed for the BC transformation, but the idea can be extended naturally to the estimation of the parameter of the YJ transformation. The estimator is then given by

$$\hat{\lambda}_{\text{Carroll}} = \underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^{n} -\frac{1}{2}\log(\hat{\sigma}^2_{\mathrm{M},\lambda}) + (\lambda - 1)\,\mathrm{sign}\,(x_i)\log(1 + |x_i|) \tag{17}$$

where $\hat{\sigma}_{\mathrm{M},\lambda}$ denotes the usual Huber M-estimate of scale (Huber 1981) of the transformed data set $(h_\lambda(x_1), \ldots, h_\lambda(x_n))$.

The second method is the maximum trimmed likelihood (MTL) estimator of Van der Veeken (2010). Given a data set of size n, and a fixed number h that has to satisfy $\lceil \frac{n}{2} \rceil < h < n$, this method looks for the parameter $\hat{\lambda}$ which produces a subset of h consecutive observations which maximize the ML criterion (11).

2.6 Sensitivity curves

In order to assess robustness against an outlier, stylized sensitivity curves were introduced on page 96 of Andrews et al. (1972). For a given estimator T and a cumulative distribution function F they are constructed as follows:

1. Generate a stylized pseudo data set X^0 of size $n-1$:
$$X^0 = (x_1, \dots, x_{n-1}) = \left(F^{-1}(p_1), \dots, F^{-1}(p_{n-1})\right)$$
where the p_i for $i = 1, \dots, n-1$ are equispaced probabilities that are symmetric about 1/2. We can for instance use $p_i = i/n$.
2. Add to this stylized data set a variable point z to obtain
$$X_z = (x_1, \dots, x_{n-1}, z).$$
3. Calculate the sensitivity curve in z by
$$\mathrm{SC}_n(z) := n\left(T(X_z) - T(X^0)\right)$$
where z ranges over a grid chosen by the user. The purpose of the factor n is to put sensitivity curves with different values of n on a similar scale.

The top panel of Fig. 5 shows the sensitivity curves for several estimators of the parameter λ of the YJ transformation. We chose $F = \Phi$ so the true transformation parameter λ is 1, and $n = 100$. The maximum likelihood estimator ML of (11) has an unbounded sensitivity curve, which is undesirable as it means that a single outlier can move $\hat{\lambda}$ arbitrarily far away. The estimator of Carroll (17) has the same property, but is less affected in the sense that for a high $|z|$ the value of $|\ \mathrm{SC}_n(z)|$ is smaller than for the ML estimator. The RewML estimator that we proposed in Sect. 2.4 has a sensitivity curve that lies close to that of the ML in the central region of z, and becomes exactly zero for more extreme values of $|z|$. Such a sensitivity curve is called *redescending*, meaning that it goes back to zero. Therefore a far outlier has little effect on the resulting estimate. We also show MTL95, the trimmed likelihood estimator described in Sect. 2.5 with $h/n = 95\%$. Its sensitivity curve is also redescending, but in the central region it is more erratic with several jumps.

The lower panel of Fig. 5 shows the sensitivity curves for the Box–Cox transformation when the true parameter is $\lambda = 0$, i.e. the clean data follows a lognormal distribution F. We now put $\log(z)$ on the horizontal axis, since this makes the plot more comparable to that for Yeo–Johnson in the top panel. Also here the ML and Carroll's estimator have an unbounded sensitivity curve. Our RewML estimator has a redescending SC which again behaves similarly to the classical ML for small $|\log(z)|$, whereas the sensitivity to an extreme outlier is zero. The maximal trimmed likelihood estimator MTL95 has large jumps reaching values over 40 in the central region. Those peaks are not shown because the other curves would be hard to distinguish on that scale.

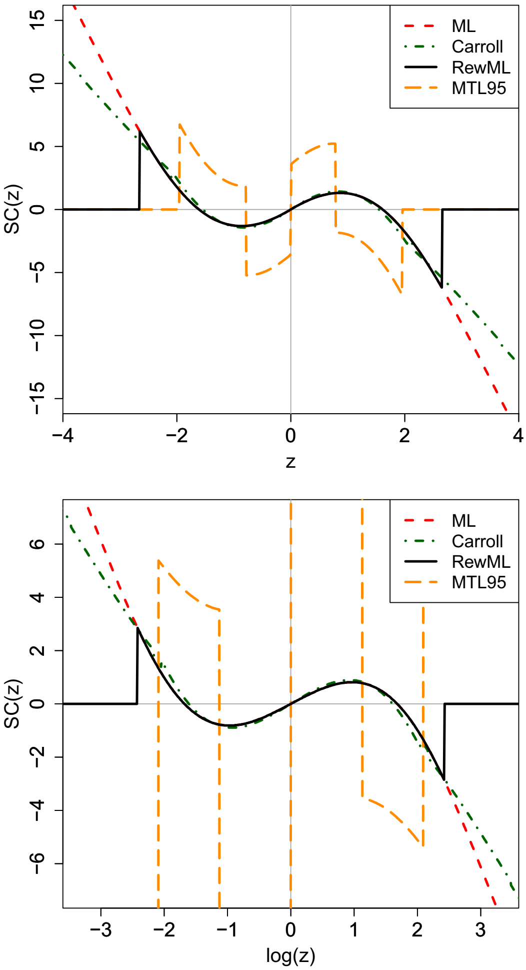

Fig. 5 Sensitivity curves of estimators of the parameter λ in the Yeo–Johnson (top) and Box–Cox (bottom) transformations, with sample size $n = 100$

3 Simulation

3.1 Compared methods

For the Box–Cox as well as the Yeo–Johnson transformations we perform a simulation study to compare the performance of several methods, including our proposal. The estimators under consideration are:

1. *ML* the classical maximum likelihood estimator given by (11), or by (15) with all $w_i = 1$.
2. *Carroll* the robustified maximum likelihood estimator of Carroll (1980) given by (17).
3. *MTL* the maximum trimmed likelihood estimator of Van der Veeken (2010). The notation MTL90 stands for the version with $h/n = 90\%$.
4. *RewML* the proposed reweighted maximum likelihood estimator described in Sect. 2.4.

5. *RewMLnr* a variation on RewML in which the first step of Sect. 2.4 applies (5) to the original Box–Cox or Yeo–Johnson transform instead of their rectified versions. This is not intended as a proposal, but included in order to show the advantage of rectification.

3.2 Data generation

We generate clean data sets as well as data with a fraction ε of outliers. The clean data are produced by generating a sample of size n from the standard normal distribution, after which the inverse of the BC or YJ transformation with a given λ is applied. For contaminated data we replace a percentage ε of the standard normal data by outliers at a fixed position before the inverse transformation is applied. For each such combination of λ and ε we generate $m = 100$ data sets.

To be more precise, the percentage ε of contaminated points takes on the values 0, 0.05, 0.1, and 0.15, where $\varepsilon = 0$ corresponds to uncontaminated data. For the YJ transformation we take the true transformation parameter λ equal to 0.5, 1.0, or 1.5. We chose these values because for λ between 0 and 2 the range of YJ given by (4) is the entire real line, so the inverse of YJ is defined for all real numbers. For the BC transformation we take $\lambda = 0$ for which the range (3) is also the real line. For a given combination of ε and λ the steps of the data generation are:

1. Generate a sample $Y = (y_1, \dots, y_n)$ from the standard normal distribution. Let $k > 0$ be a positive parameter. Then replace a fraction ε of the points in Y by k itself when $\lambda \leqslant 1$, and by $-k$ when $\lambda > 1$.
2. Apply the inverse BC transformation to Y, yielding the data set X given by$X = (g_\lambda^{-1}(y_1), \dots, g_\lambda^{-1}(y_n))$. For YJ we put $X = (h_\lambda^{-1}(y_1), \dots, h_\lambda^{-1}(y_n))$.
3. Estimate λ from X using the methods described in Sect. 3.1.

The parameter k characterizing the position of the contamination is an integer that we let range from 0 to 10.

We then estimate the bias and mean squared error (MSE) of each method by

$$\begin{aligned} \text{bias} &:= \text{ave}\,{}_{j=1}^{n}(\hat{\lambda}_j - \lambda) \\ \text{MSE} &:= \text{ave}\,{}_{j=1}^{n}(\hat{\lambda}_j - \lambda)^2 \end{aligned}$$

where $j = 1, \dots, m$ ranges over the generated data sets.

3.3 Results for the Yeo–Johnson transformation

We first consider the effect of an increasing percentage of contamination on the different estimators. In this setting we fix the position of the contamination by setting $k = 10$. (The results for $k = 6$ are qualitatively similar, as can be seen in section B of the supplementary material.) Figure 6 shows the bias and MSE of the estimators for an increasing contamination percentage ε on the horizontal axis. The results in the top row are for data generated with $\lambda = 0.5$, whereas the middle row was generated with $\lambda = 1$ and the bottom row with $\lambda = 1.5$. In all rows the classical ML and the Carroll estimator have the largest bias and MSE, meaning they react strongly to far outliers, as suggested by their unbounded sensitivity curves in Fig. 5. In contrast with this both

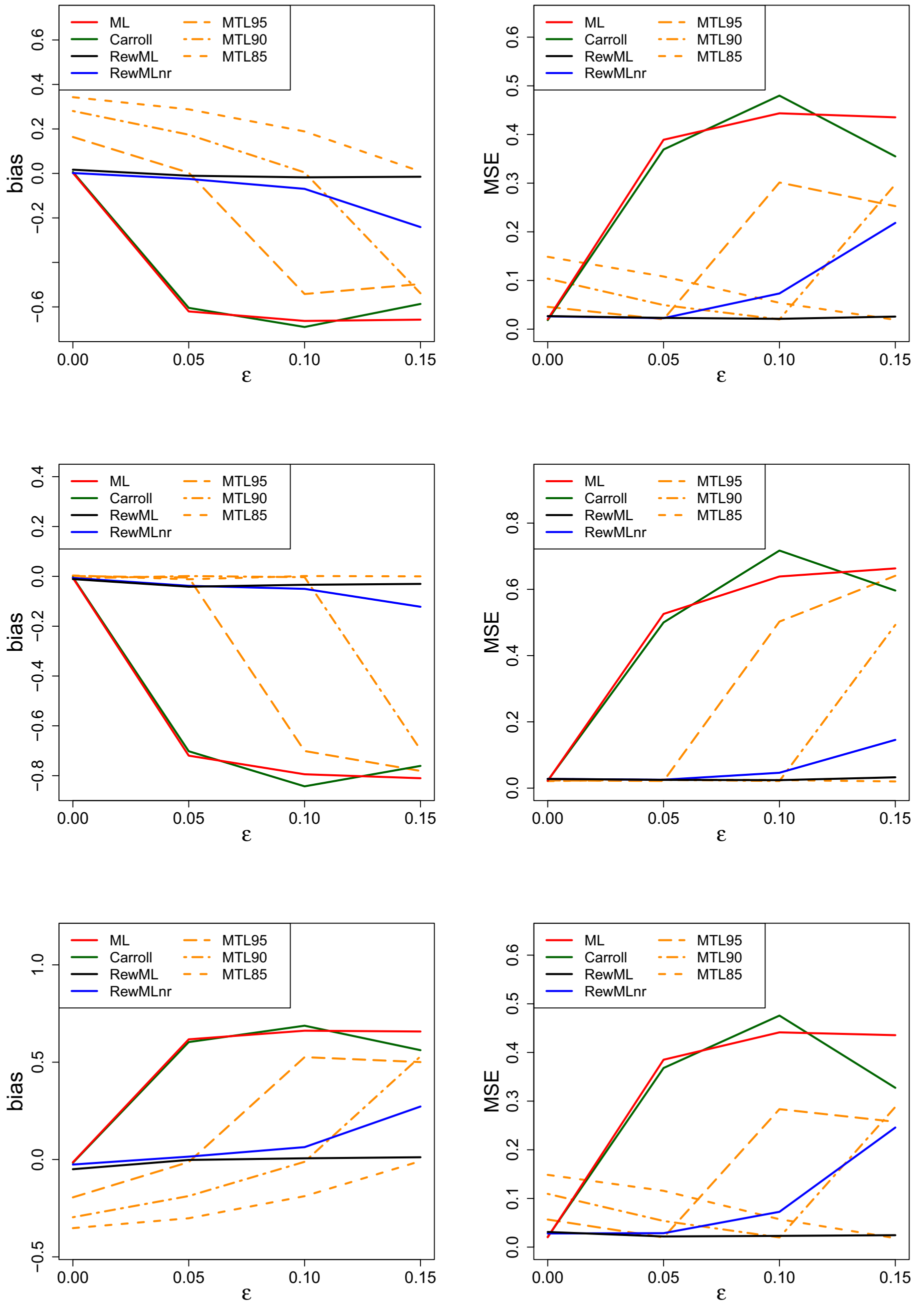

Fig. 6 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Yeo–Johnson transformation as a function of the percentage ε of outliers, when the location of the outliers is determined by setting $k = 10$. The true parameter λ used to generate the data is 0.5 in the top row, 1.0 in the middle row, and 1.5 in the bottom row

RewML and RewMLnr perform much better as their bias and MSE are closer to zero. Up to about 5% of outliers their curves are almost indistinguishable, but beyond that RewML outperforms RewMLnr by a widening margin. This indicates that using the rectified YJ transform in the first step of the estimator (see Sect. 2.4) is more robust than using the plain YJ in that step, even though the goal of the entire 3-step procedure RewML is to estimate the λ of the plain YJ transform.

In the same Fig. 6 we see the behavior of the maximum trimmed likelihood estimators MTL95, MTL90 and MTL85. In the middle row λ is 1, and we see that MTL95, which fits 95% of the data, performs well when there are up to 5% of outliers and performs poorly when there are over 5% of outliers. Analogously MTL90 performs well as long as there are at most 10% of outliers, and so on. This is the intended behavior. But note that for $\lambda \neq 1$ these estimators also have a substantial bias when the fraction of outliers is *below* what they aim for, as can be seen in the top and bottom panels of Fig. 6. For instance MTL85 is biased when ε is under 15%, even for $\varepsilon = 0\%$ when there are no outliers at all. So overall the MTL estimators only performed well when the percentage of trimming was equal to 1 minus the percentage of outliers in the data. Since the true percentage of outliers is almost never known in advance, it is not recommended to use the MTL method for variable transformation.

Let us now investigate what happens if we keep the percentage of outliers fixed, say at $\varepsilon = 10\%$, but vary the position of the contamination by letting $k = 0, 1, \ldots, 10$. Figure 7 shows the resulting bias and MSE, with again $\lambda = 0.5$ in the top row, $\lambda = 1$ in the middle row, and $\lambda = 1.5$ in the bottom row. For $k = 0$ and $k = 1$ the ML, Carroll, RewML and RewMLnr methods give similar results, since the contamination is close to the center so it cannot be considered outlying. But as k increases the classical ML and the Carroll estimator become heavily affected by the outliers. On the other hand RewML and RewMLnr perform much better, and again RewML outperforms RewMLnr. Note that the bias of RewML moves toward zero when k is large enough. We already noted this redescending behavior in its sensitivity curve in Fig. 5.

The behavior of the maximum trimmed likelihood methods depends on the value of λ used to generate the data. First focus on the middle row of Fig. 7 where $\lambda = 1$ so the clean data is generated from the standard normal distribution. In that situation both MTL90 and MTL85 behave well, whereas MTL95 can only withstand 5% of outliers and not the 10% generated here. One might expect the MTL method to work well as long as its h excludes at least the number of outliers in the data. But in fact MTL85 does not behave so well when λ differs from 1, as seen in the top and bottom panels of Fig. 7, where the bias remains substantial even though the method expects 15% of outliers and there are only 10% of them. As in Fig. 6 this suggests that one needs to know the actual percentage of outliers in the data in order to select the appropriate h for the MTL method, but that percentage is typically unknown.

3.4 Results for the Box–Cox transformation

When simulating data to apply the Box–Cox transformation to, the most natural choice of λ is zero since this is the only value for which the range of BC is the entire real line. Therefore we can carry out the inverse BC transformation on any data set generated from a normal distribution, so the clean data follows a log-normal distribution. The top panel of Fig. 8 shows the bias and MSE for 10% of outliers with $k = 1, \ldots, 10$. We see that the classical ML and the estimator of Carroll are sensitive to outliers when k grows. Our reweighted

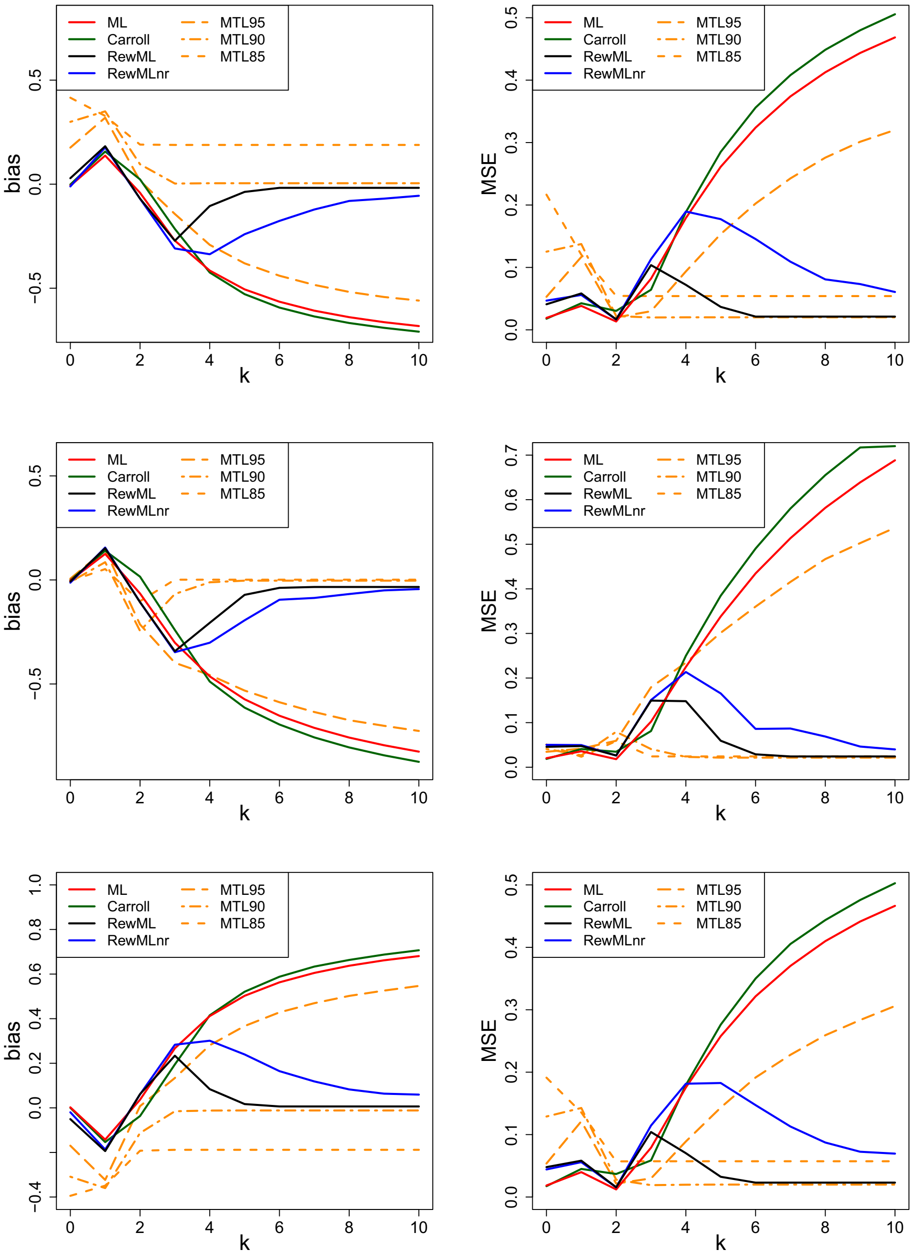

Fig. 7 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Yeo–Johnson transformation as a function of k which determines how far the outliers are. Here the percentage of outliers is fixed at 10%. The true parameter λ used to generate the data is 0.5 in the top row, 1.0 in the middle row, and 1.5 in the bottom row

method RewML performs much better. The RewMLnr method only differs from RewML in that it uses the non-rectified BC transform in the first step, and does not do as well since its bias goes back to zero at a slower rate.

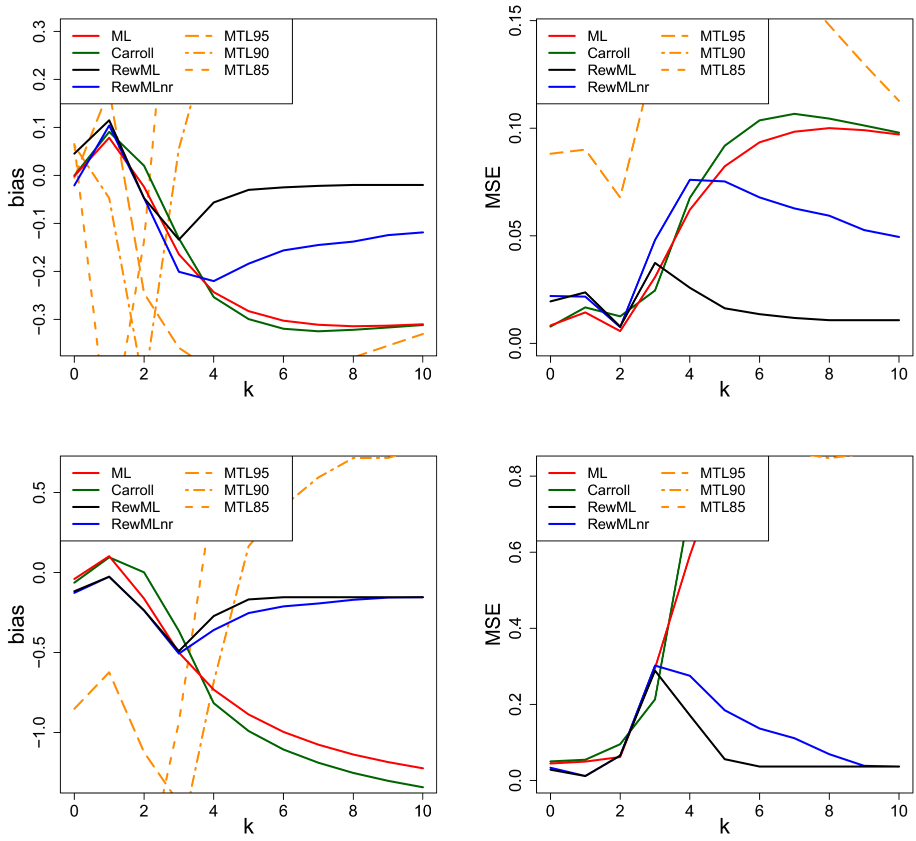

Fig. 8 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Box–Cox transformation as a function of k which determines how far the outliers are. Here the percentage of outliers is fixed at 10%. The true λ used to generate the data is 0 in the top row and 1 in the bottom row

The MTL estimators perform poorly here. The MTL95 version trims only 5% of the data points so it cannot discard the 10% of outliers, leading to a behavior resembling that of the classical ML. But also MTL90 and MTL85, which trim enough data points, obtain a large bias which goes in the opposite direction, accompanied by a high MSE. The MSE of MTL90 and MTL85 lie entirely above the plot area.

Finally, we consider a scenario with $\lambda = 1$. In that case the range of the Box–Cox transformation given by (3) is only $(-1, +\infty)$ so the transformed data cannot be normally distributed (which is already an issue for the justification of the classical maximum likelihood method). But the transformed data can have a truncated normal distribution. In this special setting we generated data from the normal distribution with mean 1 and standard deviation 1/3, and then truncated it to [0.01, 1.99] (keeping $n = 100$ points), so the clean data are strictly positive and have a symmetric distribution around 1. In the bottom panel of Fig. 8 we see that the ML and Carroll estimators are not robust in this simulation setting. The trimmed likelihood estimators also fail to deliver reasonable results, with curves that often fall outside the plot area. On the other hand RewML still performs well, and again does better than RewMLnr.

The simulation results for a fixed outlier position at $k = 6$ or $k = 10$ with contamination levels ε from 0% to 15% can be found in section B of the supplementary material, and are qualitatively similar to those for the YJ transform.

4 Empirical examples

4.1 Car data

Let us revisit the positive variable `MPG` from the TopGear data shown in the left panel of Fig. 2. The majority of these data are already roughly normal, and three far outliers at the top deviate from this pattern. Before applying a Box–Cox transformation we first scale the variable so its median becomes 1. This makes the result invariant to the unit of measurement, whether it is miles per gallon or, say, kilometers per liter. The maximum likelihood estimator for Box–Cox tries to bring in the outliers and yields $\hat{\lambda} = -0.11$, which is close to $\lambda = 0$ corresponding to a logarithmic transformation. The resulting transformed data in the left panel of Fig. 9 are quite skewed in the central part, so not normal at all, which defeats the purpose of the transformation. This result is in sharp contrast with our reweighted maximum likelihood (RewML) method which estimates the transformation parameter as $\hat{\lambda} = 0.84$. The resulting transformed data in the right panel does achieve central normality.

The variable `Weight` in the left panel of Fig. 3 is not very normal in its center and has some outliers at the bottom. The classical ML estimate is $\hat{\lambda} = 0.83$, close to $\lambda = 1$ which would not transform the data at all, as we can see in the resulting left panel of Fig. 10. In contrast, our RewML estimator obtains $\hat{\lambda} = 0.09$ which substantially transforms the data, yielding the right panel of Fig. 10. There the central part of the data is very close to normal, and the outliers at the bottom now stand out more, as they should.

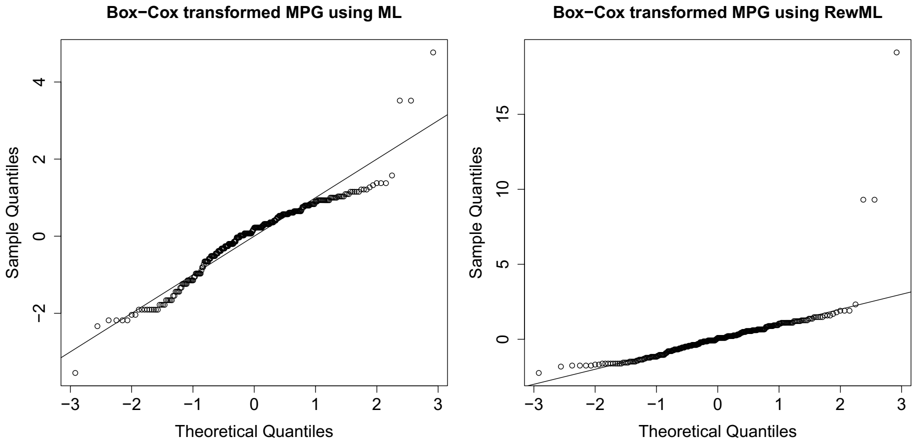

Fig. 9 Normal QQ-plot of the Box–Cox transformed variable MPG using the ML estimate of λ (left) and using the RewML estimate (right). The ML is strongly affected by the 3 outliers at the top, thereby transforming the central data away from normality. The RewML method achieves central normality and makes the outliers stand out more

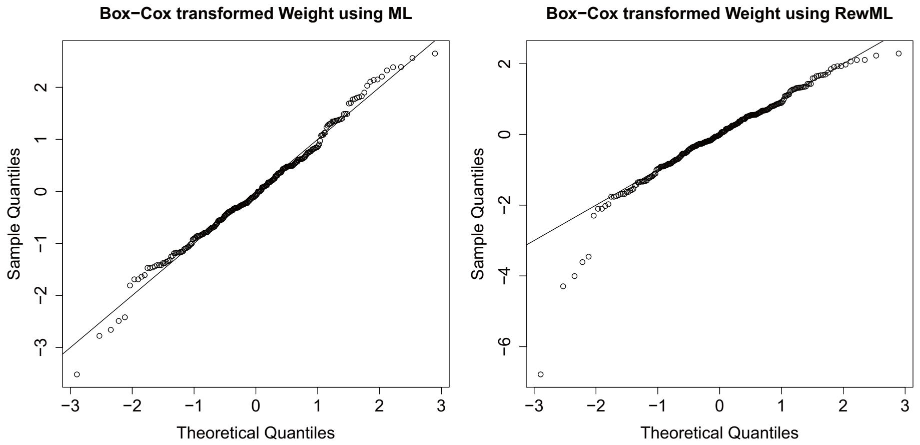

Fig. 10 Normal QQ-plot of the Box–Cox transformed variable `Weight` using the ML estimate of λ (left) and using the RewML estimate (right). The ML masks the five outliers at the bottom, whereas RewML accentuates them and achieves central normality

4.2 Glass data

For a multivariate example we turn to the `glass` data (Lemberge et al. 2000; Rousseeuw and Van den Bossche 2018) from chemometrics, which has become something of a benchmark. The data consists of $n = 180$ archeological glass samples, which were analyzed by spectroscopy. Our variables are the intensities measured at 500 wavelengths. Many of these variables do not look normally distributed.

We first applied a Yeo–Johnson transformation to each variable with $\hat{\lambda}$ obtained from the nonrobust ML method of (11). For each variable we then standardized the transformed data $h_{\hat{\lambda}}(x_i)$ to $(h_{\hat{\lambda}}(x_i) - \hat{\mu}_{\text{ML},\hat{\lambda}})/\hat{\sigma}_{\text{ML},\hat{\lambda}}$ where $\hat{\mu}_{\text{ML},\hat{\lambda}}$ and $\hat{\sigma}_{\text{ML},\hat{\lambda}}$ are given by (12). This yields a standardized transformed data set with again 180 rows and 500 columns. In order to detect outliers in this matrix we compare each value to the interval $[-2.57, 2.57]$ which has a probability of exactly 99% for standard normal data. The top panel of Fig. 11 is a heatmap of the standardized transformed data matrix where each value within $[-2.57, 2.57]$ is shown as yellow, values above 2.57 are red, and values below -2.57 are blue. This heatmap is predominantly yellow because the ML method tends to transform the data in a way that masks outliers, so not much structure is visible.

Next, we transformed each variable by Yeo–Johnson with $\hat{\lambda}$ obtained by the robust RewML method. The transformed variables were standardized accordingly to $(h_{\hat{\lambda}}(x_i) - \hat{\mu}_{\text{W},\hat{\lambda}})/\hat{\sigma}_{\text{W},\hat{\lambda}}$ where $\hat{\mu}_{\text{W},\hat{\lambda}}$ and $\hat{\sigma}_{\text{W},\hat{\lambda}}$ are given by (14) using the final weights in (13). The resulting heatmap is in the bottom panel of Fig. 11. Here we see much more structure, with red regions corresponding to glass samples with unusually high spectral intensities at certain wavelengths. This is because the RewML method aims to make the central part of each variable as normal as possible, which allows outliers to deviate from that central region. The resulting heatmap has a subject-matter interpretation since wavelengths correspond to chemical elements. It indicates that some of the glass samples (with row numbers between 22 and 30) have a higher concentration of phosphor, whereas rows 57–63 and 74–76 had an unusually high amount of calcium. The

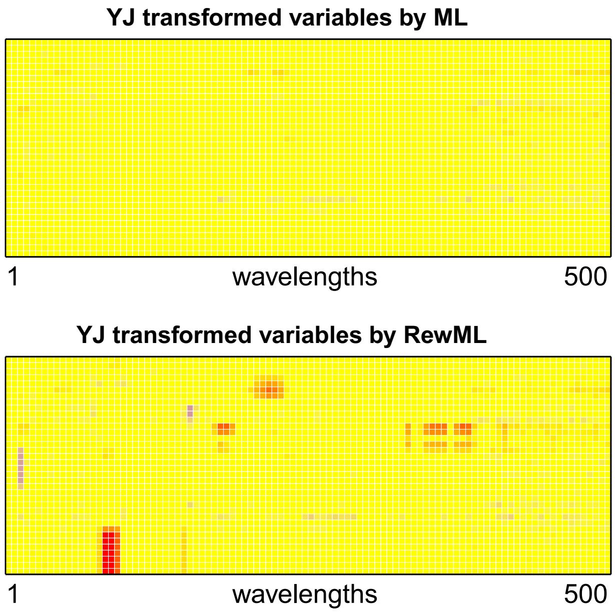

Fig. 11 Heatmap of the `glass` data after transforming each variable (column) by a Yeo–Johnson transform with parameter λ estimated by (top) the maximum likelihood method, and (bottom) the reweighted maximum likelihood method RewML. Yellow cells correspond to values in the central 99% range of the normal distribution. Red cells indicate unusually high values, and blue cells have unusually low values (Color figure online)

red zones in the bottom part of the heatmap were caused by the fact that the measuring instrument was cleaned before recording the last 38 spectra.

4.3 DPOSS data

As a final example we consider data from the Digitized Palomar Sky Survey (DPOSS) described by Djorgovski et al. (1998). We work with the dataset of 20000 stars available as `dposs` in the R package `cellWise` (Raymaekers et al. 2020). The data are measurements in three color bands, but for the purpose of illustration we restrict attention to the color band with the fewest missing values. Selecting the completely observed rows then yields a dataset of 11478 observations with 7 variables. Variables MAper, MTot and MCore measure light intensity, and variables Area, IR2, csf and Ellip are measurements of the size and shape of the images.

In order to analyze the data we first apply the YJ-transformation to each variable, with the $\hat{\lambda}$ estimates obtained from RewML. We then perform cellwise robust PCA (Hubert et al. 2019). We retained $k = 4$ components, explaining 97% of the variance. Figure 12 is the pairs plot of the robust scores. We clearly see a large cluster of regular points with some outliers around it, and a smaller cluster of rather extreme outliers in red. The red points correspond to the stars with a high value of MAper. The blue points are somewhat in

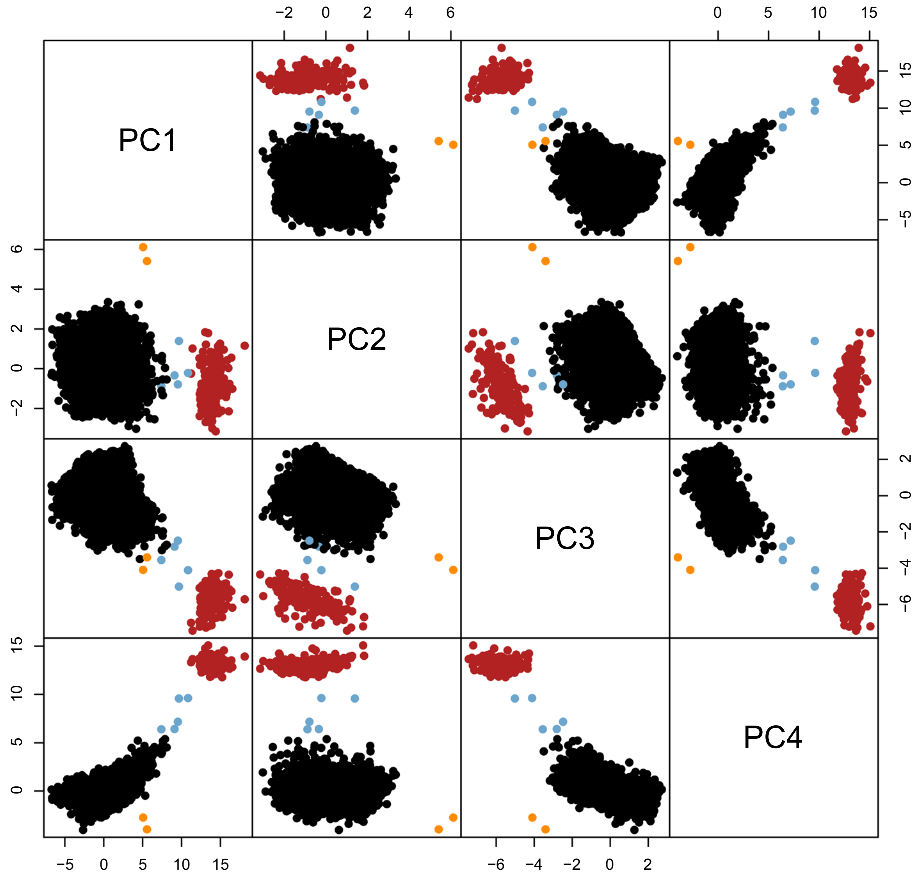

Fig. 12 DPOSS data: pairs plot of the robust PCA scores after transforming the data with the robustly fitted YJ-transformation

between the main cluster and the smaller cluster of extreme outliers. The two orange points are celestial objects with an extreme value of Ellip.

The left panel of Fig. 13 contains the *outlier map* (Hubert et al. 2005) of this robust PCA. Such an outlier map consists of two ingredients. The horizontal axis contains the *score distance* of each object $\mathbf{x}_i$. This is the robust Mahalanobis distance of the orthogonal projection of $\mathbf{x}_i$ on the subspace spanned by the k principal components, and can be computed as

$$\mathrm{SD}_i = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{\lambda_j}} \tag{18}$$

where t_{ij} are the PCA scores and λ_j is the j-th largest eigenvalue. The vertical axis shows the *orthogonal distance* OD_i of each point $\mathbf{x}_i$, which is the distance between $\mathbf{x}_i$ and its projection on the k-dimensional subspace. We see that the red points form a large cluster with extreme SD_i as well as OD_i. The blue points are intermediate, and the two orange points are still unusual but less extreme. Interestingly, when applying robust PCA to the

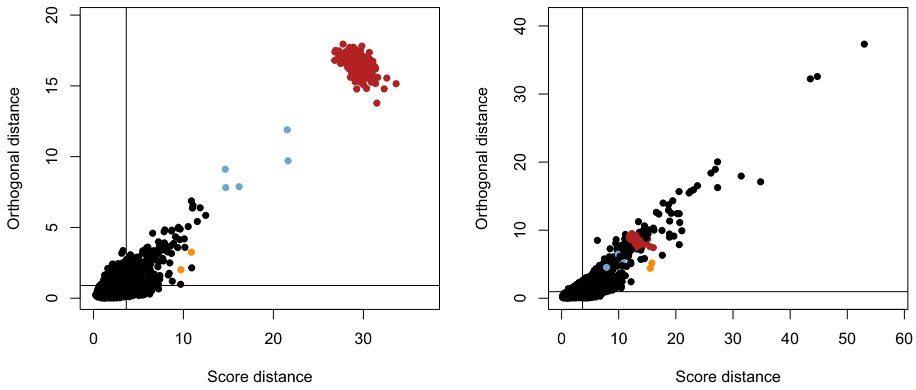

Fig. 13 DPOSS data: outlier map of robust PCA applied to the YJ-transformed variables (left), and applied to the raw variables (right). The colors are those of Fig. 12 (Color figure online)

untransformed data we obtain the right panel of Fig. 13 which shows a very different picture in which the red, blue and orange points are not clearly distinguished from the remainder. This outlier map appears to indicate a number of black points as most outlying, but in fact they should not be considered outlying since they are characterized by high values of Area and IR2 which are very right-skewed. When transforming these variables to central normality, as in the left panel, the values that appeared to be outlying become part of the regular tails of a symmetric distribution.

5 Discussion

In this discussion we address and clarify a few aspects of (robust) variable transformation.

Outliers in skewed data The question of what is an outlier in a general skewed distribution is a rather difficult one. One could flag the values below a lower cutoff or above an upper cutoff, for instance given by quantiles of a distribution. But it is hard to robustly determine cutoffs for a general skewed dataset or distribution. Our viewpoint is this. If the data can be robustly transformed by BC or YJ towards a distribution that looks normal except in the tails, then it easy to obtain cutoffs for the transformed data, e.g. by taking their median plus or minus two median absolute deviations. And then, since the BC and YJ transforms are monotone, these cutoffs can be transformed back to the original skewed data or distribution. In this way we do not flag too many points. We have illustrated this for lognormal data in section C of the supplementary material.

How many outliers can we deal with? In addition to sensitivity curves, which characterize the effect of a single outlier, it is interesting to find out how many adversely placed outliers it takes to make the estimators fail badly. For RewML this turns out to be around 15% of the sample size, which is quite low compared to robust estimators of location or scale. But this is unavoidable, as asymmetry is tied to the tails of the distribution. Empirically, skewness does not manifest itself much in the central part of the distribution, say 20% of mass above and below the median. On each side we have 30% of mass left, and if the adversary is allowed to replace half of either portion (that is, 15% of the total mass) and move it *anywhere*, they can modify the skewness a great deal.

Prestandardization In practice, one would typically apply some form of prestandardization to the data before trying to fit a transformation. For the YJ transformation, we can simply prestandardize the dataset $X = \{x_1, \dots, x_n\}$ by

$$\tilde{x}_i = \frac{x_i - \text{ median } (X)}{\text{mad } (X)} \tag{19}$$

where mad is the median absolute deviation from the median. This is what we did in the glass and DPOSS examples. An advantage of this prestandardization is that the relevant λ will typically be in the same range (say, from −4 to 6) for all variables in the data set, which is convenient for the computation of $\hat{\lambda}$. Before the BC transformation we cannot prestandardize by (19) since this would generate negative values. One option is to divide the original x_i by their median so they remain positive and have median 1. But if the resulting data is tightly concentrated around 1, we may still require a huge positive or negative λ to properly transform them by BC. Alternatively, we propose to prestandardize by

$$\tilde{x}_i = \exp\left(\frac{\log(x_i) - \text{ median } (\log X)}{\text{mad } (\log X)}\right) \tag{20}$$

which performs a standardization on the log scale and then transforms back to the positive scale. This again has the advantage that the range of λ can be kept fixed when computing $\hat{\lambda}$, making it as easy as applying the YJ transform after (19). A disadvantage is that the transformation parameter becomes harder to interpret, since e.g. a value of $\lambda = 1$ no longer corresponds to a linear transformation, but $\lambda = 0$ still corresponds to a logarithmic transform.

Tuning constants The proposed method has some tuning parameters, namely the constant c in (6), the cutoff $\Phi^{-1}(0.995)$ in the reweighting step, and the rectification points. The tuning of the constant c is addressed in section A of the supplementary material. The second parameter is the constant $\Phi^{-1}(0.995) \approx 2.57$ in the weights (16). This weight function is commonly used in existing reweighting techniques in robust statistics. If the transformed data is normally distributed we flag approximately 1% of values with this cutoff, since $P(|Z| > \Phi^{-1}(0.995)) = 0.01$ if $Z \sim \mathcal{N}(0, 1)$. Choosing a higher cutoff results in a higher efficiency, but at the cost of lower robustness. Generally, the estimates do not depend too much on the choice of this cutoff as long as it is in a reasonably high range, say from $\Phi^{-1}(0.975)$ to $\Phi^{-1}(0.995)$. A final choice in the proposal are the "rectification points" C_ℓ and C_u for which we take the first and third quartiles of the data. Note that the constraints $C_\ell < 0 < C_u$ for YJ and $C_\ell < 1 < C_u$ for BC in (7)–(10) are satisfied automatically when prestandardizing the data as described in the previous paragraph. It is worth noting that these data-dependent choices of C_ℓ and C_u do not create abrupt changes in the transformed data when moving through the range of possible values for λ because the rectified transformations are continuous in λ by construction. For instance, passing from $\lambda < 1$ to $\lambda > 1$ does not cause a jump in the transformed data because $\lambda = 1$ corresponds to a linear transformation that is inherently rectified on both sides.

Models with transformed variables The ease or difficulty of interpreting a model in the transformed variables depends on the model under consideration. For nonparametric models it makes little difference. For parametric models the transformations can make the model harder to interpret in some cases and easier in others, for instance where there is a simple linear relation in the transformed variables instead of a model with higher-order terms in the original variables. Also, the notion of leverage point in linear regression is more easily interpretable with roughly normal regressors, as it is related to the (robust) Mahalanobis distance of a point in regressor space.

The effect of the sampling variability of λ on the inference in the resulting model is also model dependent. We expect that it will typically lead to a somewhat higher variability in the estimation. However, as the BY and YJ transformations are continuous and differentiable in λ and not very sensitive to small changes of λ, the increase in variability is likely small. Moreover, if we apply BC or YJ transformations to predictor variables used in CART or Random Forest, the predictions and feature importance metrics stay exactly the same because the BC and YJ transformations are monotone.

6 Conclusion

In our view, a transformation to normality should fit the central part of the data well, and not be determined by any outliers that may be present. This is why we aim for central normality, where the transformed data is close to normal (Gaussian) with the possible exception of some outliers that can remain further out. Fitting such a transformation is not easy, because a point that appears to be outlying in the original data may not be outlying in the transformed data, and vice versa.

To address this problem we introduced a combination of three ideas: a highly robust objective function (5), the rectified Box–Cox and Yeo–Johnson transforms in Sect. 2.2 which we use in our initial estimator only, and a reweighted maximum likelihood procedure for transformations. This combination turns out to be a powerful tool for this difficult problem.

Preprocessing real data by this tool paves the way for applying subsequent methods, such as anomaly detection and well-established model fitting and predictive techniques.

Supplementary Information The online version contains supplementary material available at (https://doi.org/10.1007/s10994-021-05960-5).

Acknowledgements This work was supported by Grant C16/15/068 of KU Leuven, Belgium. We thank the referees for constructive comments improving the presentation.

Data availability The proposed method is available as the function `transfo()` in the `R` (R Core Team 2020) package `cellWise` (Raymaekers et al. 2020) on CRAN, which also includes the vignette `transfo_examples` that reproduces all the examples in this paper. It makes uses of the `R`-packages `gridExtra` (Auguie 2017), `reshape2` (Wickham 2007), `ggplot2` (Wickham 2016) and `scales` (Wickham and Seidel 2020).

References

Alfons, A. (2019). robustHD: Robust methods for high-dimensional data. Retrieved November 15, 2020 from https://CRAN.R-project.org/package=robustHD. R package version 0.6.1.

Andrews, D., Bickel, P., Hampel, F., Huber, P., Rogers, W., & Tukey, J. (1972). *Robust estimates of location*. Princeton, NJ: Princeton University Press.

Auguie, B. (2017). gridExtra: miscellaneous functions for “grid” graphics. Retrieved November 15, 2020 from https://CRAN.R-project.org/package=gridExtra. R package version 2.3.

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society, Series B, 26,* 211–252.

Carroll, R. J. (1980). A robust method for testing transformations to achieve approximate normality. *Journal of the Royal Statistical Society. Series B, 42,* 71–78.

Djorgovski, S., Gal, R., Odewahn, S., De Carvalho, R., Brunner, R., Longo, G., Scaramella, R. (1998). The palomar digital sky survey (dposs). arXiv preprint arxiv:astro-ph/9809187.

Hoaglin, D., Mosteller, F., & Tukey, J. (1983). *Understanding robust and exploratory data analysis*. New York: Wiley.

Huber, P. (1981). *Robust statistics*. New York: Wiley.

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

Hubert, M., Rousseeuw, P. J., & Vanden Branden, K. (2005). ROBPCA: A new approach to robust principal component analysis. *Technometrics, 47,* 64–79.

Lemberge, P., De Raedt, I., Janssens, K., Wei, F., & Van Espen, P. (2000). Quantitative z-analysis of 16th–17th century archaeological glass vessels using PLS regression of EPXMA and XRF data. *Journal of Chemometrics, 14,* 751–763.

Marazzi, A., Villar, A. J., & Yohai, V. J. (2009). Robust response transformations based on optimal prediction. *Journal of the American Statistical Association, 104,* 360–370.

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raymaekers, J., Rousseeuw, P., Van den Bossche, W., Hubert, M. (2020). cellWise: Analyzing data with cellwise outliers. https://CRAN.R-project.org/package=cellWise. R package version 2.2.2.

Riani, M. (2008). Robust transformations in univariate and multivariate time series. *Econometric Reviews, 28,* 262–278.

Rousseeuw, P. J., & Van den Bossche, W. (2018). Detecting deviating data cells. *Technometrics, 60,* 135–145.

Tukey, J. W. (1957). On the comparative anatomy of transformations. *Annals of Mathematical Statistics, 28,* 602–632.

Van der Veeken, S. (2010). *Robust and nonparametric methods for skewed data*. Ph.D. thesis, KU Leuven, Belgium.

Wickham, H. (2007). Reshaping data with the reshape package. *Journal of Statistical Software, 21*(12), 1–20.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. New York: Springer.

Wickham, H., Seidel, D. (2020) scales: Scale functions for visualization. https://CRAN.R-project.org/package=scales. R package version 1.1.1.

Yeo, I. K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika, 87,* 954–959.

Supplementary Material

A Choice of the tuning constant c

The function ρ_{bw} in (6) contains a tuning constant c which we set at 0.5. This tuning constant quantifies a trade-off between variance and robustness. For smaller values of c the central part of Tukey's ρ-function becomes more narrow, making it less likely that clean observations fall into this central part. For larger values of c the opposite happens. Ideally, we would like the outliers to fall outside of the central region, and as many of the inliers as possible inside the central region. In section 5 it was argued that robustly fitting a skewness-correcting transform like BC or YJ can handle at most about 15% of outliers. We have tuned c such that with 15% of extreme contamination, approximately 90% of the clean data fall into the central region.

The computation goes as follows. Suppose we have a data set $x_1, \ldots, x_n$ of size n and assume w.l.o.g. that $x_1 \leq x_2 \leq \ldots \leq x_n$. Assume furthermore that a fraction $1 - \varepsilon$ can be transformed to a gaussian distribution, but the remaining fraction ε consists of extreme outliers: $x_i \approx \infty$ for $i = (1-\varepsilon)n + 1, \ldots, n$. (The situation where $x_i \approx -\infty$ for $i = 1, \ldots, \varepsilon n$ is analogous.) For the regular observations, the differences between the observed and expected quantiles in the objective function (5) are close to

$$\Delta_i = \Phi^{-1}\left(\frac{i - 1/3}{(1-\varepsilon)n + 1/3}\right) - \Phi^{-1}\left(\frac{i - 1/3}{n + 1/3}\right)$$

for $i = 1, \ldots, (1-\varepsilon)n$. We can verify that $P(|\Delta| \leq 0.5) \approx 88\%$ when $n \geqslant 100$.

B Additional simulations

In this section we show some additional simulation results. The plots complement the figures shown in the main text by considering a different position of the outliers ($k = 6$) for an increasing contamination level ε, and also the case of $k = 10$ for the BC transformation. The results shown here are in line with the conclusions of the main text.

B.1 the BC transformation

Figure 14 shows the effect of an increasing level of contamination on the estimates of the BC-transformation parameter when the position of the contamination is fixed at $k = 10$. The proposed RewML did well, as it has a very stable performance which is not affected much by the increasing level of contamination. The reweighted estimator starting from the initial estimator without rectification (RewMLnr) performs less well, especially when the percentage of contamination increases. Carroll's method and the classical maximum likelihood estimator are more heavily affected by the outliers, and perform poorly. The trimmed likelihood estimators display a very erratic behavior and are unreliable in this setup.

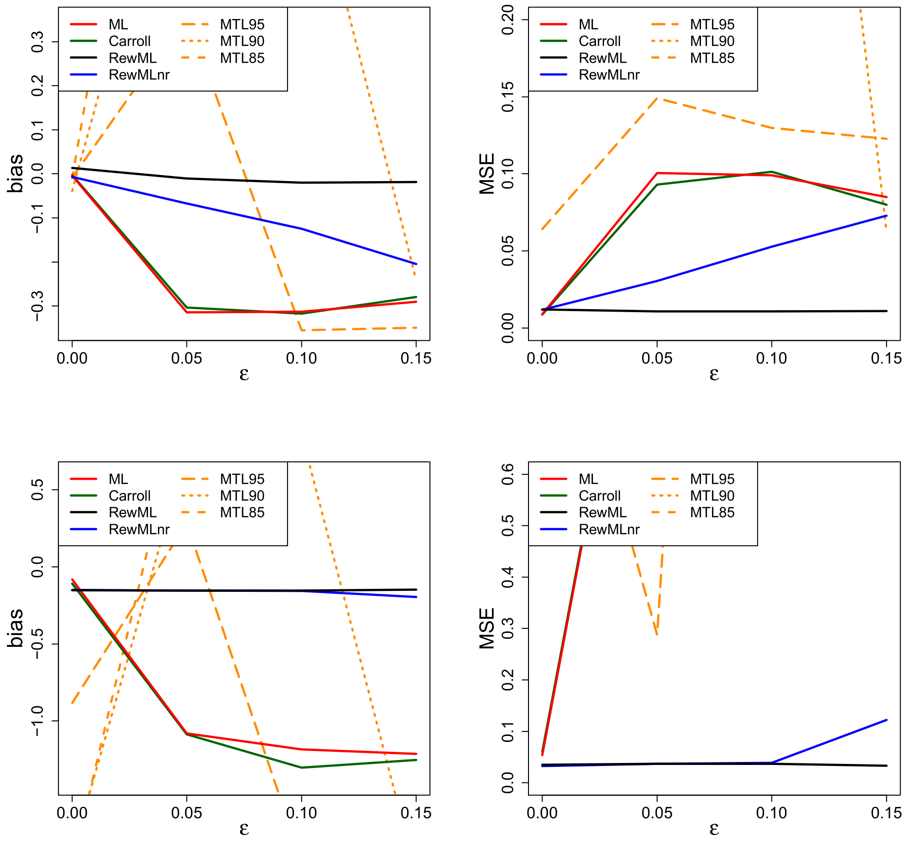

Fig. 14 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Box-Cox transformation as a function of the percentage ε of outliers, when the location of the outliers is determined by setting $k = 10$. The true parameter λ used to generate the data is 0 in the top row 1 in the bottom row.

Figure 15 shows the effect of an increasing level of contamination on the estimates of the BC-transformation parameter when the position of the contamination is fixed at $k = 6$. The general conclusions based from this figure are the same as for the case $k = 10$ in the main text. We note that $k = 6$ is slightly more difficult than $k = 10$ in Figure 14, as evidenced by the more pronounced effect of increasing the contamination level.

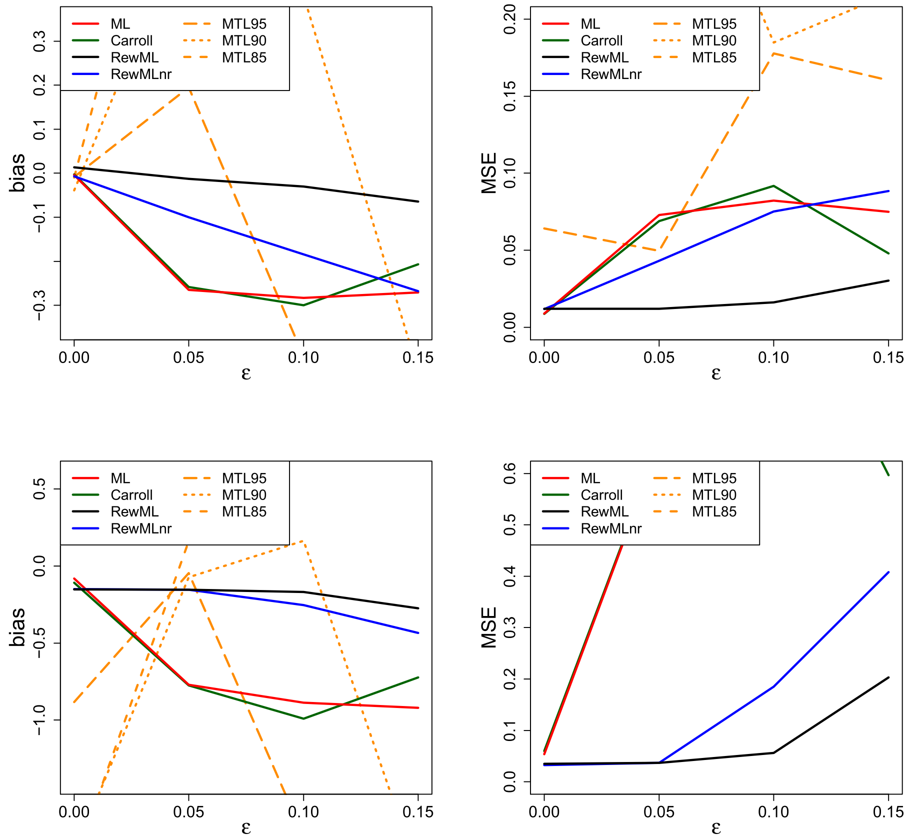

Fig. 15 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Box-Cox transformation as a function of the percentage ε of outliers, when the location of the outliers is determined by setting $k = 6$. The true parameter λ used to generate the data is 0 in the top row, and 1 in the bottom row.

B.2 The YJ transformation

Figure 16 shows the effect of an increasing level of contamination on the estimates of the YJ transformation parameter when the position of the contamination is fixed at $k = 6$. The performance of the estimators in this setting is very similar to the case of $k = 10$, as can be seen in a comparison with Figure 6 of the main text. The trimmed estimators behave as they are supposed to for $\lambda = 1$, but they become unreliable as soon as λ deviates from 1. The ML estimator and Carroll's estimator are more heavily influenced by the outliers, but the magnitude of the effect is slightly smaller than for $k = 10$. The proposed RewML estimator performs best, and we again see that starting with an initial estimator without rectification (RewMLnr) substantially reduces the robustness of the reweighted estimator.

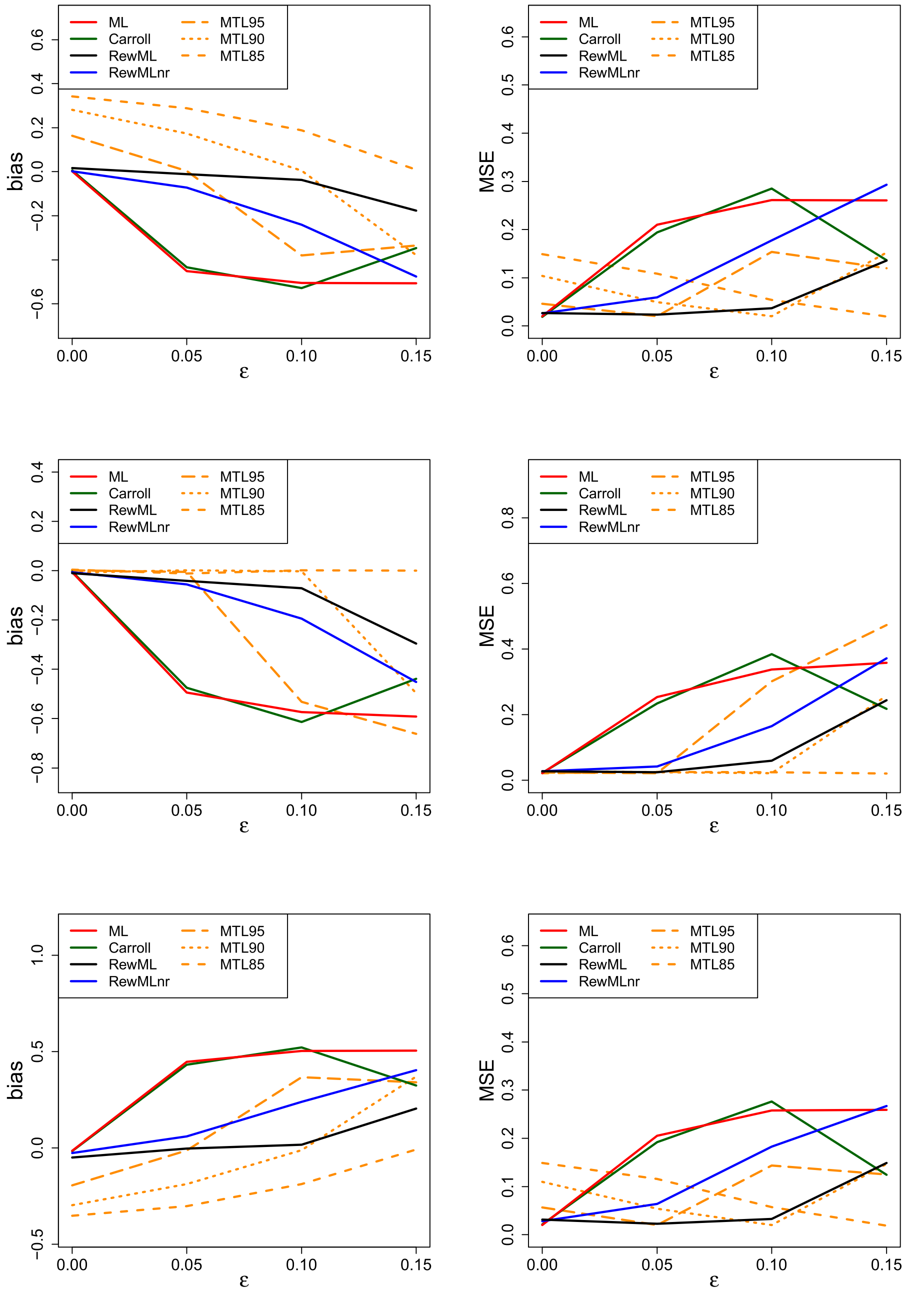

Fig. 16 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Yeo-Johnson transformation as a function of the percentage ε of outliers, when the location of the outliers is determined by setting $k = 6$. The true parameter λ used to generate the data is 0.5 in the top row, 1.0 in the middle row, and 1.5 in the bottom row.

C False detections

We briefly investigate whether the proposed method is well-calibrated when it comes to outlier detection. More specifically, we track the number of "false positives", i.e. the number of observations flagged as outliers when the data comes from a clean distribution. The clean data was generated from the lognormal distribution and the sample size was given by $n = 10^j$ for $j = 2, \dots, 6$. For each sample size 100 replications were run. These uncontaminated data were Box-Cox transformed with $\hat{\lambda}$ obtained from the proposed RewML estimator. Figure 17 shows boxplots of the percentage of points whose weight given by (16) is zero. As the sample size n increases the fraction of flagged points approaches 1%, which is in line with the expected number of flagged points since the cutoffs on the transformed data in (16) are at the 0.005 and 0.995 quantiles.

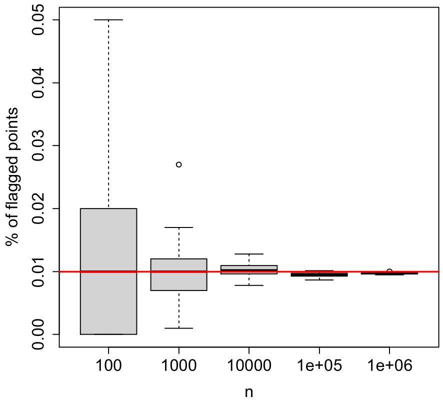

Fig. 17 Percentage of points flagged as outlying in clean lognormal data, for various sample sizes. As the sample size increases, the fraction of flagged points approaches the expected frequency of 1%.

D Results with and without reweighting

In this section we illustrate the importance of the reweighting steps in RewML. Switching them off yields the initial estimator, which was left out of the figures in the main text because it fits a rectified transform, so it is effectively estimating a different transformation. Therefore, a direct comparison between the initial rectified estimator and the other estimators is questionable.

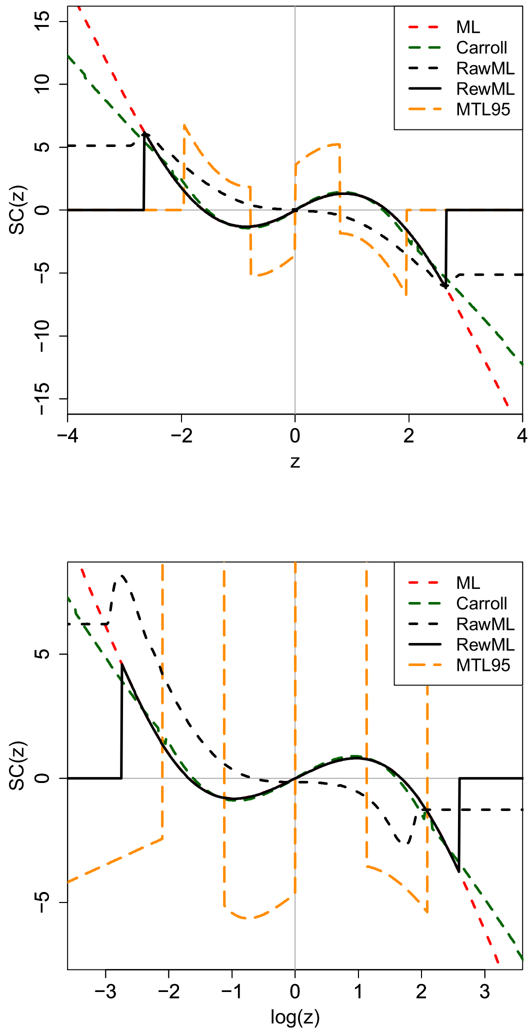

Fig. 18 Sensitivity curves of estimators of the parameter λ in the Yeo-Johnson (top) and Box-Cox (bottom) transformations, with sample size $n = 100$.

Figure 18 shows the sensitivity curves including those of the initial estimator, denoted as RawML. We see that RawML has a bounded sensitivity curve, but it does not redescend to zero for large values of $|z|$. The sensitivity curve of the reweighted estimator behaves like the maximum likelihood estimator in the center, but is zero for large values of $|z|$. In the Box-Cox setting (lower panel of Figure 18) the sensitivity curve of RawML is asymmetric because the rectification is one-sided for $\lambda < 1$, but the reweighting steps make the curve more symmetric again.

We now extend Figures 7 and 8 of the main text, which showed the bias and MSE of the estimators for 10% of contamination positioned at an increasing deviation k from the center of the distribution. Figure 19 shows the results for the YJ transformation with the initial estimator added. We see that the initial estimator behaves roughly like the reweighted estimator for small k, but the bias does not redescend as the outliers are moved further away from the center. As a result, its MSE only stabilizes at a higher value of k.

Figure 20 shows the corresponding results for the BC transformation. A similar conclusion can be drawn here. For both the YJ and BC transforms it is clear that the application of the redescending weight function (16) reduces the effect of far outliers on $\hat{\lambda}$.

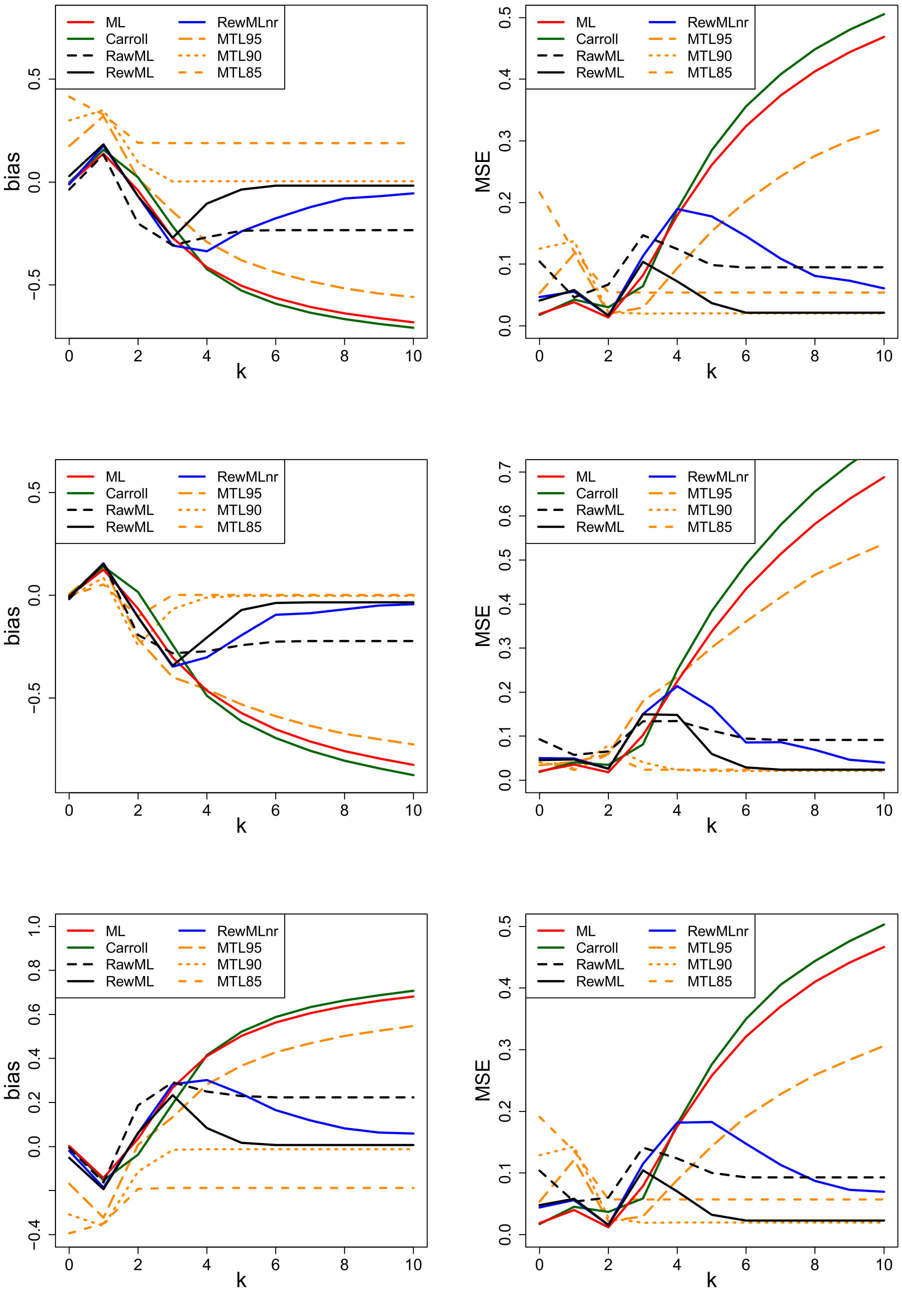

Fig. 19 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Yeo-Johnson transformation as a function of k which determines how far the outliers are. Here the percentage of outliers is fixed at 10%. The true parameter λ used to generate the data is 0.5 in the top row, 1.0 in the middle row, and 1.5 in the bottom row.

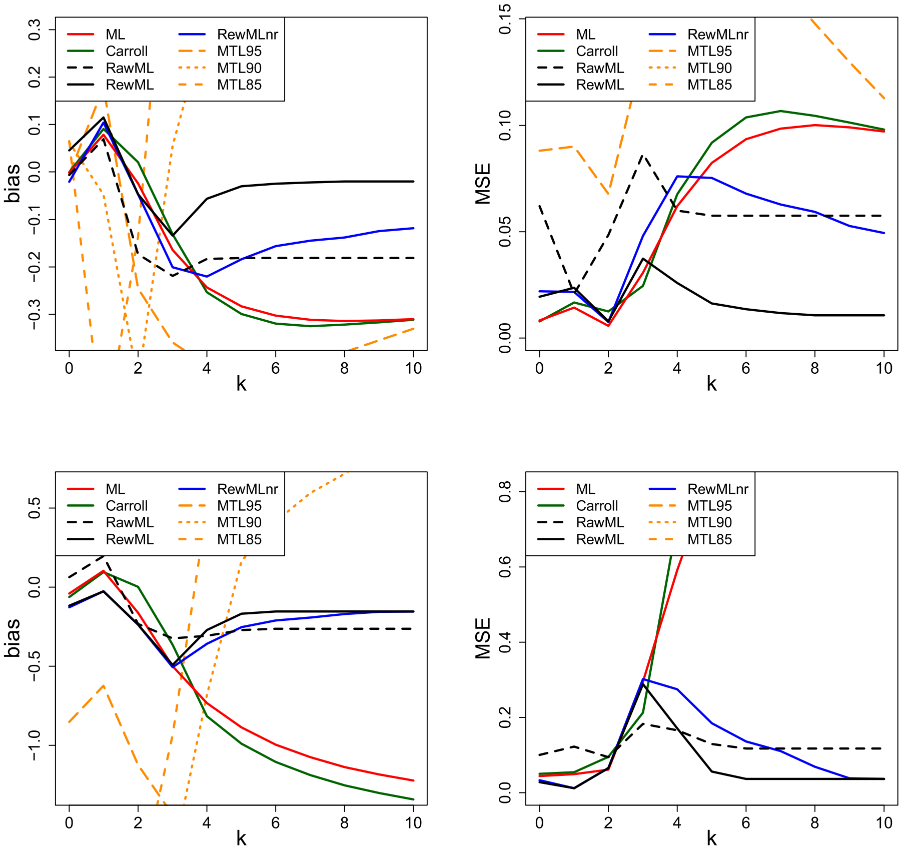

Fig. 20 Bias (left) and MSE (right) of the estimated $\hat{\lambda}$ of the Box-Cox transformation as a function of k which determines how far the outliers are. Here the percentage of outliers is fixed at 10%. The true λ used to generate the data is 0 in the top row and 1 in the bottom row.