
**UB Riskcenter** Working Paper Series

University of Barcelona

**Research Group on Risk in Insurance and Finance** www.ub.edu/riskcenter

Working paper 2014/01 \\ Number of pages 34

# Non-parametric Models for Univariate Claim Severity Distributions – an approach using R

Catalina Bolancé, Montserrat Guillén and David Pitt

Riskcenter

# Non-parametric Models for Univariate Claim Severity Distributions - an approach using R

Catalina Bolancé[1], Montserrat Guillen[1] and David Pitt[2]

[1] Dept. Econometrics, RFA-IREA, University of Barcelona

Diagonal, 690, 08034 Barcelona, Spain

E-mail: mguillen@ub.edu

[2] Department of Applied Finance and Actuarial Studies, Macquarie University

Sydney, New South Wales, 2109 Australia

E-mail: david.pitt@mq.edu.au

**Abstract**

This paper presents an analysis of motor vehicle insurance claims relating to vehicle damage and to associated medical expenses. We use univariate severity distributions estimated with non-parametric methods. The methods are implemented using the statistical package R. The nonparametric analysis presented involves kernel density estimation. We illustrate the benefits of applying transformations to data prior to employing kernel based methods. We use a log-transformation and an optimal transformation amongst a class of transformations that produces symmetry in the data. The central aim of this paper is to provide educators with material that can be used in the classroom to teach statistical estimation methods, goodness of fit analysis and importantly statistical computing in the context of insurance and risk management. To this end, we have included in the Appendix of this paper all the R code that has been used in the analysis so that readers, both students and educators, can fully explore the techniques described.

**Key words:** loss modeling; insurance; education

## 1 Introduction

Kernel density estimation is an easy nonparametric method to analyze the distribution of a random variable, that unlike parametric models, requires little assumptions. When analyzing the cost of individual claims in non-life insurance, we often encounter right skewness, because there are lots of small claims while only a few claims have a very large cost. When there is considerable skewness it is not well established in the insurance literature that classical kernel estimation is not a good method for approximating the probability density function (pdf) or the cumulative distribution function (cdf) for claim costs. In this work we show how non-parametric estimation of the pdf for right skewed random variables can be done in practice. We show an example using motor vehicle claim cost data and provide the R code that is necessary to implement this approach.

The purpose of the analysis presented here is to illustrate univariate density estimation procedures using non-parametric methods and to provide educators in insurance and risk analysis with a fully worked example of this form of data analysis using the statistical package R. We only consider the estimation of separate univariate models for two sets of positive insurance claims data. Bivariate analysis of these data, including estimation of correlations between claim cost types have been considered by [12] and [4] where bivariate skew-normal and bivariate normal distributions were fitted. Given that real claim severity data are usually positive and right-skewed, [4] also fitted the bivariate lognormal and log-skew-normal distributions along with a bivariate kernel density estimate.

Density estimation is necessary in insurance for many reasons including pricing and optimal capital allocation (see [8], [14], [9] and [26]). The book, [11] provides a comprehensive reference on the estimation of univariate and bivariate claims distribution models in insurance. In [13] an overview on risk measures for loss distributions is provided.

We study two positive claim cost datasets from a major Spanish motor insurer, namely property damage mainly resulting from third party liability and medical expenses that are not covered by the public health system. To obtain all the results in this paper we use the software R and the QRM library (see [13]).

Next, in section 2 we describe the kernel density estimator and the transformed kernel density estimator. In section 3 we present different measures of goodness of fit for non-parametric estimations. Then, in section 4 we describe the data set used in our application. Finally, we present the results and conclusions. The R programs used to obtain results are shown in the Appendix.

# 2 Kernel density estimation

## 2.1 Classical kernel density estimation

For a random sample of $n$ independent and identically distributed observations $x_1, ..., x_n$ of a random variable $X$ with pdf $f_X$, the kernel density estimator is

$$\hat{f}_X(x) = \frac{1}{n}\sum_{i=1}^{n} K_b(x - x_i), \tag{1}$$

where $K_b(\cdot) = \frac{1}{b}K(\cdot/b)$, $K$ is the kernel function and $b$ is the bandwidth (see [24]). The bandwidth parameter is used to control the amount of smoothing in the estimation so that the greater $b$ is, the smoother is the estimated density curve. The kernel function is usually a symmetric density with zero mean. In our work we use a Gaussian kernel, that is

$$K(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right).$$

Many methods have been proposed for the selection of the bandwidth parameter in kernel density estimation. In the R program function 'density' used in this paper the "rule of thumb" bandwidth parameter of Silverman (see [19], Chapter 3) is used with the sample inter-quartile range used as the dispersion parameter. Unbiased and biased cross-validation methods and the plug-in method proposed by Sheather and Jones (see [18]) are also available in R. We use all these methods and we select the one which represents better our pdf. Selection is made by comparison of the fitted pdf with the empirical histogram. Note that when we use unbiased and biased cross- validation methods with skewed data, we can have problems obtaining a value for $b$. When data have right skewness these methods may not produce a global minimum and the value of $b$ may be at the boundary of the grid. For this reason we also consider transformations.

## 2.2 Transformations and kernel density estimation

Classical kernel density estimation does not generally perform well when the true density is asymmetric. For instance, when one is interested in the density of the claim cost variable, the presence of many small claims produces a concentration of mass near the low values of the domain and the presence of some very large claims causes positive skewness.

The lack of information in the right tail of the domain makes it difficult to obtain a reliable non-parametric estimate of the density in that area. Many authors have worked with heavy-tailed distributions and have adapted kernel estimation methods to this context. Different papers have proposed different transformed kernel estimation (TKE) methods for a pdf, based on parametric families (see [25], [7], [5], [6], [3], [2] and [1]).

Let $T(.)$ be an increasing and monotonic transformation function that has a first derivative $T'(.)$. If the true density is right-skewed, then the chosen transformation $T(.)$ must be a concave function. In this paper we propose a TKE of the pdf that consists of transforming the original data $y_i = T(x_i)$ so that the new transformed data can be assumed to have been generated from a symmetric random variable $Y$ and hence the true density of the transformed variable can be reliably approximated using the classical kernel estimation method. Using a change of variable, once the kernel estimate is obtained for the transformed variable, estimation on the original scale is also obtained.

In [5] the authors proposed to select the transformation function from a transformation family proposed first in [25] namely the shifted power transformation family,

$$T_\lambda(x) = \begin{cases} (x+\lambda_1)^{\lambda_2}\, sign(\lambda_2) \\ \ln(x+\lambda_1) \end{cases}, \tag{2}$$

where $\lambda = (\lambda_1, \lambda_2)$, $\lambda_1 \geq -\min(x_i, i = 1, ..., n)$ and $\lambda_2 \leq 1$ for right-skewed data. This approach has a simple mathematical formulation and works particularly well for TKE of asymmetric distributions. In order to estimate the optimal parameters of the shifted power transformation function, the algorithm described in [5] can be used.

Let us assume a sample of $n$ independent and identically distributed observations $x_1, ..., x_n$ is available. We also assume that a transformation function $T_\lambda(\cdot)$ has been selected so that the data can be transformed to give $y_i = T_\lambda(x_i)$, $i = 1, ..., n$. We denote the transformed sample by $y_1, ..., y_n$.

Having transformed the data, we then estimate the density of the transformed data set using the classical kernel density estimator

$$\widehat{f}_Y(y) = \frac{1}{n} \sum_{i=1}^{n} K_b\left(y - y_i\right),$$

where $K_b(\cdot) = \frac{1}{b} K(\cdot/b)$, $K$ is the kernel function and $b$ is the bandwidth. The estimator of the original density is obtained by back-transformation of $\widehat{f}_Y(y)$:

$$\widehat{f}_X(x, \lambda) = T'_\lambda(x)\, \widehat{f}_Y(y) = \frac{T'_\lambda(x)}{n} \sum_{i=1}^{N} K_b\left\{T_\lambda(x) - T_\lambda(x_i)\right\}, \tag{3}$$

where as we have mentioned we have assumed that the transformations are differentiable. The estimator defined in (3) is called the transformed kernel density estimator.

In order to implement the transformation approach, a method to select the transformation parameters, $\lambda = (\lambda_1, \lambda_2)$, and the bandwidth, $b$ is necessary.

### 2.3 Selecting the transformation parameters and the bandwidth

As in [5], we restrict the set of transformation parameters, $\lambda = (\lambda_1, \lambda_2)$, to those values that give approximately zero skewness for the transformed data $y_1, .., y_n$ (which have also been scaled to have the same variance as the original sample, see [25]).

We define our sample measure of skewness as:

$$\widehat{\gamma}_y = \frac{n^{-1} \sum\limits_{i=1}^{n} (y_i - \overline{y})^3}{\left\{ n^{-1} \sum\limits_{i=1}^{n} (y_i - \overline{y})^2 \right\}^{\frac{3}{2}}}$$

where $\overline{y}$ is the sample mean.

To select the $\lambda$ parameter vector, we aim at minimizing the mean integrated square error (MISE) of $\widehat{f}_Y(y)$

$$MISE_Y\left(\widehat{f}_Y\right) = E_Y\left[\int_{-\infty}^{+\infty}\left(\widehat{f}_Y(y) - f_Y(y)\right)^2 dy\right],$$

which, as shown in [19], when $b$ is asymptotically optimal, can be approximated by:

$$\frac{5}{4}\left[k_2\alpha(K)^2\right]^{\frac{2}{5}}\beta\left(f_Y''\right)^{\frac{1}{5}}n^{-\frac{4}{5}}, \tag{4}$$

where $k_2 = \int t^2K(t)\,dt$, $\alpha(K) = \int K(t)^2\,dt$ and $\beta(f_Y'') = \int_{-\infty}^{+\infty}\left[f_Y''(y)\right]^2 dy$. Minimizing (4) with respect to the transformation parameters is equivalent to minimizing $\beta(f_Y'')$. The transformation parameters that minimize asymptotically $MISE_Y$ also minimize $MISE_X$ of $\widehat{f}_X(x,\lambda)$ in (3) (see [25]).

In [10] the following estimator for $\beta(f_Y'')$ is suggested:

$$\widehat{\beta}\left(f_Y''\right) = n^{-1}(n-1)^{-1}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c^{-5}K*K\left\{c^{-1}\left(y_i - y_j\right)\right\}, \tag{5}$$

where $K*K(t) = \int_{-\infty}^{+\infty}K(t-s)K(s)ds$ is the kernel convolution and $c$ is the bandwidth used in the estimation of $\beta(f_Y'')$, which can be estimated by minimizing the mean square error (MSE) of $\widehat{\beta}(f_Y'')$. When it is assumed that $f_Y$ is a normal density as in the "rule of thumb" approach, $c$ can be estimated by $\widehat{c} = \widehat{\sigma}_y\left(\frac{21}{40\sqrt{2}n^2}\right)^{\frac{1}{13}}$, where $\widehat{\sigma}_y = \sqrt{n^{-1}\sum_{i=1}^{n}(y_i - \overline{y})^2}$ (see [15] and [25]).

In our application we implement two strategies: we can obtain the transformation parameters by directly minimizing (5) and, alternatively, we can obtain a set of transformation parameters where skewness is zero and then search the transformation parameters that minimize (5) only within this set.

Finally, we need to make the selection of the bandwidth that is going to be used for the transformation. Here we simply use the "rule of thumb" described in Silverman (see [19], Chapter 3) for a standard normal density. Since our transformation aims at a transformed density with zero skewness, this approach seems very plausible. Following [19], the estimator of the bandwidth $b$ is $\widehat{b} = 1.059\widehat{\sigma}_x n^{-\frac{1}{5}}$ or $\widehat{b} = 0.79\widehat{R}_x n^{-\frac{1}{5}}$,where the scale measures $\widehat{\sigma}_x$ and $\widehat{R}_x$ are sample standard deviation and inter-quartile range respectively, and the corresponding transformation estimator will be denoted $\widehat{f}_X(x,\widehat{\lambda};\widehat{b})$.

# 3 Measuring the goodness of fit

We are interested in evaluating the quality of our density estimates obtained using non-parametric methods over the whole domain. Let us begin with the log-likelihood function.

Let us assume that we have $\widehat{f}_X(x)$, an estimate of the density for every point $x$ in the domain. Let us also assume that a sample of $n$ independent and identically distributed observations $x_1, ..., x_n$ is available. Then, we can estimate the log-likelihood function as:

$$\ln \hat{L}(\widehat{f}_X(\cdot); x_1, ..., x_n) = \sum_{i=1}^{n} \ln \widehat{f}_X(x_i).$$

If a transformation method were used giving estimated density $\widehat{f}_X(x, \widehat{\lambda}; \widehat{b})$, the estimated log-likelihood function would be

$$\ln \hat{L}(\widehat{f}_X(\cdot); T_{\widehat{\lambda}}(\cdot); x_1, ..., x_n) = \sum_{i=1}^{n} \ln \widehat{f}_X(x_i, \widehat{\lambda}; \widehat{b}).$$

A widely used measure for evaluating the quality of kernel density estimators over the whole domain is the integrated square error (ISE). Let $\widehat{f}_X(x)$ a kernel estimation of $f_X(x)$, then:

$$ISE_X\left(\widehat{f}_X\right) = \int_{-\infty}^{+\infty} \left(\widehat{f}_X(x) - f_X(x)\right)^2 dx.$$

The problem with applying $ISE_X$ in practice is that it depends on the true density $f_X$ that is unknown. In [19] it is proved that minimizing $ISE_X$ is equivalent to minimizing the cross-validation function:

$$CV_X = \int_{-\infty}^{+\infty} \left[\widehat{f}_X(x)\right]^2 dx - \frac{2}{n} \sum_{i=1}^{n} \widehat{f}_i(x_i), \tag{6}$$

where $\widehat{f}_i$ is the "leave-one-out" estimator, that is the kernel estimate of $f_X$ without observation $x_i$. We can obtain (6) for the transformed kernel density estimation, replacing $\widehat{f}_X(x)$ by $\widehat{f}_X(x, \widehat{\lambda}; \widehat{b})$ and $\widehat{f}_i(x_i)$ by $\widehat{f}_i(x_i, \widehat{\lambda}; \widehat{b})$.

We can generalize the definition of log-likelihood given previously by providing a statistic that gives more weight to the right tail of the distribution. This is important when we require our estimation to be more accurate in the upper right tail of the distribution. Also, we can generalize $ISE_X$ and its approximation in (6) to a weighted $ISE_X$ ($WISE_X$) that gives more weight to the right tail.

A weighted log-likelihood can be estimated if weights $w_i$, $i = 1, ..., n$ are included preceding each summation term as:

$$\ln_w \hat{L}(\widehat{f}_X(\cdot); x_1, ..., x_n) = \sum_{i=1}^{n} w_i \ln \widehat{f}_X(x_i).$$

If $w_i = 1$, $i = 1, ..., n$, then we would have the usual log-likelihood expression. We can also use some distance from zero as a weight, so that observations that are located close to zero have much less importance than those located in the right tail.

We have tried two different expressions for weights. The first one gives more weight to those observations that are distant from zero. Note that our data are always positive. The form of the weights is

$$w_i^{(1)} = \frac{n x_i}{\sum_{i=1}^{n} x_i}.$$

Using these weights in the estimated weighted log-likelihood implies that more importance is given to the fit in the right tail. Then, since for a given $i$, we have that $\ln \widehat{f}_X(x_i)$ is negative and it is smaller when $\widehat{f}_X(x_i)$ tends to zero (which is generally what happens in the long tail region) then weighting those summation terms more, means that the $\ln_w \hat{L}(\widehat{f}_X(\cdot); x_1, ..., x_n) \leq \ln \hat{L}(\widehat{f}_X(\cdot); x_1, ..., x_n)$.

The second form for the weights considered is inspired by the same principle as the weighted integrated mean squared error that was proposed in [5], where contributions are weighted with a squared distance. In this case:

$$w_i^{(2)} = \frac{n x_i^2}{\sum_{i=1}^{n} x_i^2}.$$

When a transformation is used, the corresponding expression is:

$$\ln_w \hat{L}(\widehat{f}_X(\cdot); T_{\widehat{\lambda}}(\cdot); x_1, ..., x_n) = \sum_{i=1}^{n} w_i \widehat{f}_X(x_i, \widehat{\lambda}; \widehat{b}).$$

Similarly, we can approximate a weighted $ISE_X$ ($WISE_X$), weighting by $x$ or by $x^2$:

$$WISE_X^1\left(\widehat{f}_X\right) = \int_{-\infty}^{+\infty} \left(\widehat{f}_X(x) - f_X(x)\right)^2 x dx$$

or

$$WISE_X^2\left(\widehat{f_X}\right) = \int_{-\infty}^{+\infty} \left(\widehat{f}_X(x) - f_X(x)\right)^2 x^2 dx,$$

that can be approximated with:

$$WCV_1 = \int_{-\infty}^{+\infty} \left[\widehat{f}_X(x)\right]^2 x dx - \frac{2}{n}\sum_{i=1}^{n} \widehat{f}_i(x_i)x_i \tag{7}$$

or

$$WCV_2 = \int_{-\infty}^{+\infty} \left[\widehat{f}_X(x)\right]^2 x^2 dx - \frac{2}{n}\sum_{i=1}^{n} \widehat{f}_i(x_i)x_i^2. \tag{8}$$

# 4 Data and results

The claims we considered relate to motor insurance policies of a major insurer in Spain for accidents that occurred in the year 2000. Data correspond to a cost of claims, expressed in thousands of Euros, in a random sample of all claims related to property damage expenses and to medical expenses.

Bodily injury is universally covered by the National Health System. This means that medical costs considered here are medical expenses that are not covered by the public system such as technical aids, drugs or chiropractic-related expenses. Those expenses have to be paid by the insurer. No compensation for pain and suffering or loss of income are included. Medical expenses contain medical costs related to all those who were injured in the accident. Property damage expenses include the insured's liability for damages he or she caused to vehicles, property or objects when the accident occurred.

The claims included in our sample are all claims that have already been settled. Although claims for compensation relating to bodily injury may take a long time to settle, these data were gathered in 2002, so that we can safely assume that there has been enough time for the claimant to include most costs, and we therefore consider that these are closed claims.

The sample contains 518 claims, and for each claim we observe $X_1$ the cost of property damage and $X_2$ the cost of medical expenses, both expressed in thousands of Euros.

Table 1: Univariate descriptive statistics for the positive claims data set (in 1,000 Euros)

| | Mean | Std. Dev. | Skewness | Kurtosis | Min | Max |
|---|---|---|---|---|---|---|
| $X_1$ | 10.984 | 41.276 | 15.652 | 297.142 | 0.078 | 829.012 |
| $X_2$ | 1.706 | 5.188 | 8.037 | 82.019 | 0.006 | 71.250 |

## 4.1 Results of non-parametric fitting

In this section we describe the results of kernel density estimation and different transformed kernel density estimation approaches to univariate claims data. Finally, we calculate the goodness of fit measure that we described in section 3 to compare the proposed estimations.

In Figures 1 and 2 we show the classical kernel density estimates and the log-transformation kernel estimates for both components of the univariate claims data: property damage and medical expenses. In the Appendix we have provided the R commands used to obtain the density estimates shown in Figures 1 and 2. In program R the plug-in method proposed by [18] (bw="sj") is used. A numeric value for the bandwidth $b$ in classical kernel density estimation can be imposed using R.

In Figure 1c and 1d we show the classical kernel density estimates in the right tail of the pdf. We note that the density does not have a smooth shape, as it has some bumps around the sample observations. In Figures 2a and 2b we can see that there is a considerable improvement in the kernel density estimate when it is applied to the log-transformed claim data compared to the fit obtained when applied to the untransformed data. Based on this fact, we further explore the optimal transformation to apply to our claim datasets.

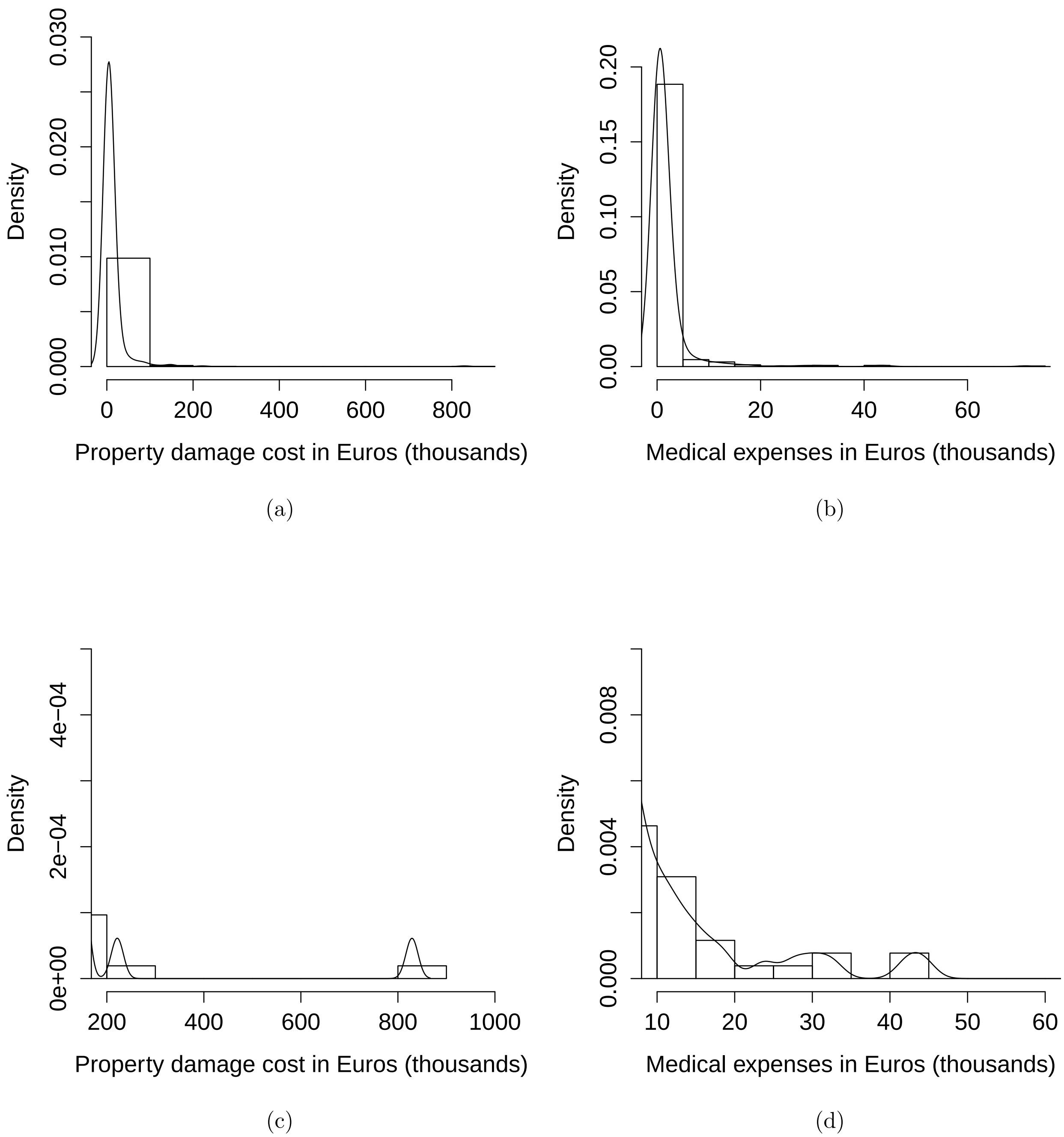
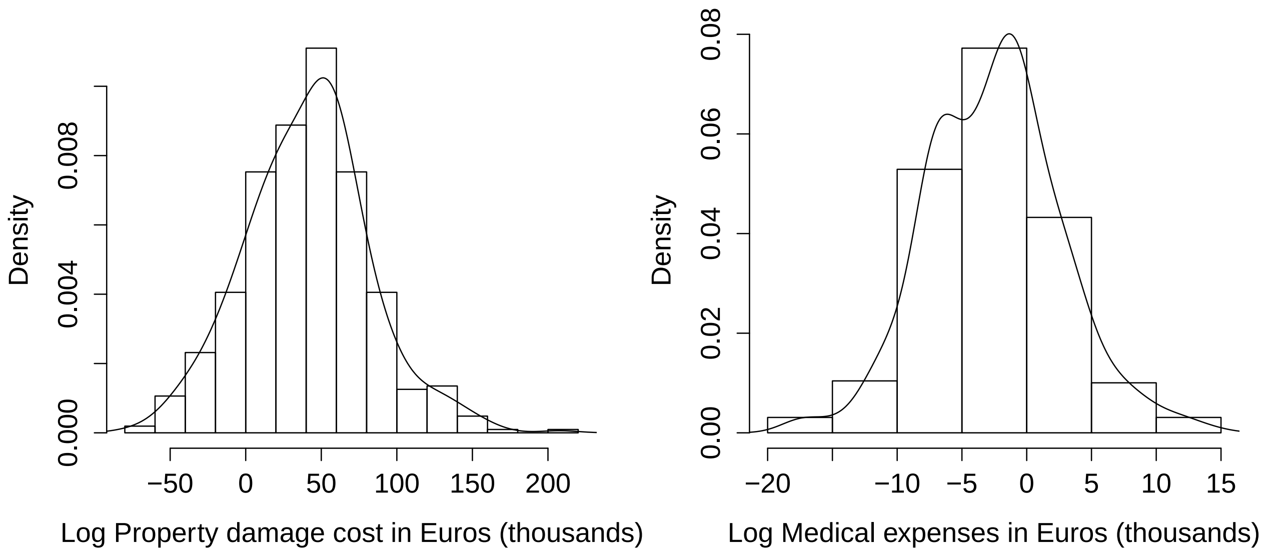



Figure 1: Histograms and classical kernel fit




(a) (b)

Figure 2: Histograms and classical kernel fit to log-transformed data

Optimal transformation parameters are obtained via expression (2). In section 2.3 we propose two methods for estimating the transformation parameters $\lambda = (\lambda_1, \lambda_2)$. The first is easier and only involves minimizing expression (5). We call this method "Method 1". The R code used to perform the minimization is shown in the Appendix under optimal transformation. Note the use of the 'optim' function in R.

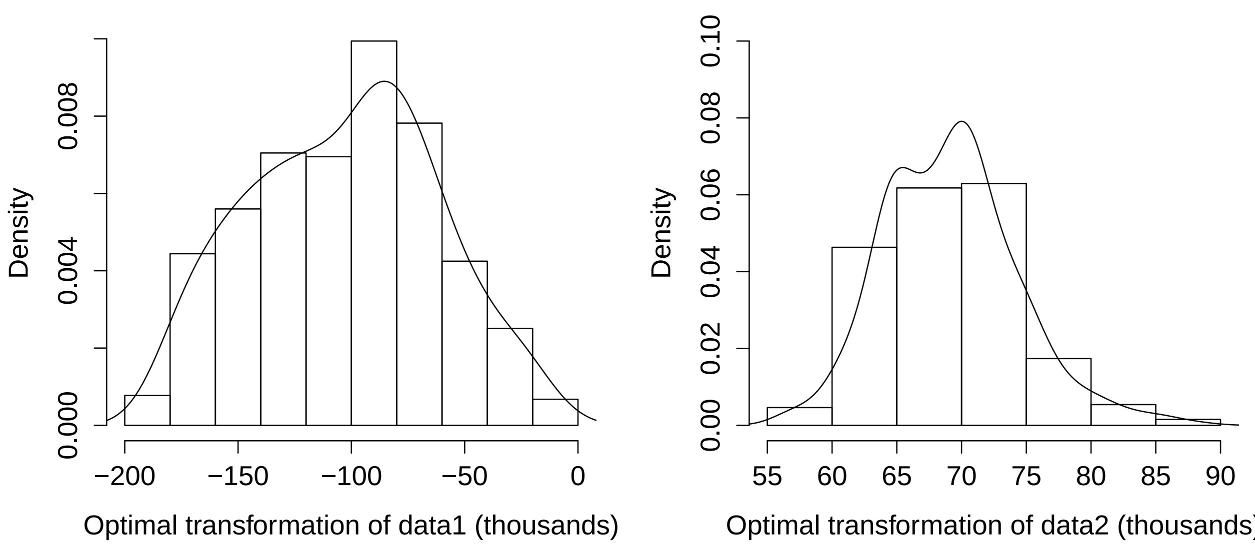
The second method which we propose to obtain the transformation parameters $\lambda = (\lambda_1, \lambda_2)$ needs to search within a set of transformation parameters that generate transformed variables with zero skewness, to look for the pair of parameters minimizing expression (5). We call this method "Method 2". The R code used to perform this algorithm is also given in the Appendix. Note the use of the 'optimize' function in R. Note that the 'optimize' function in R is used when we are finding a maximum or minimum value of a function subject to a constraint, such as here where we require the transformed data to have zero skewness.

In Table 2 the transformation estimates of parameters $\lambda = (\lambda_1, \lambda_2)$ are shown. The results using the two methods are similar; in fact, asymptotically, the two methods converge, because the density that minimizes the functional $\beta\left(f_Y''\right)$ is symmetric (see [21] and [20]). The differences between Method 1 and Method 2 are caused because $\beta\left(f_Y''\right)$ is unknown and an estimation in expression (5) must be used. Note also that for $X_2$ the values of $\lambda_1$ and $\lambda_2$ are near zero, this indicates that the distribution associated with the cost of medical expenses is very similar to a lognormal distribution.

Table 2: Estimates of transformation parameters $\lambda = (\lambda_1, \lambda_2)$

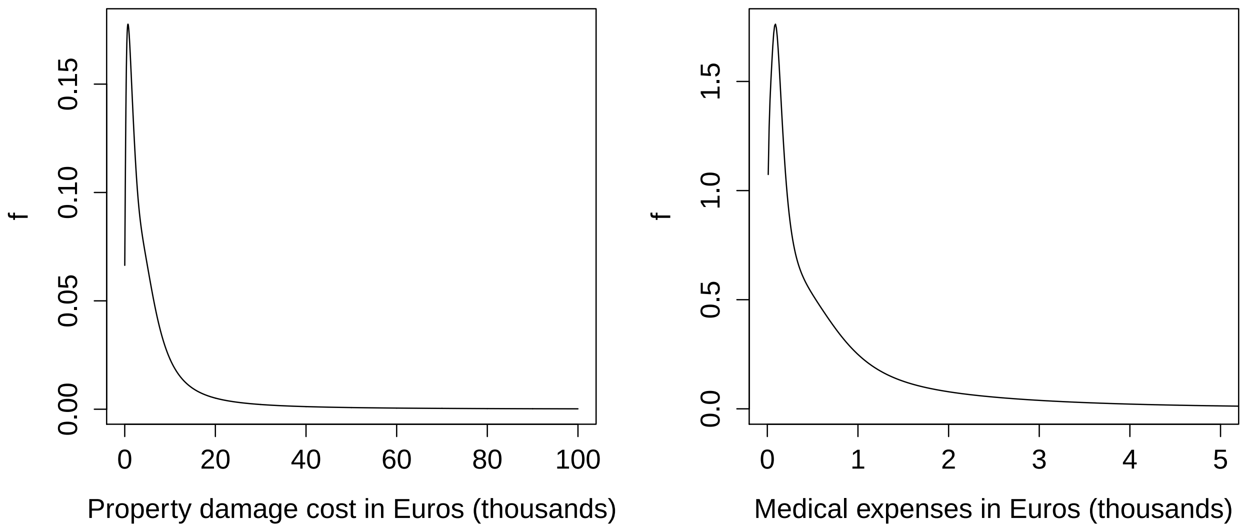
| | Method 1 | Method 2 |
|---|---|---|
| $X_1$ (data1) | $(1.9931, -0.6201)$ | $(1.8703, -0.5700)$ |
| $X_2$ (data2) | $(0.0219, -0.0054)$ | $(0.0041, 0.0500)$ |

In Figure 3 we show the kernel density estimates applied to the optimally transformed variable using Method 2 for both components of the univariate claims data. In Figure 4 we plot the TKE of pdf of property damage and medical expenses costs. We can see the smoothed shape of the pdf estimated in the right tail.




(a) (b)

Figure 3: Histograms and classical kernel fit to optimally transformed data




(a)

(b)

Figure 4: TKE pdf estimate

## 4.2 Goodness of fit results

Following the discussion about goodness of fit in Section 3, we calculate log-likelihoods and two different weighted log-likelihoods for each of the estimated models. Then we approximate $ISE_X$ using expression (6) and $WISE_X^1$ and $WISE_X^2$ using (7) and (8), respectively. In this way, we compare the non-parametric approaches. In order to store the results of these calculations in R, first we create three different R objects, namely lnL, w1lnL and w2lnL, and second we create ISE, w1ISE and w2ISE (see "Goodness of fit" subsection in the Appendix).

Note from the R code in the Appendix, that lnL, w1lnL and w2lnL have four rows and two columns. The four rows correspond to the four different estimators considered: classical kernel, transformed kernel density estimation using log-transformation and finally the transformed kernel density estimation using optimally transformed, where the optimal transformation is found as discussed in Section 2.3, initially minimizing only expression (5) (Method 1), and second, searching the optimum within a set of transformation parameters, where the transformed variable has zero skewness (Method 2). In the Appendix, we only show the R code for property damage (data1). For medical expenses (data2) the code is similar.

In Table 3 we show the results for the log-likelihood and weighted log-likelihood for the four fitted densities for the damage cost ($X_1$) and also for medical expenses cost ($X_2$). A higher value indicates a better fit. For classical kernel estimation, $\ln \hat{L}$ is clearly lower.

Results for the weighted log-likelihood ($\ln_{w^{(1)}} \hat{L}$ and $\ln_{w^{(2)}} \hat{L}$) show that the classical kernel is the method that provides the best fit once the tail of the distribution gains importance with the use of weights. However this is a distorted result, because, as we can see in Figures 1c and 1d, classical kernel is not smooth in the tail, so the fitted density in this zone forms little bubbles around the observed data points. Because of lack of smoothness $\ln_{w^{(1)}} \hat{L}$ and $\ln_{w^{(2)}} \hat{L}$ do not provide a net goodness of fit measure for classical kernel.

The transformed kernel density estimation is smooth in the tail of the distribution. If we compare the values of $\ln_{w^{(1)}} \hat{L}$ and $\ln_{w^{(2)}} \hat{L}$ for TKE with optimal transformation (Method 1 and Method 2) with the values for the log-transformation, we observe that the optimal transformation work better.

Log-likelihood and weighted log-likelihood are not good measures for comparing non-parametric fits. In section 3 we proposed the use of $CV$, $WCV_1$ and $WCV_2$ to compare the fit of classical kernel estimation and TKE. The results for damage cost $X_1$ and medical expenses cost $X_2$ are shown in Table 4. The lower the value, the better the fit. Note that the values of $CV$, $WCV_1$ and $WCV_2$ can be negative. The minimum values of $CV$, $WCV_1$ and $WCV_2$ for damage cost $X_1$ are found for TKE with Method 1 and Method 2. For medical expenses cost $X_2$ these minimum values are found for TKE with log-transformation. So, we can conclude that when the distribution has a medium tail, like $X_2$ in our case, the TKE with log-transformation is sufficient.

Table 3: Log-likelihood and weighted log-likelihoods

Damage cost ($X_1$)

| | $\ln \hat{L}$ | $\ln_{w^{(1)}} \hat{L}$ | $\ln_{w^{(2)}} \hat{L}$ |
|---|---|---|---|
| Classical Kernel | $-1998.07$ | $-3249.04$ | $-4770.33$ |
| Kernel Transformed (TKE with log transformation) | $-3340.97$ | $-5310.61$ | $-7837.46$ |
| Kernel Transformed (TKE with Method 1) | $-1574.85$ | $-3556.95$ | $-6002.81$ |
| Kernel Transformed (TKE with Method 2) | $-1574.09$ | $-3550.50$ | $-5996.93$ |

Medical expenses cost ($X_2$)

| | $\ln \hat{L}$ | $\ln_{w^{(1)}} \hat{L}$ | $\ln_{w^{(2)}} \hat{L}$ |
|---|---|---|---|
| Classical Kernel | $-980.41$ | $-2401.00$ | $-3604.68$ |
| Kernel Transformed (TKE with log transformation) | $-1191.66$ | $-3220.33$ | $-4778.72$ |
| Kernel Transformed (TKE with Method 1) | $-560.40$ | $-2587.88$ | $-4149.64$ |
| Kernel Transformed (TKE with Method 2) | $-558.90$ | $-2587.77$ | $-4153.43$ |

Table 4: Cross-valitation

Damage cost ($X_1$)

| | $CV$ | $WCV_1$ | $WCV_2$ |
|---|---|---|---|
| Classical Kernel | $-0.0360$ | $-0.1062$ | $-0.2726$ |
| Kernel Transformed (TKE with log transformation) | $-0.0254$ | $-0.0504$ | $3.2490$ |
| Kernel Transformed (TKE with Method 1) | $-0.0856$ | $-0.2207$ | $-1.2120$ |
| Kernel Transformed (TKE with Method 2) | $-0.0856$ | $-0.2207$ | $-1.2114$ |

Medical expenses cost ($X_2$)

| | $CV$ | $WCV_1$ | $WCV_2$ |
|---|---|---|---|
| Classical Kernel | $-0.2650$ | $-0.0985$ | $-0.0591$ |
| Kernel Transformed (TKE with log transformation) | $-1.2489$ | $-0.1730$ | $-0.1036$ |
| Kernel Transformed (TKE with Method 1) | $-0.7146$ | $-0.1948$ | $-0.1484$ |
| Kernel Transformed (TKE with Method 2) | $-0.7114$ | $-0.1943$ | $-0.1478$ |

# 5 Conclusions

In this paper we fitted univariate distributions to a real data set from motor insurance claim costs.

The kernel estimation approach provides a smoothed version of the empirical distribution. We also provided details of goodness of fit criteria based on standard likelihood theory and also using weighted likelihoods where greater weight is given to density estimation in the right tail of the distribution.

We can see that the value of the log-likelihood function is not a good method to compare non-parametric fits given that its value increase when the bandwidth $b$ decrease; thus we proposed alternative criteria based on the minimization of Integrated Square Error (ISE) and Weighted Integrated Square Error (WISE). Finally, we conclude that transformed kernel density estimation with a Shifted Power Transformation Family is a good alternative to fit distributions with heavy tails.

Statistical methods in insurance are increasingly being taught using statistical software to complement the theoretical developments of methods. R is freeware that is readily available and is being used widely in universities. The kernel density estimation are readily implemented using R. While obtaining estimates using R or other software is an important first step in the analysis of data, it is easy for students to overlook the importance of analysing descriptive statistics, data transformation and proper assessment of quality of fit of proposed models when doing analysis using real data. This paper provides an example of how all of these techniques can be applied to advantage when analysing motor vehicle insurance claim cost data. Class exercises or assignment exercises could be developed around the material presented in this paper. Depending on the level of students, different exercises and assignments could be based on the material here. For lower level students, replication of the results obtained using classical kernel density estimation could be developed. For intermediate level students, the benefits of transformation could be explored. More advanced students could be asked to develop some of the R code presented here, perhaps with an outline of the required code given to help get them started.

# References


[1] Bolancé, C. (2010) Optimal Inverse Beta(3,3) Transformation in kernel density estimation, SORT Statistics and Operations Research Transaction, 34, 223-238.

[2] Bolancé, C., Guillén, M. and Nielsen, J.P. (2009) Transformation kernel estimation of insurance claim cost distribution, in Corazza, M. and Pizzi, C. (Eds), Mathematical and Statistical Methods for Actuarial Sciences and Finance, Springer, Roma, 223-231.

[3] Bolancé, C., Guillén, M. and Nielsen, J.P., 2008. Inverse Beta transformation in kernel density estimation. Statistics & Probability Letters, 78, 1757-1764.

[4] Bolancé, C., Guillén, M., Pelican, E. and Vernic, R. (2008) Skewed bivariate models and nonparametric estimation for the CTE risk measure. Insurance: Mathematics and Economics, 43, 386-393.

[5] Bolancé, C., Guillén, M. and Nielsen, J.P., (2003) Kernel density estimation of actuarial loss functions, Insurance: Mathematics and Economics, 32, 19-36.

[6] Buch-Larsen, T., Guillen, M., Nielsen, J.P. and Bolancé, C. (2005) Kernel density estimation for heavy-tailed distributions using the Champernowne transformation. Statistics, 39, 503-518.

[7] Clements, A.E., Hurn, A.S. and Lindsay, K.A. (2003) Möbius-like mappings and their use in kernel density estimation, Journal of the American Statistical Association, 98, 993-1000.

[8] Denault, M. (2001) Coherent allocation of risk capital. Working paper. Ecole des H.E.C., Montreal.

[9] Dhaene, J., Goovaerts, M.J. and Kaas, R. (2003) Economic capital allocation derived from risk measures, North American Actuarial Journal 7, 44-59.

[10] Hall, P. and Marron J.S. (1987) Estimation of integrated squared density derivatives, Statistics & Probability Letters, 6, 100-115.

[11] Klugman, S.A., Panjer, H.H, and Willmot, G.E., 2008. Loss models: from data to decisions. 3rd Edition. John Wiley & Sons, New Jersey.

[12] Liu, Q., Pitt, D., Zhang, X. and Wu, X. (2011) A Bayesian approach to parameter estimation for kernel density estimation via transformation, Annals of Actuarial Science, 5, 181-193.

[13] McNeil, A.J., Frey. R. and Embrechts, P. (2005) Quantitative risk management: concepts, techniques and tools. Princeton University Press., Princeton Series in Finance, London.

[14] Panjer, H.H. (2002) Measurement of risk, solvency requirements and allocation of capital within financial conglomerates. 27th International Congress of Actuaries, Cancun 2002. (see also http://www.actuaries.org/events/congresses/Cancun/afir_subject/afir_14_panjer.pdf)

[15] Park, U.B. and Marron, J.S. (1990) Comparison of data-driven bandwidth selectors, Journal of the American Statistical Association, 8, 66-72.

[16] Reiss, R.D., 1981. Nonparametric estimation of smooth distribution functions. Scandinavian Journal of Statistics 8, 116-119.

[17] Scott, D.W., 1992. Multivariate Density Estimation. Theory, Practice and Visualization. John Wiley & Sons, Inc.

[18] Sheather, S.J. and Jones, M.C. (1991) A reliable data-based bandwidth selection method for kernel density estimation, Journal of the Royal Statistical Society, Serial B, 53, 683-690.

[19] Silverman, B.W. (1986) Density Estimation for Statistics and Data Analysis, Chapman & Hall, London.

[20] Terrell, G.R. (1990) The maximal smoothing principle in density estimation, Journal of the American Statistical Association, 85, 270-277.

[21] Terrell, G.R. and Scott, D.W. (1985) Oversmoothed nonparametric density estimates, Journal of the American Statistical Association, 80, 209-214.

[22] Valdez, E.A. and Chernih, A., 2003. Wang's capital allocation formula for elliptically contoured distributions. Insurance: Mathematics and Economics 33, 517-532.

[23] Vernic, R. 2006. Multivariate Skew-Normal distributions with applications in insurance. Insurance: Mathematics and Economics 38, 413-426.

[24] Wand, M.P. and Jones, M.C. (1995) Kernel Smoothing. Chapman & Hall, London.

[25] Wand, P., Marron, J.S. and Ruppert, D. (1991) Transformations in density estimation. Journal of the American Statistical Association, 86, 343-361.

[26] Wang, S. (2002) A set of new methods and tools for enterprise risk capital management and portfolio optimization. Working paper. SCOR reinsurance company (http://www.casact.com/pubs/forum/02sforum/02sf043.pdf).

[27] Wu, T.-J., Chen, C.-F. and Chen, H.-Y., 2007. A variable bandwidth selector in multivariate kernel density estimation. Statistics & Probability Letters 77, 462-467.

[28] Zhang, X., King, M.L. and Hyndman, R.J., 2006. A Bayesian approach to bandwidth selection for multivariate kernel density estimation. Computational Statistics & Data Analysis 50, 3009-3031.

# Appendix

## Descriptive statistics

```
cost <- read.table("KEURcostes.txt",header=TRUE)

data1 <- cost[,1]
data2 <- cost[,2]

# Descriptive statistics
library(QRM)

data <- data1 # -> choose dataset
data<-as.matrix(data)
n <- nrow(data)
colMeans(data) #Mean of variable
var(data) # Note that sample variance divides by n-1
sd(data) # Standard deviation
min(data)
max(data)
sum(data)

#Here it is necessary to load QRM library
skewness(data)
kurtosis(data)
```

## Kernel density estimation

```
# 1. Kernel density estimates. Univariate, Gaussian kernel and
#rule-of-thumb bandwidth calculated using standard deviation as dispersion parameter.
sd <- sd(data)
bw1<-1.059*sd*n^(-1/5)

dens1=density(data,bw=bw1,kernel="gaussian")

#Histogram of property damage cost data with overlaying kernel density estimate
hist(data,xlab="Property damage cost in Euros (thousands)",freq=FALSE,
      main=NULL,ylim=c(0,0.03))
lines(dens1)

#Histogram of property damage cost data with overlaying kernel density estimate
# for right tail
```

```
hist(data, xlab="Property damage cost in Euros (thousands)",freq=FALSE,
      xlim=c(200,1000), ylim=c(0,0.0005), main=NULL)
lines(dens1)

# 2. Kernel density estimates of log-transformation. Univariate, Gaussian kernel
#rule-of-thumb bandwidth calculated using standard deviation as dispersion parameter.

ln.data=log(data)

sdt<-sd(ln.data)
ln.data<-(sd/sdt)*ln.data
ln.dens1=density(ln.data,bw=bw1,kernel="gaussian")

#Histogram of log of property damage cost data with overlaying kernel
#density estimate
hist(ln.data, xlab="Log of Property damage cost in Euros (thousands)",
    freq=FALSE,main=NULL)
lines(ln.dens1)
```

## Optimal transformation

```
#METHOD 1
# Gaussian kernel, we define function ker
ker<-function(x){return(dnorm(x,0,1))}
# We define the transformation function
transf<-function(ll1,ll2,n,x)
{
  yd=(x+(ll1*rep(1,n)))
  if (ll2!=0)
    y=sign(ll2)*(yd^ll2)
  else
    y=log(yd)
  return(y)
}
# We define the derivative of the transformation function
dtransf<-function(ll1,ll2,n,x)
{
  yc=(x+(ll1*rep(1,n)))
  if (ll2!=0)
    gy=sign(ll2)*ll2*(yc^(ll2-1))
  else
    gy=1/yc
  return(gy)
```

```
}
# We define expression (5) to be minimized
beta<-function(ll)
{
  ll1=ll[1]
  ll2=ll[2]
  y<-transf(ll1,ll2,n,x)
  sy=sd(y)
  sx=sd(x)
  scal=sx/sy
  yscal=scal*y
  an=sx*((21/(8*sqrt(2)*5*n*n))^(1/13))
  ssgaus=0
  for (i in 2:n)
  {
    y1=yscal[i-1]
    n1=n-(i-1)
    y1=y1*rep(1,n1)
    y2=yscal[i:n]
    t=(y1-y2)/an;
    bgaus=dnorm(0,0,1)*(3/(2*sqrt(2)*(n*(n-1))*(an^5)))*
      sum((rep(1,n1)-(t^2)+((1/12)*(t^4)))*
            exp(-(1/4)*(t^2)))
    ssgaus=ssgaus+bgaus
  }
  ssgaus=ssgaus^(1/5)
  return(ssgaus)
}
lambda=rep(0,2)
dim(lambda)=c(1,2)
ll1=0.5
ll2=0.5

# Search for optimal parameters
x<-data
n<-nrow(x)
betaopt=optim(c(ll1,ll2),beta)
lambda[1,1]=betaopt$par[1]
lambda[1,2]=betaopt$par[2]
#we print the optimal parameters
lambda
```

```
#METHOD 2
# We define expression of squared skewness to be minimized
sk3<-function(ll1)
{y<-transf(ll1,ll2,n,x)
 sy=sd(y)
 sx=sd(x)
 scal=sx/sy
 yscal=scal*y
 a=sum(yscal^3);
 bb=sum(yscal^2);
 c=sum(yscal);
 ff=(a+((2*(c^3))/n^2)-((3*bb*c)/n))^2;
 ff=ff/(sx^6)
 return(ff)
}
nm=4000
grid=rep(0,nm)
dim(grid)=c((nm/4),4)
for(i in 1:(nm/4)){
  grid[i,2]=-3+0.01*(i-1)
}
# Search for optimal parameters
x<-data
n<-nrow(x)
for(i in 1:(nm/4)){
  ll1=0
  ll2=grid[i,2]
  skopt=optimize(sk3,c(-min(x)+0.01,1000))
  grid[i,1]=skopt$minimum
  grid[i,3]=skopt$objective
}
for(i in 1:(nm/4)){
  grid[i,4]=beta(c(grid[i,1],grid[i,2]))
}
ll1=grid[which.min(grid[,4]),1]
ll2=grid[which.min(grid[,4]),2]
ll1
ll2
```

## Transformed kernel density estimation

```
# We define the transformation function
transf<-function(ll1,ll2,n,x)
```

```
{
  yd=(x+(ll1*rep(1,n)))
  if (ll2!=0)
    y=sign(ll2)*(yd^ll2)
  else
    y=log(yd)
  return(y)
}
# We define the derivative of the transformation function
dtransf<-function(ll1,ll2,n,x)
{
  yc=(x+(ll1*rep(1,n)))
  if (ll2!=0)
    gy=sign(ll2)*ll2*(yc^(ll2-1))
  else
    gy=1/yc
  return(gy)
}

###############################################
# Calculate TKE of pdf for property damage cost
# TRANSFORMATION PARAMETERS ARE REQUIRED

#lambda method1 estimation
l1=1.993066
l2=-0.620136

grid<-as.matrix((1:10000)/100)
ng<-nrow(grid)
fkt<-as.matrix(rep(0,ng))
x<-data
y=transf(l1,l2,n,x)
tgrid=transf(l1,l2,ng,grid)
fkt<-as.matrix(rep(0,ng))
sx=sd(x)
hnt=bw1
sy=sd(y)
yscal=(sx/sy)*y
dy=dtransf(l1,l2,n,x)
dyscal=(sx/sy)*dy
tgscal=(sx/sy)*tgrid
dtg=dtransf(l1,l2,ng,grid)
dtgscal=(sx/sy)*dtg
```

```
for (i in 1:ng) {dif=(tgscal[i,]-yscal)/hnt
                 fkt[i,]=(dtgscal[i,]/(n*hnt))*sum(ker(dif))}

#Plot of TKE of pdf figure 4
plot(grid,fkt, type="l",xlab="Property damage cost in Euros (thousands)",
    ylab="f", main=NULL)

#Caculate Silverman rule-of-thumb bandwidth for optimal transformed data
# Using lambda method 1 estimations

#l1, l2   -> method 1
#ll1, ll2 -> method 2

# First, using method1 estimations:
l1.1=l1
l2.1=l2
op.data<-transf(l1.1,l2.1,n,data)

sopx1<-sd(op.data)

op.data<-(sx/sopx1)*op.data

bw.op1<-bw1
bw.op1

op.dens1=density(op.data,bw=bw.op1,kernel="gaussian")
op.dens1

#Histogram of optimal transformation data with
#overlaying kernel density estimate figure 3
hist(op.data, main="",xlab="Optimal transformation",freq=FALSE)
lines(op.dens1)

# Using method2 estimations:

l1.1=ll1
l2.1=ll2
op.data<-transf(l1.1,l2.1,n,data)

sopx1<-sd(op.data)

op.data<-(sx/sopx1)*op.data
```

```
bw.op1<-bw1
bw.op1

op.dens1=density(op.data,bw=bw.op1,kernel="gaussian")
op.dens1

#Histogram of optimal transformation of data with
#overlaying kernel density estimate figure 5 with method 2
hist(op.data, main="",xlab="Optimal transformation of data1 (thousands)",
    freq=FALSE)
lines(op.dens1)
```

## Goodness of fit

```
lnL=rep(0,4)
w1lnL=rep(0,4)
w2lnL=rep(0,4)
dim(lnL)=c(4,1)
dim(w1lnL)=c(4,1)
dim(w2lnL)=c(4,1)


#Next we give the R code used to calculate the log-likelihood and
#both versions of the weighted log-likelihood for the density estimate obtained
#by applying the classical kernel to the non-transformed data.

#Calculation of log-likelihood for univariate kernel density estimate applied
#to non-transformed data for both components of claim

n <- nrow(data)
dens1val=c(rep(0,n))

for(i in 1:n){dens1val[i]=1/(n*dens1$bw)*sum(dnorm((data[i]-data)/dens1$bw),0,1)}
lnL[1,1]=sum(log(dens1val))

#Calculation of weighted log-likelihood for univariate kernel density estimate
#applied to non-transformed data for both components of claim
#(weights=claim size)

w1dens1val=c(rep(0,n))
for(i in 1:n){w1dens1val[i]=n*data[i]*log(1/(n*dens1$bw)*
  sum(dnorm((data[i]-data)/dens1$bw,0,1)))/sum(data)}
w1lnL[1,1]=sum(w1dens1val)
```

```
#Calculation of weighted log-likelihood for univariate kernel density estimate
#applied to non-transformed data for both components of claim
#(weights=claim size^2)

w2dens1val=c(rep(0,n))
for(i in 1:n){w2dens1val[i]=n*(data[i])^2*log(1/(n*dens1$bw)*
                          sum(dnorm((data[i]-data)/dens1$bw,0,1)))/sum((data)^2)}
w2lnL[1,1]=sum(w2dens1val)
```

Below, we provide the R code used to calculate the log-likelihood and both versions of the weighted log-likelihood for the density estimate obtained by applying the classical kernel to the log-transformed data. Calculation of log-likelihood for univariate kernel density estimate obtained from log-transformed data for both components of claim

```
ln.dens1val=c(rep(0,n))
for(i in 1:n){ln.dens1val[i]=1/(n*data[i]*ln.dens1$bw)*
    sum(dnorm((ln.data[i]-ln.data)/ln.dens1$bw,0,1))}
lnL[2,1]=sum(log(ln.dens1val))
#Calculation of weighted log-likelihood for univariate kernel density estimate
#obtained from log-transformed for both components of claim
#(weights=claim size)

w1ln.dens1val=c(rep(0,n))
for(i in 1:n){w1ln.dens1val[i]=n*data[i]*log(1/(n*data[i]*ln.dens1$bw)*
    sum(dnorm((ln.data[i]-ln.data)/ln.dens1$bw,0,1)))/sum(data)}
w1lnL[2,1]=sum(w1ln.dens1val)
#Calculation of weighted log-likelihood for univariate kernel density estimate
#obtained from log-transformed for both components of claim
#(weights=claim size^2)
w2ln.dens1val=c(rep(0,n))
for(i in 1:n){w2ln.dens1val[i]=n*(data[i])^2*log(1/(n*data[i]*ln.dens1$bw)*
    sum(dnorm((ln.data[i]-ln.data)/ln.dens1$bw,0,1)))/sum((data)^2)}
w2lnL[2,1]=sum(w2ln.dens1val)
```

Now we give the R code used to obtain the log-likelihood and both forms of the weighted log-likelihood for the kernel density estimate obtained by applying the optimal transformation parameters, to our data. In this case, it is necessary to write the values of transformation parameters:

ll1 (lambda1) and ll2 (lambda2).

```
#Calculate shifted power transformed kernel density estimation
#(given optimal lambdas) and log-likelihoods for cost 1
```

```
# lambda estimated in method 1:
ll1=1.993066
ll2=-0.620136
x<-data
fkt=as.matrix(rep(0,n))
sx=sd(x)
hnt=1.059*sx*((1/n)^(1/5))
y=transf(ll1,ll2,n,x)
sy=sd(y)
yscal=(sx/sy)*y
gy=dtransf(ll1,ll2,n,x)
gyscal=(sx/sy)*gy
for (i in 1:(n-1))
{
  vecy<-as.matrix((yscal[1:i]-yscal[(n-i+1):n])/hnt)[1:i]
  newvecy=ker(vecy)
  auxy=c(as.matrix(newvecy),as.matrix(rep(0,(n-i))))
  aux2y=c(as.matrix(rep(0,(n-i))),as.matrix(newvecy))
  fkt=fkt+(auxy+aux2y)
}
k0=dnorm(0,0,1)
fkt=fkt+rep(k0,n)
fkt=(gyscal*fkt)/(n*hnt)

#Calculation of log-likelihood for transformation kernel density estimate
tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  tkd.dens1val[i]=log(fkt[i])}
lnL[3,1]=sum(tkd.dens1val)

#Calculation of weighted log-likelihood for transformation kernel density
#estimate (weights=claim size)
w1tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  w1tkd.dens1val[i]=n*x[i]*log(fkt[i])}
w1lnL[3,1]=sum(w1tkd.dens1val)/(sum(x))

#Calculation of weighted log-likelihood for transformation kernel density
#estimate (weights=claim size^2)
w2tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  w2tkd.dens1val[i]=n*x[i]*x[i]*log(fkt[i])}
w2lnL[3,1]=sum(w2tkd.dens1val)/(sum(x^2))
```

```
# lambda estimated in method 2:
ll1=1.870333
ll2=-0.57
x<-data
fkt=as.matrix(rep(0,n))
sx=sd(x)
hnt=1.059*sx*((1/n)^(1/5))
y=transf(ll1,ll2,n,x)
sy=sd(y)
yscal=(sx/sy)*y
gy=dtransf(ll1,ll2,n,x)
gyscal=(sx/sy)*gy
for (i in 1:(n-1))
{
  vecy<-as.matrix((yscal[1:i]-yscal[(n-i+1):n])/hnt)[1:i]
  newvecy=ker(vecy)
  auxy=c(as.matrix(newvecy),as.matrix(rep(0,(n-i))))
  aux2y=c(as.matrix(rep(0,(n-i))),as.matrix(newvecy))
  fkt=fkt+(auxy+aux2y)
}
k0=dnorm(0,0,1)
fkt=fkt+rep(k0,n)
fkt=(gyscal*fkt)/(n*hnt)

#Calculation of log-likelihood for transformation kernel density estimate
tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  tkd.dens1val[i]=log(fkt[i])}
lnL[4,1]=sum(tkd.dens1val)

#Calculation of weighted log-likelihood for transformation kernel density
#estimate (weights=claim size)
w1tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  w1tkd.dens1val[i]=n*x[i]*log(fkt[i])}
w1lnL[4,1]=sum(w1tkd.dens1val)/(sum(x))

#Calculation of weighted log-likelihood for transformation kernel density
#estimate (weights=claim size^2)
w2tkd.dens1val=c(rep(0,n))
for(i in 1:n){
  w2tkd.dens1val[i]=n*x[i]*x[i]*log(fkt[i])}
```

```
w2lnL[4,1]=sum(w2tkd.dens1val)/(sum(x^2))

rownames(lnL) <- c("Normal","Log-normal","Classical Kernel",
                   "TKE log transf.","TKE method1","TKE method2")
rownames(w1lnL) <- c("Normal","Log-normal","Classical Kernel",
                   "TKE log transf.","TKE method1","TKE method2")
rownames(w2lnL) <- c("Normal","Log-normal","Classical Kernel",
                   "TKE log transf.","TKE method1","TKE method2")

likeli<-rep(0,12)
dim(likeli)<-c(4,3)

likeli[,1] <- lnL
likeli[,2] <- w1lnL
likeli[,3] <- w2lnL

colnames(likeli) <- c("lnL","w1lnL","w2lnL")
rownames(likeli) <- c("Classical kernel",
    "TKE log trans","TKE Method1","TKE Method2")

round(likeli,2)
```

Newt we give the R code used to calculate the ISE and and both versions of the weighted ISE for the density estimate obtained by applying the classical kernel to the non-transformed data.

```
# 1.1 Calculation ISE for classical kernels estimations cost1

x<-data
ise<-rep(0,12)
dim(ise)<-c(4,3)
grid<-as.matrix((1:500000)/100)
ng=nrow(grid)
ng
fk<-as.matrix(rep(0,ng))
bw1<-dens1$bw
for (i in 1:ng) {dif=(grid[i,]-x)/bw1
                 fk[i,]=(1/(n*bw1))*sum(ker(dif))}
first<-sum((fk^2)*0.01)
first
fk_i<-as.matrix(rep(0,n))
```

```
dim(fk_i)<-c(n,1)
fk_i[1,]<-(1/(n*bw1))*sum(ker((x[1,]-x[2:n,])/bw1))
fk_i[n,]<-(1/(n*bw1))*sum(ker((x[n,]-x[1:(n-1),])/bw1))
for (i in 2:(n-1)) {
  fk_i[i,]<-(1/(n*bw1))*(sum(ker((x[i,]-x[1:(i-1),])/bw1))
                        +sum(ker((x[i,]-x[(i+1):n,])/bw1)))}
second<-sum(fk_i)/n
second
isek<-first-2*second
isek
ise[1,1]<-isek

# 1.2 Calculation WISE1 for classical kernels estimations cost1
first<-sum((fk^2)*grid*0.01)
first
second<-sum(fk_i*x)/n
second
w1isek<-first-2*second
w1isek
ise[1,2]<-w1isek

# 1.3 Calculation WISE2 for classical kernels estimations cost1
first<-sum((fk^2)*(grid^2)*0.01)
first
second<-sum(fk_i*(x^2))/n
second
w2isek<-first-2*second
w2isek
ise[1,3]<-w2isek
```

Below, we provide the R code used to calculate the ISE and both versions of the weighted ISE for the density estimate obtained by applying the classical kernel to the log-transformed data.

```
# 2.1 Calculation ISE for log-transformed kernels estimations
x<-data
y<-log(x)
lgrid=log(grid)
fkt<-as.matrix(rep(0,ng))
bw.ln1<-ln.dens1$bw
for (i in 1:ng) {dif=(lgrid[i,]-y)/bw.ln1
                 fkt[i,]=(1/(n*bw.ln1*grid[i,]))*sum(ker(dif))}
first<-sum((fk^2)*0.01)
```

```
first
fk_i<-as.matrix(rep(0,n))
dim(fk_i)<-c(n,1)
fk_i[1,]<-(1/(n*bw.ln1*x[1,]))*sum(ker((y[1,]-y[2:n,])/bw.ln1))
fk_i[n,]<-(1/(n*bw.ln1*x[n,]))*sum(ker((y[n,]-y[1:(n-1),])/bw.ln1))
for (i in 2:(n-1)) {
  fk_i[i,]<-(1/(n*bw.ln1*x[i,]))*(sum(ker((y[i,]-y[1:(i-
                                                        1),])/bw.ln1))
                                  +sum(ker((y[i,]-y[(i+1):n,])/bw.ln1)))}
second<-sum(fk_i)/n
second
isek<-first-2*second
isek
ise[2,1]<-isek

# 2.2 Calculation WISE1 for log-transformed kernels estimations cost1
first<-sum((fkt^2)*grid*0.01)
first
second<-sum(fk_i*x)/n
second
w1isek<-first-2*second
w1isek
ise[2,2]<-w1isek
#Calculation WISE2 for log-transformed kernels estimations cost1
first<-sum((fkt^2)*(grid^2)*0.01)
first
second<-sum(fk_i*(x^2))/n
second
w2isek<-first-2*second
w2isek
ise[2,3]<-w2isek
```

Now we give the R code used to obtain the ISE and both forms of the weighted ISE for the kernel density estimate obtained by applying the optimal transformation parameters, to our data. In this case, it is necessary to write the values of transformation parameters: ll1 (lambda1) and ll2 (lambda2).

```
#Calculation ISE for shifted power transformed kernels estimations
#(given optimal lambdas) !!!

x<-data

# 3.1 lambdas method1
```

```
ll1=1.993066
ll2=-0.620136
y=transf(ll1,ll2,n,x)
tgrid=transf(ll1,ll2,ng,grid)
fkt<-as.matrix(rep(0,ng))
sx=sd(x)
hnt=1.059*sx*((1/n)^(1/5))
sy=sd(y)
yscal=(sx/sy)*y
dy=dtransf(ll1,ll2,n,x)
dyscal=(sx/sy)*dy
tgscal=(sx/sy)*tgrid
dtg=dtransf(ll1,ll2,ng,grid)
dtgscal=(sx/sy)*dtg
for (i in 1:ng) {dif=(tgscal[i,]-yscal)/hnt
                 fkt[i,]=(dtgscal[i,]/(n*hnt))*sum(ker(dif))}
first<-sum((fkt^2)*0.01)
first
fk_i<-as.matrix(rep(0,n))
dim(fk_i)<-c(n,1)
fk_i[1,]<-(dyscal[1,]/(n*hnt))*sum(ker((yscal[1,]-
                                          yscal[2:n,])/hnt))
fk_i[n,]<-(dyscal[n,]/(n*hnt))*sum(ker((yscal[n,]-yscal[1:(n-
                                                          1),])/hnt))
for (i in 2:(n-1)) {
  fk_i[i,]<-(dyscal[i,]/(n*hnt))*(sum(ker((yscal[i,]-yscal[1:(i-
                                                             1),])/hnt))
                                  +sum(ker((yscal[i,]-yscal[(i+1):n,])/hnt)))}
second<-sum(fk_i)/n
second
isek<-first-2*second
isek
ise[3,1]<-isek

# 3.2 Calculation WISE1 for shifted power transformed kernels estimations
# (given optimal lambdas) of cost1
first<-sum((fkt^2)*grid*0.01)
first
second<-sum(fk_i*x)/n
second
w1isek<-first-2*second
w1isek
ise[3,2]<-w1isek
```

```
# 3.3 Calculation WISE2 for shifted power transformed kernels estimations
#(given optimal lambdas) of cost1
first<-sum((fkt^2)*(grid^2)*0.01)
first
second<-sum(fk_i*(x^2))/n
second
w2isek<-first-2*second
w2isek
ise[3,3]<-w2isek

#lambdas method2

# 4.1 lambdas method2
ll1=1.870333
ll2=-0.57
y=transf(ll1,ll2,n,x)
tgrid=transf(ll1,ll2,ng,grid)
fkt<-as.matrix(rep(0,ng))
sx=sd(x)
hnt=1.059*sx*((1/n)^(1/5))
sy=sd(y)
yscal=(sx/sy)*y
dy=dtransf(ll1,ll2,n,x)
dyscal=(sx/sy)*dy
tgscal=(sx/sy)*tgrid
dtg=dtransf(ll1,ll2,ng,grid)
dtgscal=(sx/sy)*dtg
for (i in 1:ng) {dif=(tgscal[i,]-yscal)/hnt
                 fkt[i,]=(dtgscal[i,]/(n*hnt))*sum(ker(dif))}
first<-sum((fkt^2)*0.01)
first
fk_i<-as.matrix(rep(0,n))
dim(fk_i)<-c(n,1)
fk_i[1,]<-(dyscal[1,]/(n*hnt))*sum(ker((yscal[1,]-
                                          yscal[2:n,])/hnt))
fk_i[n,]<-(dyscal[n,]/(n*hnt))*sum(ker((yscal[n,]-yscal[1:(n-
                                                             1),])/hnt))
for (i in 2:(n-1)) {
  fk_i[i,]<-(dyscal[i,]/(n*hnt))*(sum(ker((yscal[i,]-yscal[1:(i-
                                                                1),])/hnt))
                                  +sum(ker((yscal[i,]-yscal[(i+1):n,])/hnt)))}
second<-sum(fk_i)/n
```

```
second
isek<-first-2*second
isek
ise[4,1]<-isek

# 4.2 Calculation WISE1 for shifted power transformed kernels estimations
#(given optimal lambdas) of cost1
first<-sum((fkt^2)*grid*0.01)
first
second<-sum(fk_i*x)/n
second
w1isek<-first-2*second
w1isek
ise[4,2]<-w1isek

# 4.3 Calculation WISE2 for shifted power transformed kernels estimations
#(given optimal lambdas) of cost1
first<-sum((fkt^2)*(grid^2)*0.01)
first
second<-sum(fk_i*(x^2))/n
second
w2isek<-first-2*second
w2isek
ise[4,3]<-w2isek

colnames(ise) <- c("CV","WCV1","WCV2")
rownames(ise) <- c("Classical Kernel","TKE log-trans","TKE Method1","TKE Method2")

round(ise,4)
```

## UB·Riskcenter Working Paper Series

### List of Published Working Papers

**[WP 2014/01].** Bolancé, C., Guillén, M. and Pitt, D. (2014) "Non-parametric models for univariate claim severity distributions – an approach using R", UB Riskcenter Working Papers Series 2014-01.