boxcoxmix: An R Package for Response Transformations for Random Effect and Variance Component Models

Amani Almohaimeed and Jochen Einbeck
Qassim University and Durham University

Abstract

Random effect models have become a mainstream statistical technique over the last decades, and the same can be said for response transformation models such as the Box-Cox transformation. The latter ensures that the assumptions of normality and of homoscedasticity of the response distribution are fulfilled, which are essential conditions for the use of a linear model or a linear mixed model. However, methodology for response transformation and *simultaneous* inclusion of random effects has been developed and implemented only scarcely, and is so far restricted to Gaussian random effects. In this vignette, we introduce a new **R** package, **boxcoxmix**, that aims to ensure the validity of a normal response distribution using the Box-Cox power transformation in the presence of random effects, thereby not requiring parametric assumptions on their distribution. This is achieved by extending the "Nonparametric Maximum Likelihood" towards a "Nonparametric Profile Maximum Likelihood" technique. The implemented techniques allow to deal with overdispersion as well as two–level data scenarios.

Keywords: Box–Cox transformation, mixed model, nonparametric maximum likelihood, EM algorithm.

1. Introduction

In regression analysis, the data needs to achieve normality and homoscedasticity of the response distribution in order to enable access to linear model theory and associated inferential tools such as confidence intervals and hypothesis tests. This often requires transforming the response variable. Box and Cox (1964) proposed a parametric power transformation technique for transforming the response in univariate linear models. This transformation has been intensively studied by many researchers. Sakia (1992) briefly reviewed the work relating to this transformation. Solomon (1985) studied the application of the Box-Cox transformations to simple variance component models. The extension of the transformation to the linear mixed effects model was proposed by Gurka, Edwards, Muller, and Kupper (2006), in the case of a Gaussian random effect distribution. An obvious concern of assuming a normal random effect distribution is whether there are any harmful effects of misspecification. Bock and Aitkin (1981) showed that there is no need to make an assumption about the distribution of the random effects and it can be estimated as a discrete mixing distribution. Aitkin

(1996), Heckman and Singer (1984) and Davies (1987) showed that the parameter estimation is sensitive to the choice of the mixing distribution specification. The problem of estimating the mixing distribution using a specific parametric form (e.g. normal) can be overcome by the use of non-parametric maximum likelihood (NPML) estimation; the NPML estimate of the mixing distribution is known to be a discrete distribution involving a finite number of mass-points and corresponding masses (Laird 1978; Lindsay *et al.* 1983). An Expectation-Maximization (EM) algorithm is used for fitting the finite mixture distribution, each iteration of this algorithm is based on two steps: the expectation step (E-step) and the maximization step (M-step); see Aitkin (1999); Aitkin, Francis, Hinde, and Darnell (2009) and Einbeck, Hinde, and Darnell (2007) for details. The maximum likelihood (ML) estimate via the EM algorithm is a preferable approach due to its generality and simplicity; when the underlying complete data come from an exponential family whose ML estimates are easily computed, then each maximization step of an EM algorithm is likewise easily computed (Dempster, Laird, and Rubin 1977). For both overdispersed and variance component models, the EM algorithm for NPML estimation of the mixing distribution was regarded as "very stable and converged in every case" (Aitkin 1999).

A particular appealing aspect of the NPML approach is that the posterior probability that a certain unit belongs to a certain cluster corresponds to the weights in the final iteration of the EM algorithm (Sofroniou, Einbeck, and Hinde 2006). Another benefit of this approach is that increasing the number of mass points requires little computational effort and that the mass–points are not restricted to lie on a grid (Aitkin 1996). Aitkin concluded that "the simplicity and generality of the non–parametric model and the EM algorithm for full NPML estimation in overdispersed exponential family models make them a powerful modelling tool". The ability of the EM algorithm to locate the global maximum in fewer iterations can be affected by the choice of initial values; several methods for choosing initial values for the EM algorithm in the case of finite mixtures are discussed by Karlis and Xekalaki (2003). A grid search for setting the initial values was suggested by Laird (1978). Hou, Mahnken, Gajewski, and Dunton (2011) found limited difference from subsequent test of structural effects if the factors with structural effects were omitted during the estimating process for the Box-Cox power transformation parameter. They noted that the Box-Cox transformation works better only if the cluster sizes are very large; and it is necessary to run a grid search of the transformation in order to determine the parameter estimate that maximizes the residual (or profile) likelihood during the optimization process both under the linear and the mixed model settings. Nawata (1994) proposes a scanning Maximum likelihood method. Basically one conducts the entire methodology on a grid of fixed values of the transformation parameter λ and then optimizes over this grid. Nawata *et al.* (2013) used this method to calculate the maximum likelihood estimator of the Box-Cox transformation model. Gurka *et al.* (2006) noted that it is necessary to discuss how the estimation of λ affects inference about the other model parameters when one extends the Box-Cox transformation to the linear mixed model.

This vignette introduces (an implementation of) a transformation approach by extending the Box-Cox transformation to overdispersion and two–level variance component models. It aims to ensure the validity of a normal response distribution using the Box–Cox power transfor-

mation in the presence of random effects, thereby not requiring parametric assumptions on their distribution. This is achieved by extending the "Nonparametric Maximum Likelihood" towards a "Nonparametric Profile Maximum Likelihood (NPPML)" technique. To the best of our knowledge, the approach turns out to be the only one of its kind that has implemented the Box-Cox power transformation of the linear mixed effects model with an unspecified random effect distribution.

For an existing implementation of the Box-Cox transformation for the univariate linear model in **R**, we mention the `boxcox()` function in the **MASS** package (Venables and Ripley 2002). Essentially, `boxcox()` calculates and plots the profile log-likelihood for the univariate linear model against a set of λ values, in order to locate the transformation parameter under which the log-likelihood is maximized (yielding, after transformation, data that follow a normal distribution more closely than the untransformed data). In turn, the NPML methodology is implemented in the **npmlreg** package (Aitkin *et al.* 2009; Einbeck, Darnell, and Hinde 2014), which provides functions `alldist()` and `allvc()` for simple overdispersion models and variance component models, respectively. In this article, we introduce the **boxcoxmix** package which can be considered as a comnination of the Box–Cox and NPML concepts and which implements transformation models for random effect and variance component models using the NPPML technique. The package is available from the Comprehensive **R** Archive Network (CRAN) at `https://cran.r-project.org/package=boxcoxmix`.

The remainder of the article is organized as follows. Section 2 begins by providing a general introduction to the Box-Cox transformation for the linear model, as well as the theory and methodology underlying random effect models with unspecified random effect distribution. It proceeds with using the "Nonparametric Profile Maximum Likelihood" technique to combine these two methods. It also explains the basic usages of **boxcoxmix**'s main functions with a real data example. In Section 3, the Box–Cox transformation is extended to the two-level variance component model, along with some examples. The article concludes with a discussion in Section 4.

2. Box-Cox transformation in random effect models

2.1. Box-Cox transformation

The Box–Cox transformation (Box and Cox 1964) has been widely used in applied data analysis. The objective of the transformation is to select an appropriate parameter λ which is then used to transform data such that they follow a normal distribution more closely than the untransformed data. The transformation of the responses y_i, $i = 1, \ldots, n$, takes the form:

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & (\lambda \neq 0), \\ \log y_i & (\lambda = 0), \end{cases} \tag{1}$$

where the restriction $y_i > 0$ applies. The response variable transformed by the Box–Cox transformation is assumed to be linearly related to its covariates and the errors normally

distributed with constant variance.

2.2. Random effects

In the linear model, it is assumed that a set of explanatory variables $x_i, \quad i = 1, \ldots, n$, and a response variable y_i are linearly related such that $y_i = x_i^T\beta + \epsilon_i$ where ϵ_i is an error term which is usually assumed to be Gaussian and homescedastic. If the population from which the data are sampled consists of heterogeneous, unknown subpopulations, then the linear model described above will not fit well. In such cases, the presence of further unknown variability can be accommodated by adding a random effect z_i with density $g(z)$ to the linear predictor,

$$y_i = x_i^T\beta + z_i + \epsilon_i. \tag{2}$$

The responses y_i are independently distributed with mean function $E(y_i|z_i) = x_i^T\beta + z_i$, conditionally on the random effect z_i. Let $\phi(y; \cdot, \cdot)$ denote the univariate Gaussian probability density function, with mean and variance specified in the remaining two function arguments. The conditional probability density function of y_i given z_i is given by

$$f(y_i|z_i) = \phi(y_i; x_i^T\beta + z_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_i - x_i^T\beta - z_i)^2\right]. \tag{3}$$

Note that under the presence of a random effect, the parametric intercept term can be omitted from $x_i^T\beta$. Under the NPML estimation approach, the distribution of the random effect will be approximated by a discrete distribution at mass points $z_1, \ldots, z_K$, which can be considered as intercepts for the different unknown subgroups. This will be explained in detail in the following subsection, under inclusion of the Box–Cox transformation.

2.3. Extending the Box-Cox transformation to random effect models

In this section, the Box-Cox transformation is extended to the random effects model. In this case, it is assumed that there is a value of λ for which

$$y_i^{(\lambda)}|z_i \sim N(x_i^T\beta + z_i, \sigma^2), \tag{4}$$

where z_i is a random effect with an unspecified density $g(z_i)$. Taking account of the Jacobian of the transformation from y to $y^{(\lambda)}$, the conditional probability density function of y_i given z_i is

$$f(y_i|z_i) = \frac{y_i^{\lambda-1}}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_i^{(\lambda)} - x_i^T\beta - z_i)^2\right], \tag{5}$$

hence, the likelihood in relation to the original observations is

$$L(\lambda, \beta, \sigma^2, g) = \prod_{i=1}^{n} \int f(y_i|z_i)g(z_i)dz_i. \tag{6}$$

Under the non-parametric maximum likelihood (NPML) approach, the integral over the (unspecified) mixing distribution $g(z)$ is approximated by a discrete distribution on a finite number K of mass-points z_k, with masses π_k (Aitkin *et al.* 2009). The approximated likelihood is then

$$L(\lambda, \beta, \sigma^2, z_1, ..., z_k, \pi_1, ..., \pi_k) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_{ik} \tag{7}$$

where $f_{ik} = f(y_i|z_k)$. Defining indicators

$$G_{ik} = \begin{cases} 1 & \text{if observation } y_i \text{ comes from cluster } k, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

the complete likelihood would be

$$L^* = \prod_{i=1}^{n} \prod_{k=1}^{K} (\pi_k f_{ik})^{G_{ik}}, \tag{9}$$

so that the complete log-likelihood takes the shape

$$\ell^* = \log L^* = \sum_{i=1}^{n} \sum_{k=1}^{K} \left[G_{ik} \log \pi_k + G_{ik} \log f_{ik} \right]. \tag{10}$$

If $K = 1$, the log-likelihood would be the usual log-likelihood of the Box–Cox model without random effects.

We now apply the expectation-maximization (EM) approach to find the maximum likelihood estimate (MLE) of the model parameters. Given some starting values β^0, σ^0, z_k^0, and π_k^0 (discussed in a separate subsection below), set $\hat{\beta} = \beta^0$, $\hat{\sigma} = \sigma^0$, $\hat{z}_k = z_k^0$, $\hat{\pi}_k = \pi_k^0$, $k = 1, 2, \ldots, K$, and iterate between

E-step: Estimate G_{ik} by its expectation

$$w_{ik} = \frac{\hat{\pi}_k f_{ik}}{\sum_\ell \hat{\pi}_\ell f_{i\ell}} \tag{11}$$

which is the posterior probability that observation y_i comes from cluster k. Note that f_{ik} depends via equation (5) implicitly on the current values of $\hat{z}_k$, $\hat{\beta}$ and $\hat{\sigma}^2$.

M-step: The estimators $\hat{\beta}, \hat{\sigma}^2, \hat{z}_k$ and $\hat{\pi}_k$ can be obtained using the current w_{ik}, via the following four equations which were obtained through manual derivation of the NPML estimators for fixed K:

$$\hat{\beta} = \left(\sum_{i=1}^{n} x_i x_i^T \right)^{-1} \sum_{i=1}^{n} x_i \left(y_i^{(\lambda)} - \sum_{k=1}^{K} w_{ik} \hat{z}_k \right), \tag{12}$$

$$\hat{\sigma}^2 = \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{w_{ik} (y_i^{(\lambda)} - x^T \hat{\beta} - \hat{z}_k)^2}{n}, \tag{13}$$

$$\hat{z}_k = \frac{\sum_{i=1}^{n} w_{ik}(y_i^{(\lambda)} - x_i^T \hat{\beta})}{\sum_{i=1}^{n} w_{ik}}, \tag{14}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{n} w_{ik}}{n}. \tag{15}$$

We see from this that $\hat{\pi}_k$ is the average posterior probability for component k.

Replacing the results into Equation (7) we get the non-parametric profile likelihood function $L_P(\lambda)$, or its logarithmic version

$$\ell_P(\lambda) = \log\Big(\sum_{k=1}^{K} \hat{\pi}_k^{(\lambda)} \hat{f}_{ik}^{(\lambda)}\Big). \tag{16}$$

The non-parametric profile maximum likelihood (NPPML) estimator is therefore given by

$$\hat{\lambda} = arg \max_{\lambda} \ \ell_P(\lambda). \tag{17}$$

In practice, the EM–algorithm needs to be stopped after a certain number of iterations when it has reached its point of convergence. Polańska (2003) defined this convergence criterion as the absolute change in the successive log-likelihood function values being less than an arbitrary parameter such as $\delta = 0.0001$.

In package **boxcoxmix**, the main function for fitting random effect models with response transformations is `optim.boxcox()`, which performs a grid search of (16) over the parameter λ and then optimizes over this grid, in order to calculate the maximum likelihood estimator $\hat{\lambda}$ of the transformation. It produces a plot of the non-parametric profile likelihood function that summarises information concerning λ, including a vertical line indicating the best value of λ that maximizes the non–parametric profile log–likelihood. In order to fit models with fixed value of λ, one can use function `np.boxcoxmix()`. When λ =1 (no transformation), the results of the proposed approach will be very similar to that of the **npmlreg** function `alldist()`. However, the function `np.boxcoxmix()` is not a copy or extension of the `alldist()` function; the implementation is based on directly computing (12)-(15) rather than relying on the output of the `glm()` function.

Beside the parameter estimates, the function produces the standard errors of the estimates and the log–likelihood value. Further, the the Akaike's Information Criterion (AIC) and Bayesian Information Criteria (BIC) are calculated to find the best fitting line for the data, using the expressions

$$\text{AIC} = -2\ell_P(\lambda) + 2 \times (p + 2K - 1 + c) \tag{18}$$

$$BIC = -2\ell_P(\lambda) + \log(n) \times (p + 2K - 1 + c) \tag{19}$$

where $\ell_P(\lambda)$ is the profile log-likelihood function given in (16) which is obtained by substituting the maximum likelihood estimators of the model parameters (i.e. $z = \hat{z}$, $\pi = \hat{\pi}$, $\beta = \hat{\beta}$ and $\sigma = \hat{\sigma}$), and the second part of the AIC and BIC equations computes the number of parameters estimated in the model. p is the number of regression parameters in $\hat{\beta}$, K is the number of mixture classes, c is the value 1 if the transformation parameter is estimated and zero otherwise, and n is the number of observations.

To support diagnostics and model checking, a plot of the disparity with the iteration number on the x-axis and the mass points on the y-axis, as well as normal Q-Q plots to determine how well a set of values follow a normal distribution, can be obtained. Furthermore, control charts of the residuals of the data before and after applying the transformation can be produced to detect special causes of variation. There are many possible causes of an out–of–control point, including non-normal data and the number of classes, K.

2.4. Starting point selection and the first cycle

In the first cycle of the algorithm, the model is fitted initially without random effect, given some starting values β^0 and σ^0. It remains to choose the starting mass points z_k^0 and corresponding masses π_k^0, for which the implementation of **boxcoxmix** provides two different methods as outlined below:

- Gauss-Hermite quadrature points (Einbeck and Hinde 2006):

$$z_k^0 = \hat{\beta}_0 + tol \times s \times g_k \tag{20}$$

  where β_0 is the intercept of the fitted model, *tol* is a scaling parameter restricted to the choice $0 \leq tol \leq 2$, g_k are Gauss-Hermite quadrature points, and s is the standard deviation of residuals defined as,

$$s = \sqrt{\frac{1}{n-p}\sum_{i=1}^{n} \hat{\varepsilon}_i{}^2} \tag{21}$$

  where $n - p$ is the degrees of freedom for $\hat{\varepsilon}_i$, n is the sample size, p represents the number of parameters used to fit the model $y_i^{(\lambda)} = x_i^T\beta + \varepsilon_i$ and $\hat{\varepsilon}_i$ is the difference between the observed data of the dependent variable $y_i^{(\lambda)}$ and the fitted values $\hat{y_i}^{(\lambda)}$(i.e. $\hat{\varepsilon}_i = y_i^{(\lambda)} - \hat{y_i}^{(\lambda)}$).

- Quantile-based version

$$z_k^0 = \bar{y}^{(\lambda)} + tol \times q_k^{(\lambda)} \tag{22}$$

  where $\bar{y}^{(\lambda)}$ is the mean of the responses $y_i^{(\lambda)}$ and $q_k^{(\lambda)} = \frac{k}{K} - \frac{1}{2K}$ are quantiles of the empirical distribution of $y_i^{(\lambda)} - \bar{y}^{(\lambda)}$.

(For either case, **boxcoxmix** provides the functions `Kfind.boxcox` and `tolfind.boxcox()` to identify optimal values of K and *tol*, respectively.)

From this one obtains the extended linear predictor for the k-th component $E(y_i^{(\lambda)}|z_k^0) = x_i^T\beta + z_k^0$. Using formula (11) with current parameter estimates, one gets an "initial E-step" and in the subsequent M-step one obtains the parameter estimates by solving the score equations. From the resulting estimates of this cycle, one gets an updated value of the weights, and so on.

2.5. Generic functions

boxcoxmix supports generic functions such as `summary()`, `print()` and `plot()`. Specifically, `plot()` can be applied on the output of `np.boxcoxmix()`, `optim.boxcox()`, `Kfind.boxcox` and `tolfind.boxcox()`. The plots to be printed depend on the choice of the argument `plot.opt`,

- 1, the disparities with the iteration number against the mass points;
- 2, the fitted values against the response of the untransformed and the transformed data;
- 3, probability plot of residuals of the untransformed against the transformed data;
- 4, individual posterior probabilities;
- 5, control charts of residuals of the untransformed against the transformed data;
- 6, the histograms of residuals of the untransformed against the transformed data;
- 7, plots the specified range of *tol* against the disparities (works only for the `tolfind.boxcox()` function);
- 8, gives the profile likelihood function that summarises information concerning λ (works only for the `optim.boxcox()` function);
- 7, plots the specified range of K against the aic or bic values (works only for the `Kfind.boxcox` function).

2.6. Application to the strength data

In this section we analyze the **strength** data from the **R** library **mdscore** (da Silva-Júnior, da Silva, and Ferrari 2014) which is a subsample of the 5 x 2 factorial experiment given by Ostle and Malone (1954). The objective here is to investigate the effects of the covariates `lot` and `cut` on the impact strength, where `lot` denotes the lot of the material (I, II, III, IV, V) and `cut` denotes the type of specimen cut (Lengthwise, Crosswise). The model presented is a two-way `lot` $\times$ `cut` interaction model. For the i–th `cut` and j–th `lot`, we have

$$y_{ij} = \gamma_i + \beta_j + \delta_{ij} + z, \qquad i = 1, 2, \quad j = 1, 2, .., 5, \tag{23}$$

where $\gamma_1 = 0$, $\beta_1 = 0$, $\delta_{1,1} = \delta_{1,2} = \cdots = \delta_{1,5} = \delta_{2,1} = 0$, and z is the random effect with an unspecified mixing distribution.

Shuster and Miura (1972) considered the Inverse Gaussian distribution as an adequate distribution in modelling strength data. We therefore suggest to fit a number of models including the Inverse Gaussian model and compare the results below.

For a fixed value of λ, we fit the model with settings $\lambda = -1$, $tol = 1.8$ and $K = 3$ (the latter two choices to be justified below), so the response will be transformed as

$$y^{(\lambda)} = (y^{-1} - 1)/-1 = -y^{-1} + 1.$$

Using `np.boxcoxmix()`,

```
> library(boxcoxmix)
> data(strength, package="mdscore")
> test.inv <- np.boxcoxmix(y ~ cut *lot, data = strength, K = 3,
+                          tol = 1.8,  start = "gq", lambda = -1,
+                          verbose=FALSE)
> test.inv

Call:
np.boxcoxmix(formula = y ~ cut * lot, data = strength, K = 3,
    tol = 1.8, lambda = -1, verbose = FALSE, start = "gq")



Coefficients
:
        cut Crosswise                 lot II
             -0.41743               -0.13097
              lot III                 lot IV
             -0.45223               -0.03384
                 lot V   cut Crosswise:lot II
             -0.81609                0.49649
cut Crosswise:lot III   cut Crosswise:lot IV
              0.18130                0.34043
  cut Crosswise:lot V
              0.25951

MLE of sigma:           0.06169

Mixture proportions:
    MASS1      MASS2      MASS3
0.2309757  0.3024323  0.4665921
-2 log L:             -73.7 and AIC =  -45.7085
```

For comparison, we also fit the same model without transformation, using function `np.boxcoxmix()` with setting $\lambda = 1$, $tol = 1.8$ and $K = 3$:

```
> test.gauss <- np.boxcoxmix(y ~ cut *lot, data = strength, K = 3,
+                            tol = 1.8,  start = "gq", lambda = 1,
+                            verbose=FALSE)
> test.gauss

Call:
np.boxcoxmix(formula = y ~ cut * lot, data = strength, K = 3,
    tol = 1.8, lambda = 1, verbose = FALSE, start = "gq")



Coefficients
:
        cut Crosswise                 lot II
              -0.2555                -0.0801
              lot III                 lot IV
              -0.2722                -0.2203
                lot V   cut Crosswise:lot II
              -0.5401                 0.3322
cut Crosswise:lot III   cut Crosswise:lot IV
               0.1554                 0.4070
  cut Crosswise:lot V
               0.3535

MLE of sigma:           0.02059

Mixture proportions:
    MASS1      MASS2      MASS3
0.3666667  0.4665295  0.1668039
-2 log L:            -86.6 and AIC =  -58.6193
```

Using now our grid search method `optim.boxcox()` that calculates and plots the profile log-likelihood values for the fitted model (23) against a set of λ values, and locates the MLE $\hat{\lambda}$ (see Fig. 1):

```
> test.optim <- optim.boxcox(y ~ cut*lot, data = strength,  K = 3,
+                            tol = 1.8, start = "gq", find.in.range = c(-3, 3),
+                            s = 60)
> plot(test.optim, 8)
```

Figure 1 shows that the best value of λ that maximizes the profile log-likelihood is 0.1 which is close to zero, suggesting that some transformation need to be carried out to make the data distribution appear more normal.

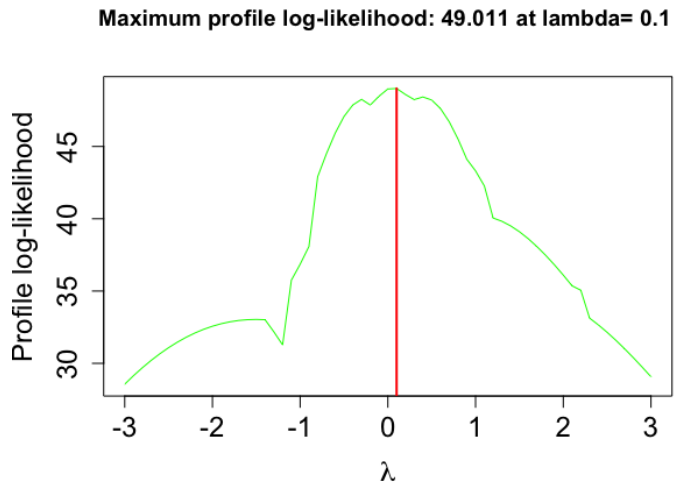

Figure 1: A grid search over λ, using $K = 3$ and $tol = 1.8$

We also fit the model shown in (23) with an Inverse Gaussian distribution using the **npmlreg** function `alldist()`, using $tol = 0.45$ and $K = 3$.

```
> library(npmlreg)
> inv.gauss <- alldist(y ~ cut*lot, data = strength, k = 3, tol = 0.45,
+                      verbose=FALSE,  family = "inverse.gaussian")
> inv.gauss

Call:  alldist(formula = y ~ cut * lot, family = "inverse.gaussian",      data = strength,

Coefficients:
        cut Crosswise                 lot II
              0.36114               -0.32801
              lot III                 lot IV
              0.44347                0.08572
                lot V   cut Crosswise:lot II
              2.25158               -0.51105
cut Crosswise:lot III   cut Crosswise:lot IV
              0.51464               -0.19985
  cut Crosswise:lot V                   MASS1
             -0.19233                0.73629
                MASS2                   MASS3
              1.25598                1.95567

Random effect distribution - standard deviation:           0.3965868

Mixture proportions:
```

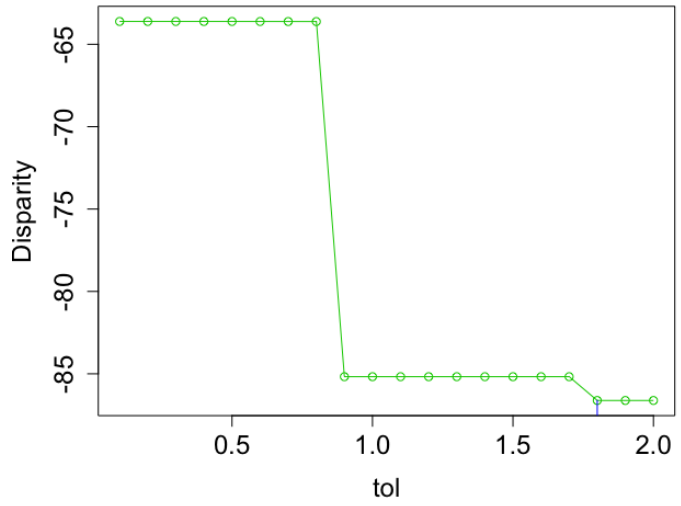

Figure 2: For the strength data, a grid search over *tol*, using $K = 3$ and $\lambda = 1$

```
    MASS1      MASS2      MASS3
0.1681332  0.5895689  0.2422979
-2 log L:           -68
```

For the starting point selection, the optimal value of *tol* can be selected prior to this analysis using a grid search over *tol* using **boxcoxmix** function `tolfind.boxcox()` (see Fig. 2).

```
> tol.find <- tolfind.boxcox(y ~ cut*lot,  data = strength,  K = 3,
      start = "gq", lambda = 1,  find.in.range = c(0, 2),  s = 20)
```

Similarly, the value $tol = 0.45$ used by `alldist()` has been selected as the optimal value of *tol* using the **npmlreg** function `tolfind()`.

The Akaike Information Criteria (AIC) defined in (18), is used as a criterion for choosing amongst the models. The model with the lowest AIC value is considered as the best model. Table 1 displays summary statistics for the Inverse Gaussian distribution model (Inv.Gauss), transformed models using $\lambda = -1$ and $\hat{\lambda} = 0.1$, and the untransformed model ($\lambda = 1$).

The Inverse Gaussian model gives the worst AIC. Better AIC values are given by the transformed model using $\lambda = -1$, the Gaussian ($\lambda = 1$) and $\hat{\lambda}$. The lowest AIC found was for the transformed model using $\hat{\lambda}$ with -68.0224. The parameter estimates of the untransformed and the Box–Cox–transformed model using $\hat{\lambda}$ are broadly in agreement but the latter has better disparity and AIC values. However, the results from the other models are quite different and the worst disparity was found for the Inverse Gaussian model. Among the four models, the one with $\hat{\lambda} = 0.1$ provides the best fit of the data, which does not necessarily support the model choice taken in Shuster and Miura (1972).

	Inv.Gauss	$\lambda = -1$	$\hat{\lambda} = 0.1$	$\lambda = 1$
γ_2	0.3611	-0.4174	-0.2943	-0.2555
β_2	-0.3280	-0.1310	-0.0887	-0.0801
β_3	0.4435	-0.4522	-0.3175	-0.2722
β_4	0.0857	-0.0338	-0.2383	-0.2203
β_5	2.2516	-0.8161	-0.6845	-0.5401
$\delta_{2,2}$	-0.5111	0.4965	0.3715	0.3323
$\delta_{2,3}$	0.5146	0.1813	0.1141	0.1554
$\delta_{2,4}$	-0.1999	0.3404	0.4604	0.4070
$\delta_{2,5}$	-0.1923	0.2595	0.3378	0.3536
σ	0.3966	0.06169	0.0207	0.0206
$-2 \log L$	-68	-73.70853	-98.02242	-86.61931
AIC	-40	-45.7085	-68.02242	-58.6193

Table 1: Comparison of results from original & transformed data, using $K = 3$.

K	$\lambda = -1$	$\hat{\lambda} = 0.1$	$\lambda = 1$
1	-30.01438	-33.57915	-29.45051
2	-50.10725	-56.71019	-44.64449
3	-45.70853	-70.02242	-58.61931
4	-50.42968	-59.40018	-52.4271
5	-57.4437	-60.17015	-49.17725
6	-64.53892	-51.40021	-44.42724
7	-49.44363	-52.17016	-54.39248

Table 2: Comparison of AIC values

The appropriate number of classes K could be obtained by comparing the AIC from fitting several mixture models with different numbers of classes K, as illustrated in Table 2.

3. Box-Cox transformation in variance component models

3.1. Variance component model

We now consider the two-level variance component model. An unobserved random effect z_i with upper-level indexed by $i = 1 \ldots, r$, and lower-level indexed by $j = 1, \ldots, n_i$, $\sum n_i = n$ is added to the linear predictor $x_{ij}^T\beta$. The responses y_{ij} are independently distributed with conditional mean function

$$E(y_{ij}|z_i) = x_{ij}^T\beta + z_i \tag{1}$$

where the distribution of the z_i is again unspecified. The conditional probability density function of y_{ij} given z_i is given by

$$f(y_{ij}|z_i) = \phi(y_{ij}; x_{ij}^T\beta + z_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_{ij} - x_{ij}^T\beta - z_i)^2\right]. \tag{2}$$

3.2. Extending the Box-Cox Transformation to variance component models

For the two-level variance component model with responses y_{ij}, the Box-Cox transformation (Box and Cox 1964) can be written as

$$y_{ij}^{(\lambda)} = \begin{cases} \frac{y_{ij}^{\lambda}-1}{\lambda} & \lambda \neq 0, \\ \log y_{ij} & \lambda = 0 \end{cases} \tag{3}$$

for $y_{ij} > 0$, $i = 1, ..., r$, $j = 1, ..., n_i$, and $\sum n_i = n$. It is assumed that there is a value of λ for which

$$y_{ij}^{(\lambda)}|z_i \sim N(x_{ij}^T\beta + z_i, \sigma^2) \tag{4}$$

where z_i is a random effect with an unspecified mixing distribution $g(z_i)$. The likelihood can now be approximated as (Aitkin *et al.* 2009)

$$L(\lambda, \beta, \sigma^2, g) = \prod_{i=1}^{r} \int \left[\prod_{j=1}^{n_i} f(y_{ij}|z_i)\right] g(z_i)dz_i \approx \prod_{i=1}^{r} \sum_{k=1}^{K} \pi_k m_{ik}, \tag{5}$$

where $m_{ik} = \prod_{j=1}^{n_i} f(y_{ij}|z_k)$. The complete log-likelihood is thus

$$\ell^* = \log L^* = \sum_{i=1}^{r} \sum_{k=1}^{K} [G_{ik} \log \pi_k + G_{ik} \log m_{ik}] \tag{6}$$

where $L^* = \prod_{i=1}^{r} \prod_{k=1}^{K} (\pi_k m_{ik})^{G_{ik}}$.

We apply the expectation-maximization (EM) approach similar as before, with the following adjustments:

E-step: This is identical to (11), but with f_{ik} replaced by m_{ik}.
M-step: Using the current w_{ik}, the four estimators are now:

$$\begin{aligned}
\hat{\beta} &= \left(\sum_{i=1}^{r}\sum_{j=1}^{n_i} x_{ij}x_{ij}^T\right)^{-1} \sum_{i=1}^{r}\sum_{j=1}^{n_i} x_{ij}\left(y_{ij}^{(\lambda)} - \sum_{k=1}^{K} w_{ik}\hat{z}_k\right), \\
\hat{\sigma}^2 &= \frac{\sum_{i=1}^{r}\sum_{k=1}^{K} w_{ik}\left[\sum_{j=1}^{n_i}(y_{ij}^{(\lambda)} - x_{ij}^T\hat{\beta} - \hat{z}_k)^2\right]}{\sum_{i=1}^{r} n_i}, \\
\hat{z}_k &= \frac{\sum_{i=1}^{r} w_{ik}\left[\sum_{j=1}^{n_i}(y_{ij}^{(\lambda)} - x_{ij}^T\hat{\beta})\right]}{\sum_{i=1}^{r} n_i w_{ik}}, \\
\hat{\pi}_k &= \frac{\sum_{i=1}^{r} w_{ik}}{r},
\end{aligned}$$

where $\hat{\pi}_k$ is the average posterior probability for component k. Substituting the results into Equation (5) we get the non-parametric profile likelihood function $L_P(\lambda)$, or its logarithmic version $\ell_P(\lambda) = \log(L_P(\lambda))$. The non–parametric profile maximum likelihood (NPPML) estimator is therefore given by

$$\hat{\lambda} = arg\ \max_{\lambda}\ \ell_P(\lambda). \tag{7}$$

For fixed λ, such variance component models under response transformations are again estimated using the function `np.boxcoxmix()`. When λ =1 (no transformation), the results of the proposed approach will be similar to that of the **npmlreg** function `allvc()`.

3.3. Application to the heights of boys in Oxford data

In order to demonstrate how the `optim.boxcox()` function may be used effectively, we consider a data set giving the heights of boys in Oxford. The data set is part of the **R** package nlme (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2016) and consists of measurements of `age` and `height` for 26 boys, yielding a total of 234 observations. The response variable `height` is defined as the height of the boy in (cm), associated with the covariate `age` that is the standardized age (dimensionless). The results were obtained by fitting the variance component model

$$E(y_{ij}|z_i) = \texttt{age}_j + z_i \tag{8}$$

where z_i is boy–specific random effect and $\texttt{age}_j$ is the j-th standardized age measurement, $j = 1, \ldots, 9$, which is equal for all boys for fixed j. A model with $K = 6$ mass points without response transformation can be fitted using the `np.boxcoxmix()` function setting $\lambda = 1$,

```
> data(Oxboys, package="nlme")
> Oxboys$boy <- gl(26,9)
> Oxboys$boy
```

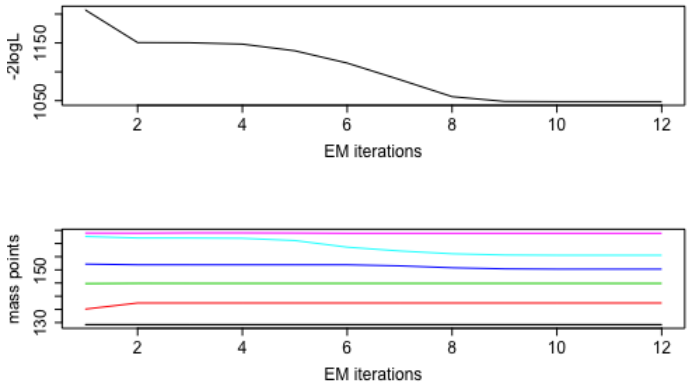

Figure 3: For the Oxboys data, estimated mass points versus EM iterations.

```
 [1] 1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2
[19] 3  3  3  3  3  3  3  3  3  4  4  4  4  4  4  4  4  4
[37] 5  5  5  5  5  5  5  5  5  6  6  6  6  6  6  6  6  6
[55] 7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8
[73] 9  9  9  9  9  9  9  9  9  10 10 10 10 10 10 10 10 10
[91] 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12
[109] 13 13 13 13 13 13 13 13 13 14 14 14 14 14 14 14 14 14
[127] 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 16
[145] 17 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18
[163] 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20 20 20
[181] 21 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22
[199] 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24
[217] 25 25 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26
26 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ... 26
```

```
> testox <- np.boxcoxmix(height ~  age, groups = Oxboys$boy,
+                        data = Oxboys, K = 6, tol = 1, start = "gq",
+                        lambda=1, verbose=FALSE)

> plot(testox, 1)
```

The manual specification of `Oxboys$boy <- gl(26,9)` is necessary since the second argument of `np.boxcoxmix` requires a vector of group labels in order to work correctly.

The function `optim.boxcox()` can again be used to perform a grid search over λ to obtain the optimum:

```
> testo <- optim.boxcox(height ~ age, groups = Oxboys$boy, data = Oxboys,
+          K = 6, tol =1, start = "gq", find.in.range = c( -1.2, 0.1), s=15)
> plot(testo, 8)
```

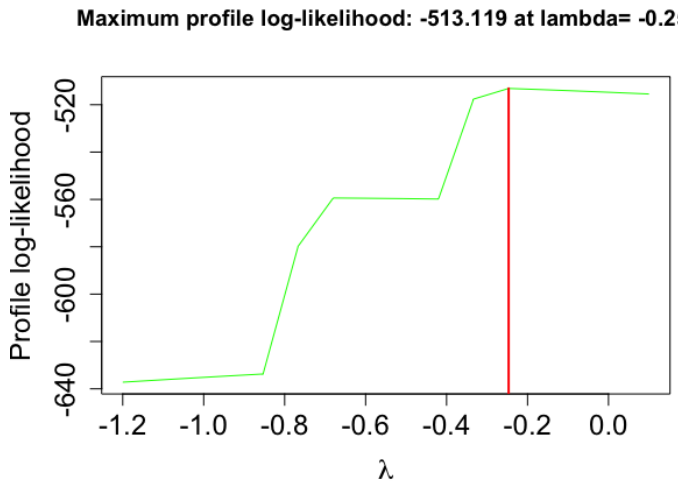

Figure 4: For the Oxboys data, a grid search over λ, with $K = 6$ and $tol = 1$.

	$K=4$		$K=5$		$K=6$		$K=7$	
	$\hat{\lambda}=0.1$	$\lambda=1$	$\hat{\lambda}=-0.51$	$\lambda=1$	$\hat{\lambda}=-0.25$	$\lambda=1$	$\hat{\lambda}=-0.25$	$\lambda=1$
$\hat{\beta}$	0.0716	6.5264	0.0034	6.5218	0.0126	6.5245	0.0082	6.5218
$SE(\hat{\beta})$	0.0031	0.2841	0.0001	0.2367	0.0004	0.1918	0.0002	0.2367
$\hat{\sigma}$	0.0310	2.806	0.0012	2.341	0.0035	1.903	0.0023	2.341
$-2\log L$	1211.8	1212.7	1119.3	1132.8	1026.2	1048.3	1022.3	1132.8
AIC	1229.8	1228.659	1141.324	1152.849	1052.2	1072.27	1052.302	1160.849

Table 3: Comparison of results from original & transformed data, using $K = 4, 5, 6$ and 7

From Figure 4, it can be seen that the best estimate of λ that maximizes the non-parametric profile log-likelihood is -0.25, suggesting that some transformation need to be carried out to make the data distribution more normal. The results before and after applying the response transformation shown in Table 3 prove that the decision of transforming the response is reasonable.

As can be seen from Table 3, comparing the Akaike Information Criterion (AIC) values of the untransformed model fit ($\lambda = 1$) and our method using $K = 4, 5, 6$ and 7, respectively, showed a slightly better performance of the NPPML approach. In other words, using the response data after applying the response transformation leads to a better fitting model than the original data. This gives further support to the decision of using the transformation.

Concerning the choice of K, it is transparent from Table 3 that there is no gain in going from $K = 6$ to $K = 7$ as the AIC values in fact increase when doing so. There is a consistent improvement, however, when increasing the number of mass points from $K = 4$ over $K = 5$ to $K = 6$, and it is also clear from Figure 3 that the six estimated mass points are distinct and identifiable. For the untransformed model, Aitkin *et al.* (2009) recommend the use of $K = 8$ mass points.

4. Discussion

We have introduced a new **R** package, **boxcoxmix**, that identifies the appropriate power transformation for achieving normality of the response distribution in random effect models. To the best of our knowledge, there is no other widely available statistical package that has implemented the Box-Cox power transformation of the linear mixed effects model with an unspecified random effect distribution. **boxcoxmix** is able to fit random effect and variance component models, and estimates the transformation and regression parameters simultaneously through its main function `optim.boxcox()`. This function operates similarly to the existing R function `boxcox()`, by creating a profile likelihood and carrying out a grid search over the transformation parameter λ. It is noted that, just as in `boxcox()`, this procedure cannot make use of built–in **R** optimization routines such as `optim()` or `optimize()` since the profile likelihood itself depends on estimated parameters, estimation of which involves a full EM algorithm.

In addition, **boxcoxmix** also can be used to fit models with fixed value of λ using function `np.boxcoxmix()`, and to perform a grid search over *tol* using the function `tolfind.boxcox()` to identify optimal starting values for the mass points. Our package provides some further diagnostic tools, such as a QQ–plot and a control chart of residuals, which help validating the need for transformation.

In this paper we have shown how **boxcoxmix** can successfully fit models through response transformation rather than adjustment of the response distribution. The examples have demonstrated that the **boxcoxmix** function `optim.boxcox()` works well in finding the model with maximum likelihood. All transformed models using $\hat{\lambda}$ that were obtained by the `optim.boxcox()` function gave substantially better fits than the untransformed model, when considering the AIC criterion or the disparity $(-2 \log L)$. Also, in all considered scenarios, the estimated value of $\hat{\lambda}$ was quite far away from the value $\lambda = 1$. However, it should be added that it is not possible to report a simple likelihood–based standard error for $\hat{\lambda}$ as in **R** function `boxcox()`, the reason being that the likelihood in the considered model class is highly non–concave, as visible for instance from Fig. 1. Hence, when faced with the decision of whether or not needing to transform the response, not only the value of $\hat{\lambda}$ but also relevant model selection criteria such as AIC should be taken into account. It is then essential that these are always based on likelihoods which are reported on the original response scale, as in models (6) and (7) — of course, this is the case for the values $-2 \log L$ and AIC provided in our summary output. The experimental results verify the accuracy and the efficiency of the **boxcoxmix** package, which is available from the Comprehensive **R** Archive Network (CRAN) at `https://cran.r-project.org/package=boxcoxmix`.

Acknowledgements

This **R** package has made use of R package statmod (Giner and Smyth 2016) to obtain the function `gqz`.

References

Aitkin M (1996). "A General Maximum Likelihood Analysis of Overdispersion in Generalized Linear Models." *Statistics and Computing*, **6**(3), 251–262.

Aitkin M (1999). "A General Maximum Likelihood Analysis of Variance Components in Generalized Linear Models." *Biometrics*, **55**(1), 117–128.

Aitkin MA, Francis B, Hinde J, Darnell R (2009). *Statistical Modelling in R*. Oxford University Press Oxford.

Bock RD, Aitkin M (1981). "Marginal Maximum Likelihood Estimation of Item Parameters: Application of an EM Algorithm." *Psychometrika*, **46**(4), 443–459.

Box GE, Cox DR (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211–252.

da Silva-Júnior AHM, da Silva DN, Ferrari SLP (2014). "mdscore: An R Package to Compute Improved Score Tests in Generalized Linear Models." *Journal of Statistical Software*, **61**(2), 1–16. URL http://www.jstatsoft.org/v61/c02/.

Davies R (1987). "Mass Point Methods for Dealing with Nuisance Parameters in Longitudinal Studies." *Longitudinal Data Analysis*, pp. 88–109.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society, Series B*, pp. 1–38.

Einbeck J, Darnell R, Hinde J (2014). *npmlreg: Nonparametric Maximum Likelihood Estimation for Random Effect Models*. R package version 0.46-1, URL https://CRAN.R-project.org/package=npmlreg.

Einbeck J, Hinde J (2006). "A Note on NPML Estimation for Exponential Family Regression Models with Unspecified Dispersion Parameter." *Austrian Journal of Statistics.*, **35**(2&3), 233–243.

Einbeck J, Hinde J, Darnell R (2007). "A New Package for Fitting Random Effect Models." *R News.*, **7**(1), 26–30. URL http://dro.dur.ac.uk/18462/.

Giner G, Smyth GK (2016). "statmod: Probability Calculations for the Inverse Gaussian Distribution." *arXiv preprint arXiv:1603.06687*.

Gurka MJ, Edwards LJ, Muller KE, Kupper LL (2006). "Extending the Box-Cox Transformation to the Linear Mixed Model." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **169**(2), 273–288.

Heckman J, Singer B (1984). "A Method for Minimizing the Impact of Distributional Assumptions in Econometric Models for Duration Data." *Econometrica: Journal of the Econometric Society*, pp. 271–320.

Hou Q, Mahnken JD, Gajewski BJ, Dunton N (2011). "The Box-Cox Power Transformation on Nursing Sensitive Indicators: Does it Matter if Structural Effects are Omitted During the Estimation of the Transformation Parameter?" *BMC Medical Research Methodology*, **11**(1), 1.

Karlis D, Xekalaki E (2003). "Choosing Initial Values for the EM Algorithm for Finite Mixtures." *Computational Statistics & Data Analysis*, **41**(3), 577–590.

Laird N (1978). "Nonparametric Maximum Likelihood Estimation of a Mixing Distribution." *Journal of the American Statistical Association*, **73**(364), 805–811.

Lindsay BG, *et al.* (1983). "The Geometry of Mixture Likelihoods: a General Theory." *The Annals of Statistics*, **11**(1), 86–94.

Nawata K (1994). "Estimation of Sample Selection Bias Models by the Maximum Likelihood Estimator and Heckman's Two-Step Estimator." *Economics Letters*, **45**(1), 33–40.

Nawata K, *et al.* (2013). "A new estimator of the Box-Cox Transformation Model Using Moment Conditions." *Economics Bulletin*, **33**(3), 2287–2297.

Ostle B, Malone LC (1954). "Statistics in Research: Basic Concepts and Techniques for Research Workers." *Technical report*, JSTOR.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2016). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-128, URL http://CRAN.R-project.org/package=nlme.

Polańska J (2003). "The EM Algorithm and its Implementation for the Estimation of Frequencies of SNP-Haplotypes." *International Journal of Applied Mathematics and Computer Science*, pp. 419–429.

Sakia R (1992). "The Box-Cox transformation technique: a review." *The Statistician*, pp. 169–178.

Shuster J, Miura C (1972). "Two-Way Analysis of Reciprocals." *Biometrika*, **59**(2), 478–481.

Sofroniou N, Einbeck J, Hinde J (2006). "Analyzing Irish suicide rates with mixture models." National University of Ireland.

Solomon P (1985). "Transformations for Components of Variance and Covariance." *Biometrika*, **72**(2), 233–239.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Affiliation:

Amani Almohaimeed
Department of Mathematical Sciences
Qassim University
Qassim, KSA
and
Department of Mathematical Sciences
Durham University
Durham, UK
E-mail: ama.almohaimeed@qu.edu.sa

Jochen Einbeck
Department of Mathematical Sciences
Durham University
Durham, UK
E-mail: jochen.einbeck@durham.ac.uk