`tramME`: Mixed-Effects Transformation Models Using Template Model Builder

Bálint Tamási Torsten Hothorn

February 4, 2021

Abstract

Linear transformation models constitute a general family of parametric regression models for discrete and continuous responses. To accommodate correlated responses, the model is extended by incorporating mixed effects. This article presents the R package `tramME`, which builds on existing implementations of transformation models (`mlt` and `tram` packages) as well as Laplace approximation and automatic differentiation (using the `TMB` package), to calculate estimates and perform likelihood inference in mixed-effects transformation models. The resulting framework can be readily applied to a wide range of regression problems with grouped data structures.

1 Introduction

Datasets with grouped observations are abundant in the applied statistical practice. Clustering, hierarchical designs, longitudinal studies or repeated measurements can all lead to grouped data structures. The common property of these datasets is that observations within groups, defined by one or more grouping factors, cannot be treated as independent. In order to draw valid inference, the statistical model has to address the issue of correlated observations. Mixed-effects models represent one of the main approaches dealing with this type of regression problems. In this approach, the observations are assumed to be independent *conditionally* on a set of random effects that aim to capture unmodeled group-level heterogeneity. The reader is referred, for example, to the textbook by Demidenko (2013) for an exposition and examples of the usage of mixed-effects models. Several R packages exist that implement mixed-effects models for specific types of regression problems. The two most notable examples are `nlme` by Pinheiro et al. (2019) and `lme4` by Bates et al. (2015) for linear, non-linear and generalized linear mixed-effects models, respectively.

Linear transformation models aim to directly specify the conditional distribution function of an outcome variable in a regression setting. Hothorn (2020b) proposed a fully parametric approach using a flexible monotone increasing transformation function that is estimated from the data. The resulting general model family can be applied to a wide range of problems with at least ordered discrete outcome variables. In fact, many of the popular regression models can be expressed as special cases of the linear transformation model framework. Most recently, Tian et al. (2020) reviewed the approach followed in this study and compared it to an alternative semiparametric formulation using extensive simulations. With introducing random effects in the linear transformation model, it becomes applicable in a very diverse set of regression problems where the observations are correlated due to repeated measurements or grouped designs.

The structure of this article is as follows: After a brief, and somewhat technical, introduction of the methodology and the implementation in Section 2, Section 3 demonstrates, through a series of examples, that the package `tramME` can be flexibly applied to regression models with various response types and data structures. Finally, Section 4 discusses a few issues concerning the implementation of our model.

2 Mixed-effects transformation models

The model class in the R package `tramME` is an extension of the transformation model approach described by Hothorn et al. (2018) and implemented in the R packages `mlt` and `tram` by Hothorn (2020b) and Hothorn (2020a), respectively.

Formally, we are interested in models that parameterize the conditional distribution function directly,

$$\mathbb{P}\left(Y \leq y \mid \mathbf{x}, \mathbf{u}, \boldsymbol{\gamma}\right) = F_Z\left(h(y; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}\right) \qquad \boldsymbol{\gamma} \sim \mathcal{N}_q(\mathbf{0}, \boldsymbol{\Sigma}), \tag{1}$$

where F_Z denotes a pre-specified error distribution function (or inverse-link function) which is monotone increasing and maps from the real numbers to the closed interval $[0, 1]$. Typically, F_Z is set to the CDF of a simple continuous distribution, hence the name "error distribution". $h(y; \boldsymbol{\vartheta})$ is the baseline transformation function, which is also a monotonic increasing function parameterized with the vector $\boldsymbol{\vartheta}$. $\mathbf{x}^\top$ and $\mathbf{u}^\top$ are the suitable row vectors of fixed effects and random effects design matrices, respectively. The vector $\boldsymbol{\beta}$ contains the fixed effects, while $\boldsymbol{\gamma}$ comprises of the stacked (possibly multiple) random effects. The distribution of the random effects is assumed to be multivariate Gaussian with zero mean and covariance matrix $\boldsymbol{\Sigma}$, which typically has a sparse block structure.

As Table 1 shows, specific choices for the error distribution and the baseline transformation function lead to different types of regression models. In the R package `tramME`, seven main model types are distinguished, mainly based on the class of their outcome variable. Moreover, the functions `SurvregME()` and `PolrME()` allow to specify multiple error distributions or baseline transformations and hence increasing the number of available model types.

Table 1: Model types implemented in the `tramME` package. F_Z denotes the error distribution and the column $h(y; \boldsymbol{\vartheta})$ lists the basis functions the baseline transformation function utilizes.

Function	Name	F_Z	$h(y; \boldsymbol{\vartheta})$
`LmME()`	Mixed-effects normal linear regression	Standard Gaussian	Linear basis
`BoxCoxME()`	Non-normal (Box-Cox-type) linear mixed-effects regression	Standard Gaussian	Bernstein basis
`ColrME()`	Mixed-effects continuous outcome logistic regression	Standard logistic	Bernstein basis
`CoxphME()`	Mixed-effects parametric Cox regression	Minimum extreme value	Bernstein basis
`SurvregME()`	Mixed-effects parametric survival models	Multiple options	Multiple options
`PolrME()`	Mixed-effects regression models for ordinal outcomes	Multiple options	Discrete basis
`LehmannME()`	Mixed-effects Lehmann-alternative linear regression	Maximum extreme value	Bernstein basis

As the table indicates, some of the models specify their transformation functions as general smooth functions, approximated with the use of Bernstein polynomials. The function $h(y; \boldsymbol{\vartheta})$ has to be monotonic increasing, so that the conditional distribution function is also increasing. When using a set of order p Bernstein polynomials for the approximation of a general function, this restriction conveniently translates to the parameter restriction $\vartheta_i \leq \vartheta_{i+1}$ for all $i = 0, \ldots, p-1$.

The observations are assumed to be *conditionally independent*, and hence the likelihood has the form

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\vartheta}, \boldsymbol{\beta}, \boldsymbol{\Sigma}\right) &= \int_{\mathbb{R}^q} \mathcal{L}(\boldsymbol{\vartheta}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \, \mathrm{d}\boldsymbol{\gamma} \\ &= \int_{\mathbb{R}^q} \prod_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\vartheta}, \boldsymbol{\beta} \mid \boldsymbol{\gamma}) \phi(\boldsymbol{\gamma}; \boldsymbol{\Sigma}) \, \mathrm{d}\boldsymbol{\gamma}, \end{aligned} \tag{2}$$

where $\mathcal{L}(\boldsymbol{\vartheta}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ is the joint likelihood function, given the all observations, and $\mathcal{L}_i(\boldsymbol{\vartheta}, \boldsymbol{\beta} \mid \boldsymbol{\gamma})$ denotes the individual *conditional* likelihood contributions. $\phi(\boldsymbol{\gamma}; \boldsymbol{\Sigma})$ stands for the probability density function of the multivariate normal distribution with vector zero mean and covariance matrix $\boldsymbol{\Sigma}$. This latter function can be factorized further according to the covariance structure of the random effects.

One of the main advantages of working directly with the distribution function of the outcome is that it is simple to introduce (random) censoring and truncation in the estimation procedure. The conditional likelihood contributions under different types of censoring can be written as

$$\mathcal{L}_i(\boldsymbol{\vartheta}, \boldsymbol{\beta} \mid \boldsymbol{\gamma}) = \begin{cases} f_Z(h(y; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}) h'(y; \boldsymbol{\vartheta}) & y \in \Xi & \text{“exact continuous”} \\ 1 - F_Z(h(\underline{y}; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}) & y \in (\underline{y}, \infty) \cap \Xi & \text{“right censored”} \\ F_Z(h(\bar{y}; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}) & y \in (-\infty, \bar{y}] \cap \Xi & \text{“left censored”} \\ F_Z(h(\bar{y}; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}) \\ \quad - F_Z(h(\underline{y}; \boldsymbol{\vartheta}) - \mathbf{x}^\top \boldsymbol{\beta} - \mathbf{u}^\top \boldsymbol{\gamma}) & y \in (\underline{y}, \bar{y}] \cap \Xi & \text{“interval censored”}, \end{cases}$$

where $f_Z()$ is the density function of the error distribution, $h'(y; \boldsymbol{\vartheta})$ is the first derivative of the baseline transformation function with respect to y, and Ξ denotes the sample space of Y.

The multidimensional integral in Equation (2), in general, does not have an analytical solution, but its value can be approximated using numerical methods. The package `tramME` applies the Laplace approximation to this problem, which relies on the quadratic Taylor expansion of the corresponding joint log-likelihood function.

The maximization of the logarithm of the likelihood function with respect to $\boldsymbol{\vartheta}$, $\boldsymbol{\beta}$, and $\boldsymbol{\Sigma}$, under a set of suitable constraints on $\boldsymbol{\vartheta}$ to make $h(y; \boldsymbol{\vartheta})$ monotone increasing, results in the maximum likelihood estimates of the model parameters. Standard likelihood theory, utilizing the ability to evaluate the log-likelihood function, the score function, and the Hessian, provides a basis for asymptotic inference in this family of models; see Hothorn et al. (2018) for more details on likelihood inference in transformation models.

The maximum likelihood estimation in `tramME` is done using the `TMB` package by Kristensen et al. (2016). The Template Model Builder (`TMB`) allows the user to define and estimate general, non-linear mixed effects models. It was built on well-tested and high-performance C++ libraries, which results in a flexible yet efficient framework for estimating mixed models with possibly complex random effects structures; see for example Brooks et al. (2017) for performance comparisons in the context of the package `glmmTMB`. In `tramME`, `TMB` is used to evaluate the integral in Equation (2), using Laplace's method, and to calculate the derivatives of the log-likelihood function using automatic (or algorithmic) differentiation.

3 Applications

In this section several applications of the transformation mixed models are presented, and wherever it is possible, also compared to other existing implementations. The examples shown here are by no means intended as complete analyses. They demonstrate how mixed-effects transformation models can be used in a broad range of regression problems, and showcase the most important features implemented in the package `tramME`.

3.1 Normal linear mixed model

As a first example, we model the average reaction times to a specific task from a sleep deprivation study described in Belenky et al. (2003). Figure 1 presents the reaction times against days of sleep deprivation for each of the 18 participants.

In this first example, we model the distribution of the average reaction time using random intercepts and random slopes for the effects of days of sleep deprivation.

$$\mathbb{P}\left(\texttt{Reaction} \leq y \mid \texttt{Days}, \alpha_i, \beta_i\right) = \Phi\left(\vartheta_1 + \vartheta_2 y - \beta \texttt{Days} - \gamma_{1_i} - \gamma_{2_i} \texttt{Days}\right), \tag{3}$$

$$\begin{pmatrix} \gamma_{1_i} \\ \gamma_{2_i} \end{pmatrix} \sim \mathcal{N}_2 \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & \tau_{12} \\ \tau_{12} & \tau_2^2 \end{pmatrix} \right\}$$

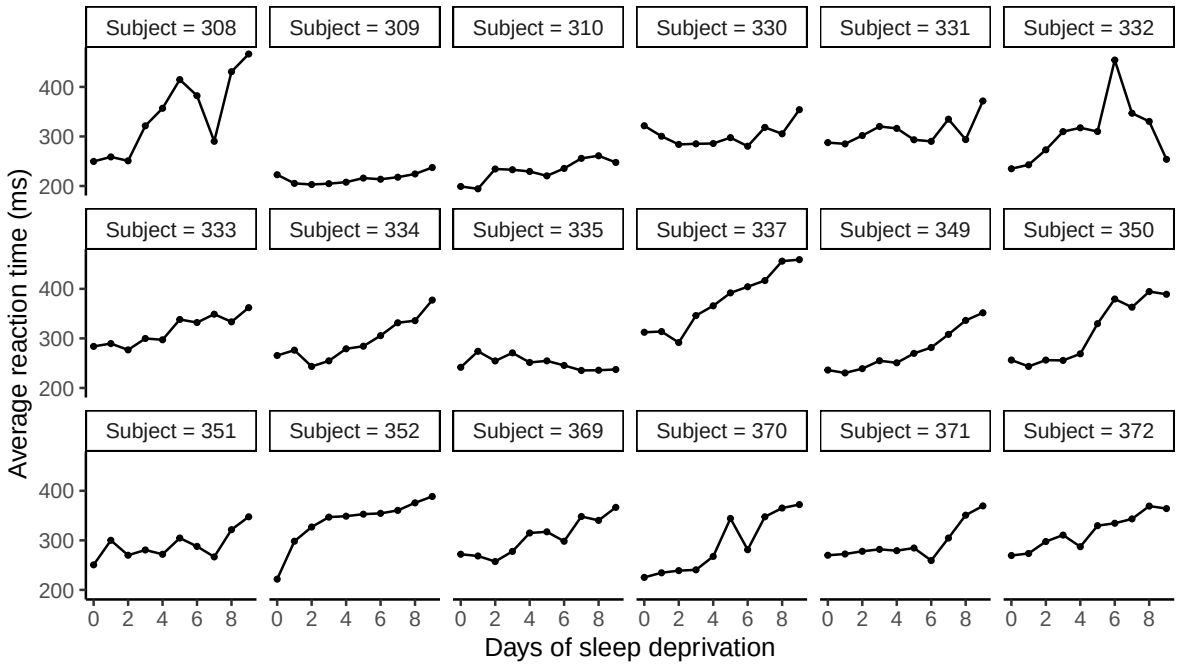

Figure 1: Sleep deprivation study: Average reaction times to a specific task of 18 participants after several days of sleep deprivation reported by Belenky et al. (2003).

Note that when the transformation function is assumed to be linear in the outcome variable, *i.e.*, $h(y) = \vartheta_1 + \vartheta_2 y$, we arrive at a re-parameterized version of the normal linear mixed effects model, and hence the results from `tramME::LmME()` are directly comparable to estimates using other mixed-effects regression packages such as `lme4`. Estimating the normal linear model with the `tramME`:

```
R> library("tramME")
R> sleep_lmME <- LmME(Reaction ~ Days + (Days | Subject), data = sleepstudy)
R> logLik(sleep_lmME)

'log Lik.' -876 (df=6)
```

To make the results from `lme4` comparable to the previous results, we set `REML = FALSE`, as the transformation mixed model implementation only supports the maximum likelihood estimation of the normal linear model specification.

```
R> library("lme4")
R> sleep_lmer <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy,
+                     REML = FALSE)
R> logLik(sleep_lmer)

'log Lik.' -876 (df=6)
```

The `as.lm = TRUE` option of various methods in `tramME` facilitates the comparisons between the transformation model parametrization and the results of a linear mixed model parametrization. Coefficient estimates and their standard errors from the transformation model approach are

```
R> cbind(coef = coef(sleep_lmME, as.lm = TRUE),
+        se = sqrt(diag(vcov(sleep_lmME, as.lm = TRUE, pargroup = "fixef"))))

              coef   se
(Intercept) 251.4 6.63
```

```
Days         10.5 1.50
```

while the results from `lmer` are

```
R> summary(sleep_lmer)$coefficients

            Estimate Std. Error t value
(Intercept)    251.4       6.63   37.91
Days            10.5       1.50    6.97
```

Similarly, the standard deviations and correlations of the random effects, as well as the standard deviations of the error terms are essentially the same

```
R> VarCorr(sleep_lmME, as.lm = TRUE) ## random effects


Grouping factor: Subject (18 levels)
Standard deviation:
(Intercept)        Days
      23.80        5.72

Correlations:
     (Intercept)
Days      0.0813

R> sigma(sleep_lmME) ## residual SD

[1] 25.6

R> VarCorr(sleep_lmer)

 Groups   Name        Std.Dev. Corr
 Subject  (Intercept) 23.78
          Days         5.72    0.08
 Residual             25.59
```

As the results show, the transformation model approach implemented by `LmME()` leads to the same results as the maximum likelihood estimation of the traditional linear mixed model parametrization. The advantage using the package `tramME`, as opposed to other well established implementations, is that it allows for incorporating randomly censored outcomes. Let us assume that the device in the sleep deprivation study is only able to measure reaction times larger than 200 ms and only in 50 ms step sizes. If we want to take this reduced accuracy in the measurements into account, we have to deal with interval-censored observations, as ignoring the censored nature of the outcomes could lead to biased parameter estimates.

With the following code, we create the interval-censored outcome vector using the `Surv` function of the `survival` package by Therneau (2015).

```
R> library("survival")
R> ub <- ceiling(sleepstudy$Reaction / 50) * 50
R> lb <- floor(sleepstudy$Reaction / 50) * 50
R> lb[ub == 200] <- 0
R> sleepstudy$Reaction_ic <- Surv(lb, ub, type = "interval2")
R> head(sleepstudy$Reaction_ic)

[1] [200, 250] [250, 300] [250, 300] [300, 350] [350, 400] [400, 450]
```

Using the interval-censored outcomes in the `LmME()` function call will maximize the correct likelihood function.

```
R> sleep_lmME2 <- LmME(Reaction_ic ~ Days + (Days | Subject), data = sleepstudy)
R> logLik(sleep_lmME2)

'log Lik.' -201 (df=6)
```

The value of the log-likelihood is different, because we are now calculating log-probabilities instead of log-densities of a continuous distribution, but, despite the decreased precision of the measurements, the parameter estimates are similar to what we got from the exactly observed outcomes.

```
R> cbind(coef = coef(sleep_lmME2, as.lm = TRUE),
+        se = sqrt(diag(vcov(sleep_lmME2, as.lm = TRUE, pargroup = "fixef"))))

             coef   se
(Intercept) 251.4 6.83
Days         10.5 1.62

R> sigma(sleep_lmME2)

[1] 28

R> VarCorr(sleep_lmME2, as.lm = TRUE)


Grouping factor: Subject (18 levels)
Standard deviation:
(Intercept)        Days
      22.30        5.94

Correlations:
     (Intercept)
Days      0.0536
```

The small estimated value of the correlation coefficient between the random slope and intercept suggests that a model with independent random effects might be more appropriate. To estimate such a model, we can use the same notation as in `lme4`

```
R> sleep_lmME3 <- LmME(Reaction_ic ~ Days + (Days || Subject), data = sleepstudy)
R> logLik(sleep_lmME3)

'log Lik.' -201 (df=5)
```

Comparing the two models using a likelihood ratio test, we see no evidence against the more parsimonious model (`sleep_lmME3`).

```
R> anova(sleep_lmME2, sleep_lmME3)

Model comparison

 Model 1: Reaction_ic ~ Days + (Days | Subject)
 Model 2: Reaction_ic ~ Days + (Days || Subject)

        Df logLik AIC BIC Chisq Chisq df Pr(>Chisq)
Model 2  5   -200 411 427
Model 1  6   -200 413 432  0.02        1       0.89
```

3.2 Box-Cox-type mixed-effects models

Substituting the linear baseline transformation function, $h(y) = \vartheta_1 + \vartheta_2 y$, with a general, monotonic increasing smooth function, we can relax the conditional normality assumption of the model discussed in Section 3.1.

The transformation model approach proposed by Hothorn et al. (2018) uses Bernstein bases to approximate this general increasing function in a fully parametric manner, *i.e.*, $h(y) = \mathbf{a}_{\text{Bs},K+1}(y)^\top \boldsymbol{\vartheta}$. The resulting model can be regarded as a version of the Box-Cox regression (Box and Cox, 1964), where the transformation of the response is estimated simultaneously with the model parameters. It should be pointed out that, although its approach is similar in spirit, `tramME` does not use the Box-Cox power transformation to approximate $h(y)$. For an implementation utilizing the original Box-Cox transform in the context of mixed-effects models, see the R package `boxcoxmix` by Almohaimeed and Einbeck (2018).

A more flexible version of the model described in (3) will take the form

$$\mathbb{P}\left(\texttt{Reaction} \leq y \mid \texttt{Days}, \alpha_i, \beta_i\right) = \Phi\left(\mathbf{a}(y)^\top \boldsymbol{\vartheta} - \beta\texttt{Days} - \gamma_{1_i} - \gamma_{2_i}\texttt{Days}\right), \tag{4}$$
$$\begin{pmatrix}\gamma_{1_i}\\ \gamma_{2_i}\end{pmatrix} \sim \mathcal{N}_2\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\tau_1^2 & \tau_{12}\\ \tau_{12} & \tau_2^2\end{pmatrix}\right\}$$

The Box-Cox-type transformation mixed model can be estimated using the `BoxCoxME()` function of the `tramME` package.

```
R> sleep_bc <- BoxCoxME(Reaction ~ Days + (Days | Subject), data = sleepstudy)
R> logLik(sleep_bc)

'log Lik.' -860 (df=11)
```

Note that the log-likelihood of this model is higher than that of the normal linear model, because we are now approximating the baseline transformation function flexibly, at the expense of a larger number of parameters.

Plotting the baseline transformation, $h(y) = \mathbf{a}(y)^\top \boldsymbol{\vartheta}$, can be used as a tool for checking the conditional normality assumption in the normal linear mixed model. In our example, Figure 2 reveals a slight departure from conditional normality in the outcome variable.

In many cases, the goal of the analysis is to estimate the *marginal* distribution of the outcomes, *i.e.*, integrating out the random effects from the conditional model (1). In the general formulation, there is no analytical solution for the integral, but we can use numerical methods to approximate the marginal distributions at various values of the outcome. The following code utilizes the `simulate` and `predict` methods implemented in the `tramME` package to get Monte Carlo estimates of the outcome distribution implied by the model (4).

```
R> ndraws <- 1000
R> nd <- expand.grid(
+    Reaction = seq(min(sleepstudy$Reaction), max(sleepstudy$Reaction), length.out = 100),
+    Days = 0:9,
+    Subject = 1)
R> re <- simulate(sleep_bc, newdata = nd, nsim = ndraws, what = "ranef", seed = 100)
R> cp <- parallel::mclapply(re, function(x) {
+    predict(sleep_bc, newdata = nd, ranef = x, type = "distribution")
+  }, mc.cores = 8)
R> cp <- array(unlist(cp), dim = c(100, 10, ndraws))
R> mp_bc <- apply(cp, c(1, 2), mean)
```

Figure 3 compares the conditional distributions obtained by integrating over the vector of random effects in models (3) and (4).

3.3 Mixed-effects continuous outcome logistic regression

The increased flexibility of the Box-Cox-type model, *i.e.*, using a general baseline transformation function instead of a linear one, comes with the price that the coefficient estimates will not be easily interpretable as expected changes in the mean in the conditional model. Switching to the standard logistic error distribution provides a solution to this problem, as the parameter estimates in the resulting model can be interpreted as log-odds ratios. This *continuous outcome logistic regression* model was used by Lohse et al. (2017) to analyze body mass index (BMI) distributions.

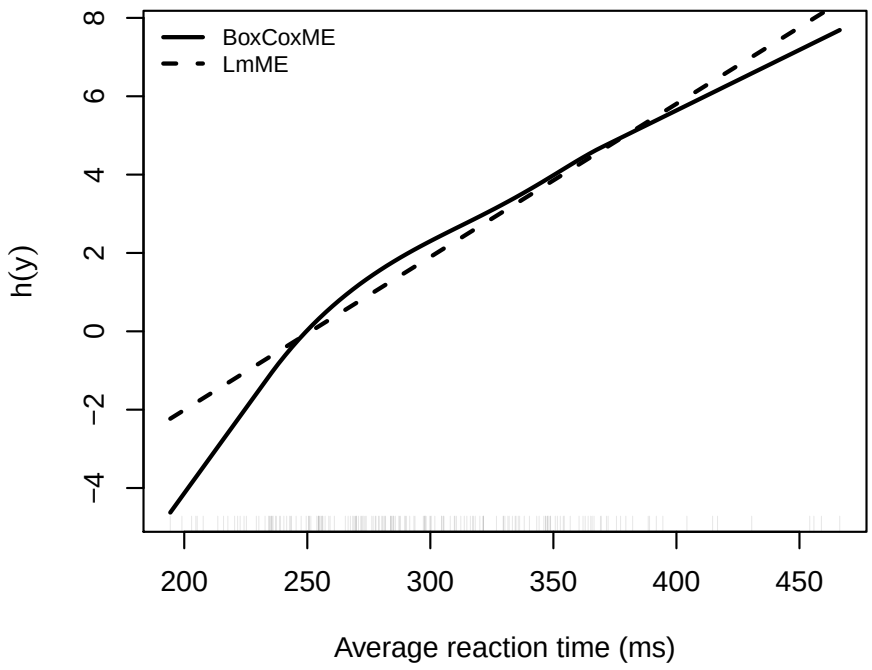

Figure 2: Comparison of baseline trasformations from the normal linear (`LmME`) and the Box-Cox-type (`BoxCoxME`) mixed-effects models fitted to the `sleepstudy` dataset.

Manuguerra and Heller (2010) proposed a mixed-effects logistic regression model for bounded, continuous measurements of pain levels in a randomized, double-blind, placebo-controlled trial of low-level laser therapy for subjects with chronic neck pain presented by Chow et al. (2006). The levels of pain, measured on a visual analog scale, and normalized between 0 and 1, are plotted in Figure 4 for each subject at the different follow-up times.

The mixed-effects model suggested by Manuguerra and Heller (2010) parameterizes the log-odds of experiencing smaller pain levels as a linear function of fixed and random effects and the baseline transformation. With the treatment group indicator, `laser`, and `time` denoting the follow-up times,

$$\log\left[\frac{\mathbb{P}\left(\texttt{pain} \leq y \mid \texttt{laser}, \texttt{time}, \alpha_i\right)}{\mathbb{P}\left(\texttt{pain} > y \mid \texttt{laser}, \texttt{time}, \alpha_i\right)}\right] = h(y) + \beta_{\text{Active}} + \beta_{7\text{w}} + \beta_{12\text{w}} + \beta_{7\text{w, Active}} + \beta_{12\text{w, Active}} + \alpha_i$$
$$\alpha_i \sim \mathcal{N}(0, \tau^2),$$

where $h(y)$ is an increasing function of the outcome. Rearranging the terms in the model above reveals that this indeed is a mixed-effects transformation model, with the distribution function of the standard logistic distribution ("expit" function) as F_Z

$$\begin{aligned}\mathbb{P}\left(\texttt{pain} \leq y \mid \texttt{laser}, \texttt{time}, \alpha_i\right) = \text{expit}\,\big(&h(y) + \beta_{\text{Active}} + \beta_{7\text{w}} + \beta_{12\text{w}} \\ &+ \beta_{7\text{w,Active}} + \beta_{12\text{w, Active}} + \alpha_i\big) \\ \alpha_i \sim \mathcal{N}(0, \tau^2)\end{aligned} \tag{5}$$

The `ColrME()` function of the `tramME` package estimates mixed-effects continuous outcome logistic regression models using Bernstein polynomials to approximate $h(y)$. Applying this model to the `neck_pain` dataset:

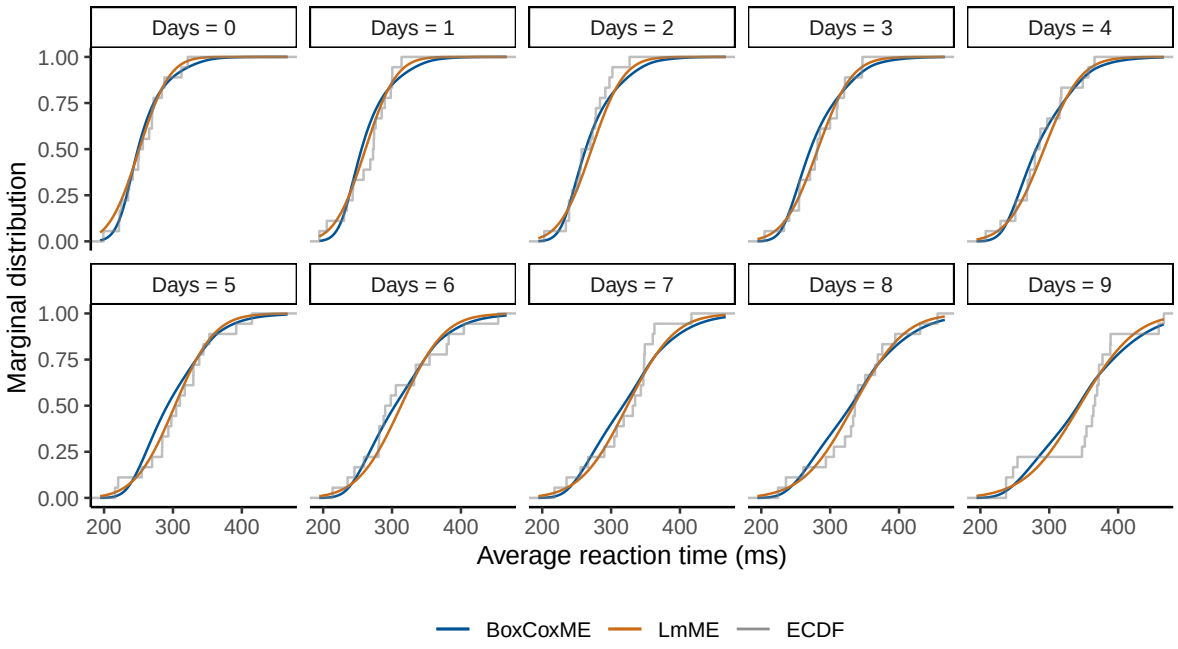

Figure 3: Comparison of the marginal distributions implied by the normal linear(`LmME`) and the Box-Cox-type (`BoxCoxME`) mixed-effects models fitted to the sleep deprivation study dataset. The empirical cumulative distribution functions (ECDF) are also plotted, conditionally on the days of sleep deprivation.

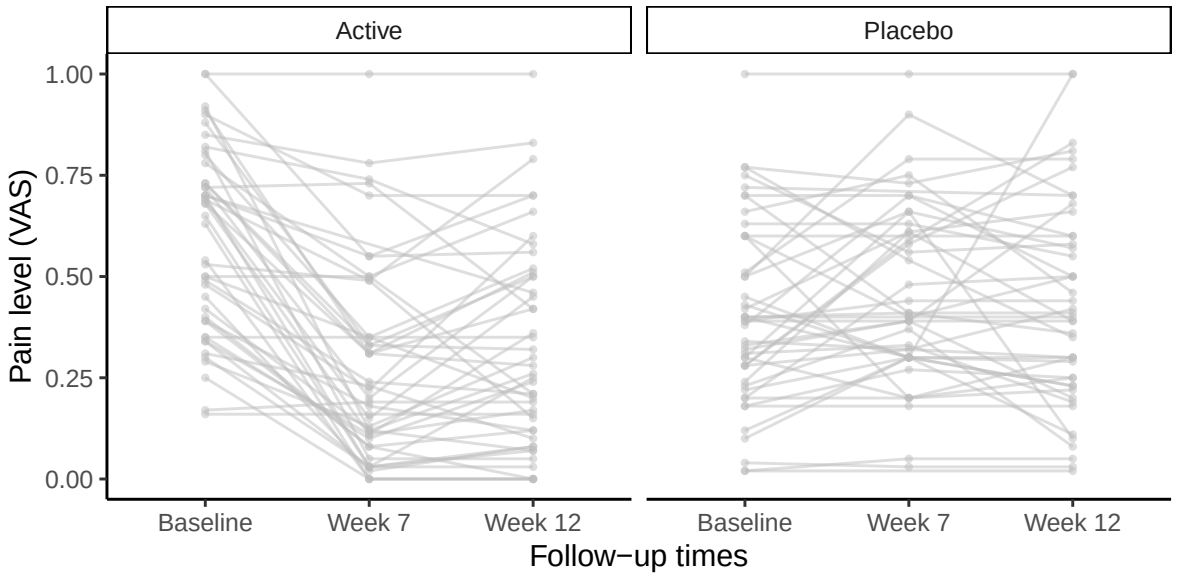

Figure 4: Neck pain dataset: Trajectories of pain levels measured on a visual analog scale (VAS) in the active treatment and placebo-controlled groups reported in Chow et al. (2006).

```
R> neck_tr <- ColrME(vas ~ laser * time + (1 | id), data = neck_pain,
+                    bounds = c(0, 1), support = c(0, 1))
```

Notice that we explicitly set the bounds and the support of the outcome variable to $[0, 1]$ because the pain levels are measured on a bounded scale.

The ordinalCont package by Manuguerra and Heller (2019) implements an alternative formulation of the model (5) based on the method described in Manuguerra et al. (2017). In their approach, the baseline transformation is parameterized using B-splines and the estimation is carried out in a penalized likelihood framework.

```
R> library("ordinalCont")
R> neck_ocm <- ocm(vas ~ laser * time + (1 | id), data = neck_pain, scale = c(0, 1))
```

Figure 5 compares the results of the mixed-effects transformation model approach to the estimates obtained using the **ordinalCont** package. Because the two models are not exactly the same, we see some differences in the parameter estimates as well as in the fitted baseline transformation functions, but the two model fits are reasonably close to each other.

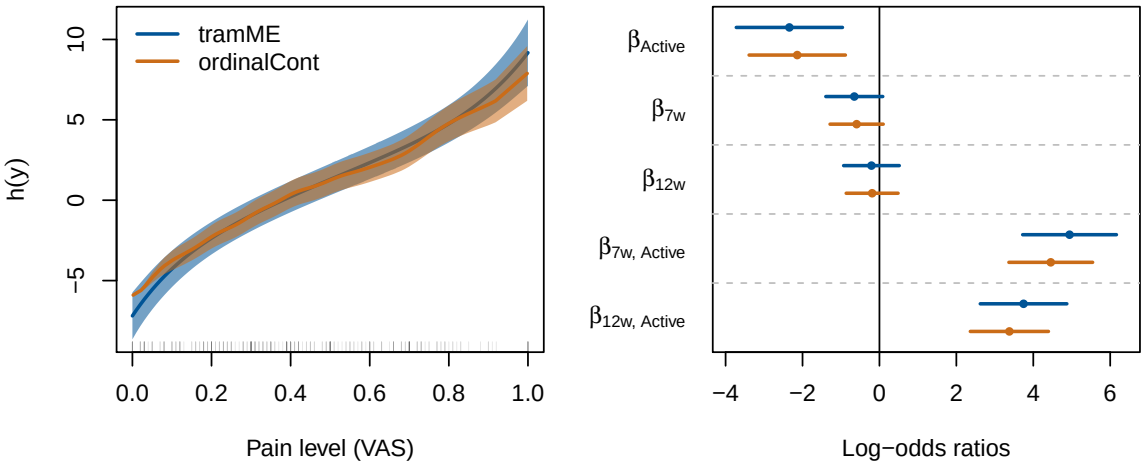

Figure 5: Left: Baseline transformations in continuous outcome logistic regressions estimated with the **tramME** and **ordinalCont** packages on the **neck_pain** dataset. The solid lines denote the point estimates and the areas indicate the 95% point-wise confidence intervals. Right: Coefficient estimates from **tramME** and **ordinalCont** packages and their 95% Wald confidence intervals.

The odds ratio estimates of the model fitted by **ColrME()**,

```
R> exp(coef(neck_tr))

      laser1        time2        time3 laser1:time2 laser1:time3
      0.0961       0.5200       0.8125     140.2076      42.4076
```

as well as the results from the **ordinalCont** package, suggest that there is an imbalance in the sample at baseline, *i.e.*, the odds of experiencing less pain in the active treatment group is only about 10% that of in the control group for any pain levels. Based on the estimates, the treatment has strong significant effect, especially at the seven-week follow-up, but seems to level off after 12 weeks.

If we want to compare the marginal distributions in the treatment and control groups directly, we have to average over the distribution of the random effects. Because in this example we only have a random intercept,

we have to evaluate a one-dimensional integral. We could use the same Monte Carlo method as we did in Section 3.2, or we can apply the adaptive quadrature method implemented in the `stats` package of R. The example below uses this approach to demonstrate the multiple options the analyst has in dealing with such problems.

```
R> jointCDF <- function(re, nd, mod) {
+    nd <- nd[rep(1, length(re)), ]
+    nd$id <- seq(nrow(nd)) ## to take vector-valued REs
+    pr <- predict(mod, newdata = nd, ranef = re, type = "distribution") *
+      dnorm(re, 0, sd = sqrt(varcov(mod)[[1]][1, 1]))
+    c(pr)
+  }
R> marginalCDF <- function(nd, mod) {
+    nd$cdf <- integrate(jointCDF, lower = -Inf, upper = Inf, nd = nd, mod = mod)$value
+    nd
+  }
R> nd <- expand.grid(vas = seq(0, 1, length.out = 100),
+                     time = unique(neck_pain$time),
+                     laser = unique(neck_pain$laser))
R> mp_colr <- parallel::mclapply(split(nd, seq(nrow(nd))),
+                                 marginalCDF, mod = neck_tr, mc.cores = 8)
R> mp_colr <- do.call("rbind", mp_colr)
```

Figure 6 compares the marginal distributions at different time points, and confirms our previous conclusions on baseline imbalance and treatment effect dynamics.

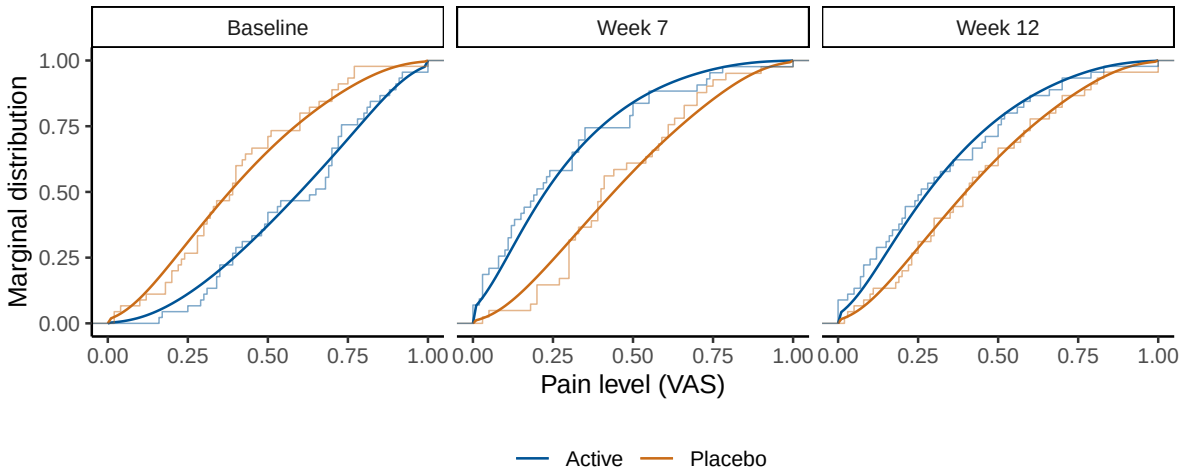

Figure 6: Comparison of marginal distributions, calculated from a mixed-effects continuous outcome logistic regression model, in the treatment (Active) and control (Placebo) groups at baseline and the two follow-up times. The step functions represent the empirical cumulative distribution functions of the specific groups.

3.4 Mixed-effects transformation models for time-to-event outcomes

Consider the example dataset `eortc` in the `coxme` package by Therneau (2019). The simulated dataset emulates the structure of the outcomes of a breast cancer trial by the European Organization for Research and Treatment of Cancer, and consists of 2323, possibly right censored, data points from 37 enrolling centers. We define

a proportional hazards mixed-effects model with random center ($i = 1, \ldots, 37$) and treatment (`trt`) effects (nested within centers and indexed by $j = 0, 1$).

$$\mathbb{P}\left(Y \leq y \mid \texttt{trt}, \alpha_i, \beta_{j(i)}\right) = 1 - \exp\left(-\exp\left(h(y) + \beta_{\texttt{trt}} + \alpha_i + \beta_{j(i)}\right)\right) \tag{6}$$
$$\alpha_i \sim \mathcal{N}(0, \tau_1^2), \quad \beta_{j(i)} \sim \mathcal{N}(0, \tau_2^2)$$

This model corresponds to a mixed-effects transformation model with the *minimum extreme value distribution* as the error distribution. Treating the baseline transformation as a general smooth function, approximated using Bernstein polynomials, we get the fully parametric version of the Cox proportional hazards model with normally distributed random effects.

We can fit this model with the `CoxphME()` function of `tramME`.

```
R> data("eortc", package = "coxme")
R> eortc$trt <- factor(eortc$trt, levels = c(0, 1))
R> eortc_cp <- CoxphME(Surv(y, uncens) ~ trt + (1 | center/trt), data = eortc,
+                      log_first = TRUE, order = 10)
```

The nested random effects structure is defined with the `/` operator. The `log_first = TRUE` option casts the outcome variable to the log-scale before defining the Bernstein bases, which usually improves the model fit when dealing with skewed conditional distributions, while we explicitly set the order of the Bernstein polynomials with `order = 10`. The confidence interval for the treatment effect (transformed to the hazard ratio scale) suggests evidence for the effectiveness of the treatment

```
R> exp(confint(eortc_cp, parm = "trt1", estimate = TRUE))

     lwr  upr  est
trt1 1.8 2.51 2.12
```

while the profile intervals of the random effects standard deviations indicate similar magnitude of center-level and treatment-level (within center) variabilities.

```
R> exp(confint(eortc_cp, pargroup = "ranef", type = "profile", estimate = TRUE,
+              ncpus = 2, parallel = "multicore"))

                          lwr   upr   est
trt:center|(Intercept) 0.0841 0.338 0.208
center|(Intercept)     0.0796 0.384 0.254
```

The transformation model framework by Hothorn (2020a) allows for stratification, *i.e.*, specifying separate transformation functions for different groups defined by a stratification factor. To check the appropriateness of the proportional hazards assumption between treatment and control groups visually, we re-estimate the model stratifying for the treatment indicator, *i.e.*, fitting baseline cumulative hazards for the treatment and control groups separately, and inspect whether the two baseline transformation functions, which are the log-cumulative baseline hazards when the error distribution is the minimum extreme value distribution, are parallel.

```
R> eortc_cp2 <- CoxphME(Surv(y, uncens) | 0 + trt ~ 0 + (1 | center/trt), data = eortc,
+                       log_first = TRUE, order = 10)
R> tr <- trafo(eortc_cp2, confidence = "interval")
```

Figure 7 plots the stratified transformation functions against log-time. The two curves are very close to parallel, which indicates that the treatment effect is constant over time, *i.e.*, the proportionality assumption is appropriate in the original model specification.

In addition to proportionality, Figure 7 reveals another important aspect of the data generating process: The fact that the baseline log-cumulative hazards are linear in log-time suggests that the conditional distributions are close to the Weibull distribution, *i.e.*, we can substitute the general baseline transformation function with

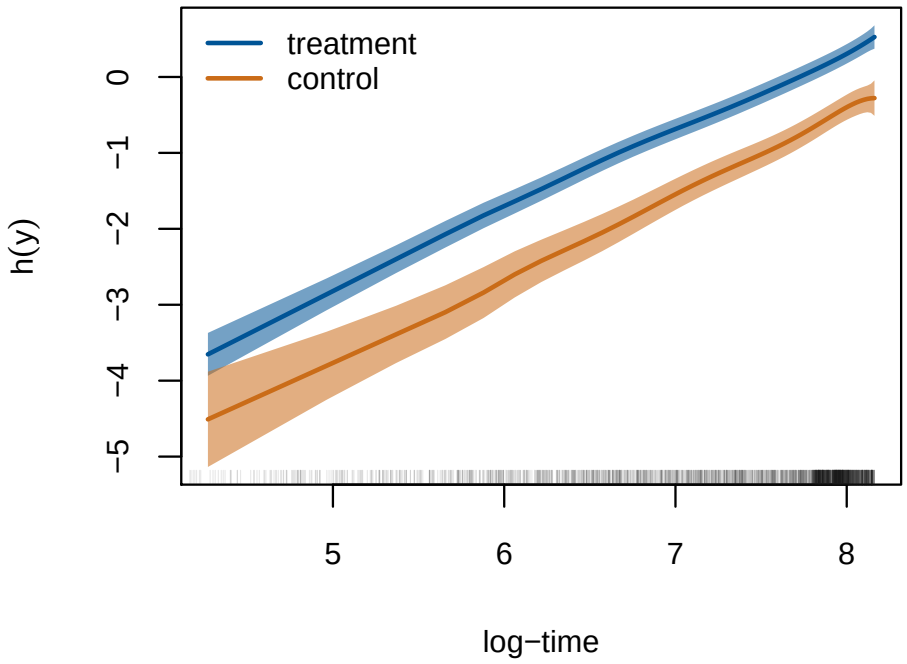

Figure 7: Comparison of baseline transfromation functions in treatment and control groups, estimated using a stratified parametric mixed-effects Cox proportional hazards model on the `eortc` dataset.

$h(y) = \vartheta_1 + \vartheta_2 \log(y)$. Flipping the signs of the fixed and random effects terms of (6) and substituting the log-linear function to the baseline transfromation, we get the model

$$\mathbb{P}\left(Y \leq y \mid \texttt{trt}, \alpha_i, \beta_{j(i)}\right) = 1 - \exp\left(-\exp\left(\vartheta_1 + \vartheta_2 \log(y) - \beta_{\texttt{trt}} - \alpha_i - \beta_{j(i)}\right)\right)$$
$$\alpha_i \sim \mathcal{N}(0, \tau_1^2), \quad \beta_{j(i)} \sim \mathcal{N}(0, \tau_2^2).$$

The `SurvregME()` function of the `tramME` package implements a variety of parametric mixed-effects models that represent specific choices of the error distribution and the baseline transformation function in the general formulation of Equation (1). There are several other R packages available for estimating parametric survival models with mixed effects, such as `parfm` by Munda et al. (2012) and `frailtypack` by Rondeau et al. (2012), but they typically do not allow for nested random-effects structures when assuming (log-)normally distributed frailty terms.

Fitting a mixed-effects Weibull model to the dataset `eortc`

```
R> eortc_w <- SurvregME(Surv(y, uncens) ~ trt + (1 | center/trt), data = eortc,
+                       dist = "weibull")
```

Comparing the parameter estimates of the Cox proportional hazards model to those from the mixed-effects Weibull model,

```
R> ## --- CoxphME
R> c(coef = coef(eortc_cp), se = sqrt(diag(vcov(eortc_cp, pargroup = "shift"))))

coef.trt1   se.trt1
```

```
   0.7535    0.0852
R> VarCorr(eortc_cp)


Grouping factor: trt:center (74 levels)
Standard deviation:
(Intercept)
      0.208

Grouping factor: center (37 levels)
Standard deviation:
(Intercept)
      0.254
R> ## --- SurvregME
R> c(coef = -coef(eortc_w), se = sqrt(diag(vcov(eortc_w, pargroup = "shift"))))
coef.trt1   se.trt1
   0.7531    0.0851
R> VarCorr(eortc_w)


Grouping factor: trt:center (74 levels)
Standard deviation:
(Intercept)
      0.208

Grouping factor: center (37 levels)
Standard deviation:
(Intercept)
      0.255
```

as well as their log-likelihood values

```
R> c(logLik(eortc_cp), logLik(eortc_w))
[1] -13027 -13032
```

confirms our suspicion that the dataset was indeed simulated from a conditional Weibull model.

Finally, we can compare the results from `tramME` to parameters estimated with the R package `coxme`.

```
R> library("coxme")
R> eortc_cm <- coxme(Surv(y, uncens) ~ trt + (1 | center/trt), data = eortc)
R> summary(eortc_cm)
Cox mixed-effects model fit by maximum likelihood
  Data: eortc
  events, n = 1463, 2323
  Iterations= 10 54
                NULL Integrated Fitted
Log-likelihood -10639     -10518 -10464

                  Chisq   df p AIC   BIC
Integrated loglik   242  3.0 0 236 220.4
 Penalized loglik   349 39.3 0 270  62.6
```

```
Model:  Surv(y, uncens) ~ trt + (1 | center/trt)
Fixed coefficients
      coef exp(coef) se(coef)    z p
trt1 0.742       2.1   0.0827 8.97 0

Random effects
 Group      Variable    Std Dev Variance
 center/trt (Intercept) 0.2045  0.0418
 center     (Intercept) 0.2627  0.0690
```

This package follows a different approach to estimate a mixed-effects Cox model by leaving the baseline hazards unspecified and maximizing the integrated partial likelihood. As a result, the parameter estimates are slightly different from the ones we got using the `CoxphME()` function, but the results are comparable and the conclusions are identical, nevertheless.

3.5 Mixed-effects transformation models for discrete ordinal outcomes

Our last example demonstrates how the mixed-effects transformation framework can be used in modeling correlated discrete ordinal outcomes. As an example, we take the soup tasting dataset by Christensen et al. (2011). The dataset contains 1847 observations from 185 respondents in a soup tasting experiment. The subjects were familiarized with a reference product prior the experiment, and, during the experiment, were asked to distinguish between samples from the reference product and test product using a six-level ordinal scale indicating their level of confidence. The scale ranges from "reference, sure" (`sureness` = 1) to "not reference, sure" (`sureness` = 6). Figure 8 presents the proportions of response categories for the test and reference samples, for respondent groups defined by how often they consume soup.

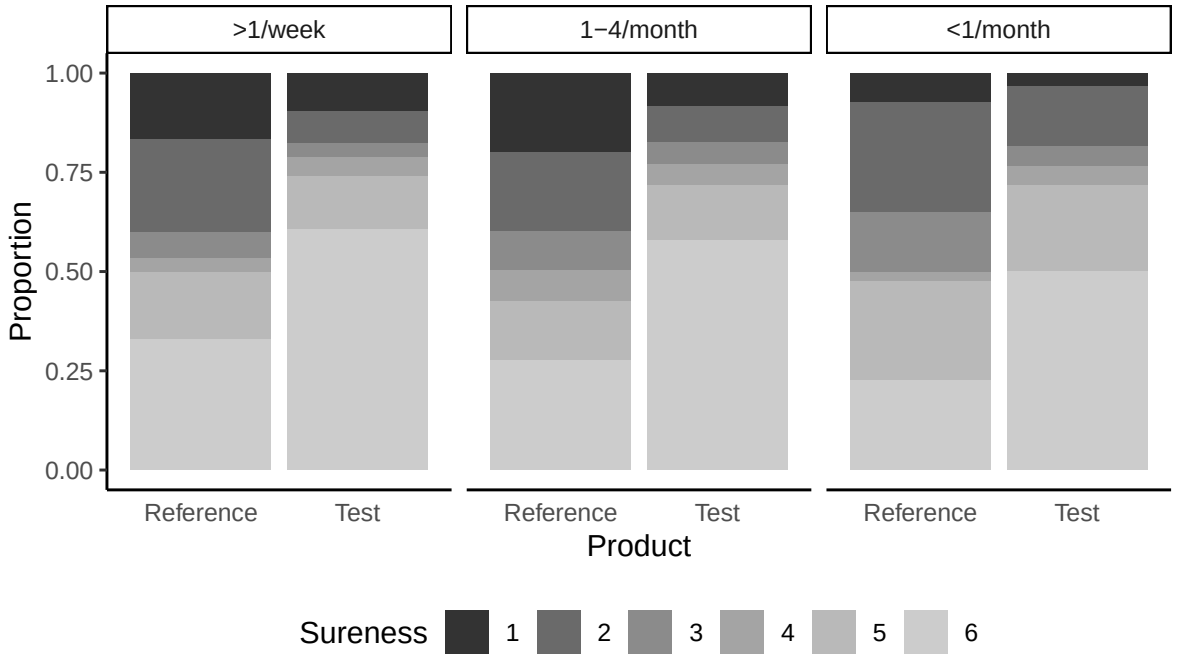

Figure 8: Data from the soup tasting study reported by Christensen et al. (2011): Proportions of responses of sureness levels (six levels, ranging from "reference, sure" to "not reference, sure") after tasting test and reference products. The data points are grouped by how often the respondents consume soup (more than once a week, one to four times a month, less than once a month).

Let us assume that we are interested in comparing the distributions of sureness ratings for reference products and test products, while taking the repeated nature of the design into account. Moreover, in doing so, we also want to control for how often the respondents usually consume soup (denoted by the covariate `freq`). With $k = 1 \dots 5$, indicating the sureness levels except the last one, $i = 1 \dots, 185$ indexing the respondents, and $j = 0, 1$ indexing the reference and test products (covariate `prod`), respectively, the regression model we estimate can be written as

$$\begin{aligned} \mathbb{P}\left(\texttt{sureness} \leq k \mid \texttt{prod}, \texttt{freq}, \alpha_i, \beta_{j(i)}\right) &= \Phi\left(\vartheta_k - \beta_{\text{test}} - \beta_{\text{1-4/month}} - \beta_{\text{<1/month}} - \alpha_i - \beta_{j(i)}\right) \\ \alpha_i \sim \mathcal{N}(0, \tau_1^2), &\quad \beta_{j(i)} \sim \mathcal{N}(0, \tau_2^2) \end{aligned} \tag{7}$$

The `PolrME()` function of the `tramME` package estimates models for ordered discrete outcomes. Depending on the choice of the error distribution, the user can fit proportional odds (logistic distribution), ordinal probit (standard normal distribution), proportional hazards (minimum extreme value distribution), or cumulative maximum extreme value models. In our example, we set `method = 'probit'` to estimate the probit model,

```
R> soup_pr <- PolrME(SURENESS ~ PROD + SOUPFREQ + (1 | RESP/PROD),
+                    data = soup, method = "probit")
R> logLik(soup_pr)

'log Lik.' -2666 (df=10)
```

The R package `ordinal` by Christensen (2019) also implements mixed-effects regression models for ordered discrete outcomes. As a cross-check, we can re-estimate the same model with the function `clmm()`,

```
R> library("ordinal")
R> soup_or <- clmm(SURENESS ~ PROD + SOUPFREQ + (1 | RESP/PROD), data = soup,
+                  link = "probit")
R> logLik(soup_or)

'log Lik.' -2666 (df=10)
```

Based on the likelihood values and the parameter estimates,

```
R> max(abs(coef(soup_or) - coef(soup_pr, with_baseline = TRUE)))

[1] 1.76e-05
```

the results are essentially the same.

We can introduce non-proportional effects in the transformation model framework by stratifying on a covariate. In our example, we might want to extend the model to allow for different effect sizes of the soup consumption frequency covariate, depending on the level of the outcome variable. Rewriting model (7),

$$\begin{aligned} \mathbb{P}\left(\texttt{sureness} \leq k \mid \texttt{prod}, \texttt{freq}, \alpha_i, \beta_{j(i)}\right) &= \Phi\left(\vartheta_k - \beta_{\text{test}} - \beta_{\text{1-4/month},k} - \beta_{\text{<1/month},k} - \alpha_i - \beta_{j(i)}\right) \\ \alpha_i \sim \mathcal{N}(0, \tau_1^2), &\quad \beta_{j(i)} \sim \mathcal{N}(0, \tau_2^2) \end{aligned}$$

and estimating it with `tramME` by stratifying for the soup frequency factor

```
R> soup_pr2 <- PolrME(SURENESS | SOUPFREQ ~ PROD + (1 | RESP/PROD),
+                     data = soup, method = "probit")
R> logLik(soup_pr2)

'log Lik.' -2655 (df=18)
```

The likelihood ratio test comparing the two specifications suggests some evidence that the extended, partially proportional model fits the data better.

```
R> anova(soup_pr, soup_pr2)
```

```
Model comparison

 Model 1: SURENESS ~ PROD + SOUPFREQ + (1 | RESP/PROD)
 Model 2: SURENESS | SOUPFREQ ~ PROD + (1 | RESP/PROD)

        Df logLik  AIC  BIC Chisq Chisq df Pr(>Chisq)
Model 1 10  -2666 5352 5408
Model 2 18  -2655 5347 5446  21.9        8     0.0051 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4 Discussion

Building the implementation of mixed-effects transformation models on the package `TMB` leads to significant efficiency gains in the computationally intensive steps of the maximum likelihood estimation. This computational efficiency is partly due to the use of Laplace approximation to integrate over the vector of random effects. However, several sources point out that Laplace's method can lead to biased estimates in some distributional settings. Pinheiro and Chao (2006) provide detailed numerical comparisons of the Laplacian approximation to adaptive Gaussian quadrature algorithms in the context of multilevel generalized models. Joe (2008) evaluates the method in the case of discrete outcome mixed-effects models, and concludes that the inaccuracy increases with the amount of discreteness of the response variable and decreases as the cluster sizes increase.

It is worth mentioning that the conditional approach of modeling the distribution of the response, which is the basis of the transformation models implemented in the `tramME` package, is not the only way one could approach the problem of correlated outcomes in regression settings. The main alternative to a conditional (mixed-effects) modeling approach is a marginal model that parameterizes the marginal distribution of the outcome and treats the covariance structure as nuisance parameters. Generalized estimating equations (GEE, Hardin and Hilbe, 2013) models represent prominent examples of such an approach. Proponents of marginal models point out that, in a conditional model, the fixed effects parameter estimates cannot be interpreted as population averages, which is usually of primary interest in a regression analysis. Lindsey and Lambert (1998) emphasize that marginal parameter estimates from longitudinal studies can only be interpreted as population averages when the participants are representative to their populations, which is usually not the case. Moreover, they argue that defining models based on marginal distributions very often leads to complicated and implausible conditional distributions, whereas conditional models can more easily express physiologically plausible mechanisms on the level of the individual. Lee and Nelder (2004) argue that conditional models are more fundamental as they allow for both marginal and conditional inferences, which is not true in the case of marginal models. As we demonstrated in Sections 3.2 and 3.3, the marginal distributions implied by the conditional transformation model can be easily approximated using numerical techniques.

The `tramME` package, introduced in this article, extends the available options for modeling grouped data structures with mixed-effects regressions in several ways: Through its dense code base, `tramME` provides an unified and efficient estimation framework for a broad range of regression models. Examples in Section 3 demonstrate that, using this single package, several very specific regression problems can be addressed. Relying only on a limited number of packages, in turn, decreases the likelihood of errors in the statistical analysis. As the examples showed, the modular structure of our approach naturally leads to extensions of existing models (such as accounting for censoring or introducing nested or crossed random effects structures) that would otherwise require a lot of effort to re-implement from scratch. Moreover, the underlying theory of linear transformation models provides a flexible basis for the implementation of the package and for its future extensions.

References

Amani Almohaimeed and Jochen Einbeck. *boxcoxmix: Box-Cox-Type Transformations for Linear and Logistic Models with Random Effects*, 2018. URL https://CRAN.R-project.org/package=boxcoxmix. R package version 0.21.

Douglas Bates, Martin Mächler, Benjamin M. Bolker, and Steve Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi:10.18637/jss.v067.i01.

Gregory Belenky, Nancy J. Wesensten, David R. Thorne, Maria L. Thomas, Helen C. Sing, Daniel P. Redmond, Michael B. Russo, and Thomas J. Balkin. Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: a sleep dose-response study. *Journal of Sleep Research*, 12(1):1–12, 2003. doi:10.1046/j.1365-2869.2003.00337.x.

George E. P. Box and David R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964. doi:10.1111/j.2517-6161.1964.tb00553.x.

Mollie E. Brooks, Kasper Kristensen, Koen J. van Benthem, Arni Magnusson, Casper W. Berg, Anders Nielsen, Hans J. Skaug, Martin Mächler, and Benjamin M. Bolker. glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling. *The R Journal*, 9(2):378–400, 2017. doi:10.32614/RJ-2017-066.

Roberta T. Chow, Gillian Z. Heller, and Les Barnsley. The effect of 300 mW, 830 nm laser on chronic neck pain: a double-blind, randomized, placebo-controlled study. *Pain*, 124(1-2):201–210, 2006. doi:10.1016/j.pain.2006.05.018.

Rune Haubo Bojesen Christensen. *ordinal—Regression Models for Ordinal Data*, 2019. URL http://www.cran.r-project.org/package=ordinal/. R package version 2019.4-25.

Rune Haubo Bojesen Christensen, Graham Cleaver, and Per Bruun Brockhoff. Statistical and Thurstonian models for the A-not A protocol with and without sureness. *Food Quality and Preference*, 22(6):542–549, 2011. doi:10.1016/j.foodqual.2011.03.003.

Eugene Demidenko. *Mixed models: theory and applications with R*. Wiley series in probability and statistics. Wiley, second edition, 2013. doi:10.1002/9781118651537.

James W. Hardin and Joseph M. Hilbe. *Generalized estimating equations*. CRC Press, second edition, 2013. doi:10.1201/b13880.

Torsten Hothorn. *Transformation Models: The tram Package*, 2020a. URL https://CRAN.R-project.org/package=tram. R package vignette version 0.3-2.

Torsten Hothorn. Most likely transformations: The mlt package. *Journal of Statistical Software*, 92(1):1–68, 2020b. doi:10.18637/jss.v092.i01.

Torsten Hothorn, Lisa Möst, and Peter Bühlmann. Most likely transformations. *Scandinavian Journal of Statistics*, 45(1):110–134, 2018. doi:10.1111/sjos.12291.

Harry Joe. Accuracy of Laplace approximation for discrete response mixed models. *Computational Statistics & Data Analysis*, 52(12):5066–5074, 2008. doi:10.1016/j.csda.2008.05.002.

Kasper Kristensen, Anders Nielsen, Casper W. Berg, Hans Skaug, and Bradley M. Bell. TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, 70(5):1–21, 2016. doi:10.18637/jss.v070.i05.

Youngjo Lee and John A. Nelder. Conditional and Marginal Models: Another View. *Statistical Science*, 19(2): 219–238, 2004. doi:10.1214/088342304000000305.

James K. Lindsey and Philippe Lambert. On the appropriateness of marginal models for repeated measurements in clinical trials. *Statistics in Medicine*, 17(4):447–469, 1998. doi:10.1002/(SICI)1097-0258(19980228)17:4<447::AID-SIM752>3.0.CO;2-G.

Tina Lohse, Sabine Rohrmann, David Faeh, and Torsten Hothorn. Continuous outcome logistic regression for analyzing body mass index distributions. *F1000Research*, 6:1933, 2017. doi:10.12688/f1000research.12934.1.

Maurizio Manuguerra and Gillian Heller. Ordinal regression models for continuous scales. *The International Journal of Biostatistics*, 6:14–14, 2010. doi:10.2202/1557-4679.1230.

Maurizio Manuguerra and Gillian Heller. *ordinalCont: Ordinal Regression Analysis for Continuous Scales*, 2019. URL https://CRAN.R-project.org/package=ordinalCont. R package version 2.0.1.

Maurizio Manuguerra, Gillian Heller, and Jun Ma. Semi-parametric ordinal regression models for continuous scales. In Marco Grzegorczyk and Giacomo Ceoldo, editors, *Proceedings of the 32nd International Workshop on Statistical Modelling*, volume 1, pages 236–241, 2017.

Marco Munda, Federico Rotolo, and Catherine Legrand. parfm: Parametric frailty models in R. *Journal of Statistical Software*, 51(11):1–20, 2012. doi:10.18637/jss.v051.i11.

Jose Pinheiro and Edward C. Chao. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 15(1):58–81, 2006. doi:10.1198/106186006X96962.

Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Core Team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2019. URL https://CRAN.R-project.org/package=nlme. R package version 3.1-140.

Virginie Rondeau, Yassin Mazroui, and Juan R. Gonzalez. frailtypack: An R package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametrical estimation. *Journal of Statistical Software*, 47(4):1–28, 2012. doi:10.18637/jss.v047.i04.

Terry M. Therneau. *A Package for Survival Analysis in S*, 2015. URL https://CRAN.R-project.org/package=survival. version 2.38.

Terry M. Therneau. *coxme: Mixed Effects Cox Models*, 2019. URL https://CRAN.R-project.org/package=coxme. R package version 2.2-14.

Yuqi Tian, Torsten Hothorn, Chun Li, Frank E. Harrell Jr., and Bryan E. Shepherd. An empirical comparison of two novel transformation models. *Statistics in Medicine*, 39(5):562–576, 2020. doi:10.1002/sim.8425.

Affiliations:
Bálint Tamási and Torsten Hothorn
Institut für Epidemiologie, Biostatistik und Prävention
Universität Zürich
Hirschengraben 84, CH-8001 Zürich
Switzerland
balint.tamasi@uzh.ch, Torsten.Hothorn@R-project.org