

Some Applications of Marginally Interpretable Linear Transformation Models for Clustered Observations

Torsten Hothorn
Universität Zürich

Abstract

Owing to their generality, transformation models can be used to set-up and compute many interesting regression models for discrete and continuous responses. This document focuses on the analysis of clustered observations. Marginal predictive distributions are defined by transformation models and their joint normal distribution depends on a structured covariance matrix. Applications with skewed, bounded, and survival continuous outcomes as well as binary and ordered categorical responses are presented. Data is analysed by a proof-of-concept implementation of parametric linear transformation models for clustered observations available in the **tram** add-on package to the R system for statistical computing.

Keywords: conditional mixed models, marginal models, marginal predictive distributions, survival analysis, categorical data analysis.

1. Introduction

The purpose of this document is to compare marginally interpretable linear transformation models for clustered observations (Hothorn 2019) to conventional conditional formulations of mixed-effects models where such an overlap exists. In addition, novel transformation models going beyond the capabilities of conventional mixed-effects models are estimated and interpreted. A proof-of-concept implementation (meaning: the algorithms work but need optimisation, the user interface is very rough) available in package **tram** (Hothorn and Barbanti 2020) is applied. The results presented in this document can be reproduced from the **mtram** package vignette

```
R> install.packages("tram")  
R> vignette("mtram", package = "tram")
```

2. Normal and Non-normal Mixed-effects Models

First we consider mixed-effects models for reaction times in the sleep deprivation study (Be-

Please cite this document as: Torsten Hothorn (2020) Some Applications of Marginally Interpretable Linear Transformation Models for Clustered Observations. R package vignette version 0.5-2, URL <https://CRAN.R-project.org/package=tram>.

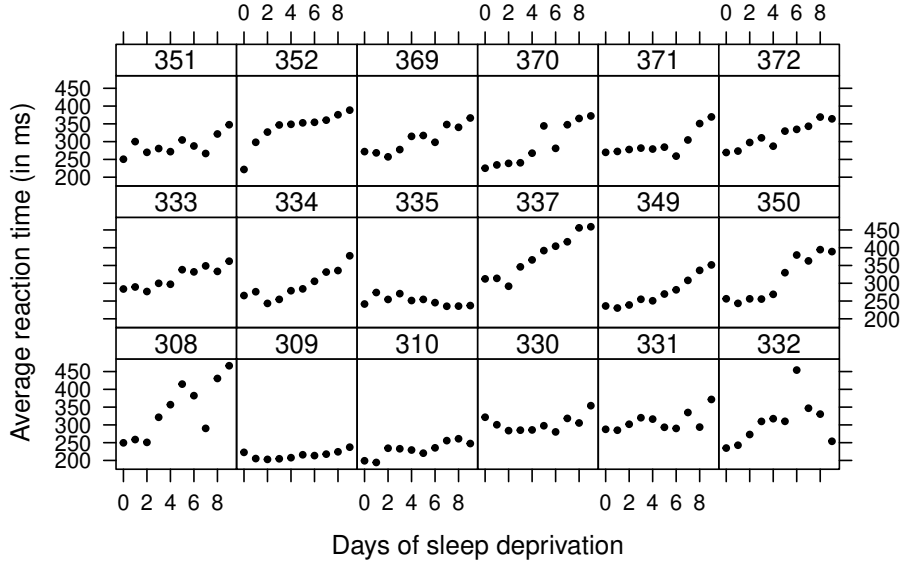


Figure 1: Sleep deprivation: Average reaction times to a specific task over several days of sleep deprivation for 18 subjects from [Belenky *et al.* \(2003\)](#).

[lenky *et al.* 2003](#)). The average reaction times to a specific task over several days of sleep deprivation are given for 18 subjects in Figure 1.

The classical normal linear random-intercept/random-slope model, treating the study participants as independent observations, is fitted by maximum likelihood to the data using the `lmer()` function from the `lme4` add-on package ([Bates *et al.* 2015](#)):

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

The corresponding conditional model for subject i reads

$$\mathbb{P}(\text{Reaction} \leq y \mid \text{day}, i) = \Phi\left(\frac{y - \alpha - \beta \text{day} - \alpha_i - \beta_i \text{day}}{\sigma}\right), \quad (\alpha_i, \beta_i) \sim N_2(\mathbf{0}, \mathbf{G}(\boldsymbol{\gamma}))$$

with $\sigma^{-2}\mathbf{G} = \boldsymbol{\Lambda}(\boldsymbol{\gamma})\boldsymbol{\Lambda}(\boldsymbol{\gamma})^\top$ and

$$\boldsymbol{\Lambda}(\boldsymbol{\gamma}) = \begin{pmatrix} \gamma_1 & 0 \\ \gamma_2 & \gamma_3 \end{pmatrix}, \quad \boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^\top.$$

The same model, however using the alternative parameterisation and an independent (of `lme4`, only the `update()` method for Cholesky factors is reused) gradient-based maximisation of the log-likelihood, is estimated in a two-step approach as

```
R> sleep_LM <- Lm(Reaction ~ Days, data = sleepstudy)
R> sleep_LMmer <- mtram(sleep_LM, ~ (Days | Subject), data = sleepstudy)
```

The first call to `Lm()` computes the equivalent of a normal linear regression model parameterised as a linear transformation model *ignoring* the longitudinal nature of the observations. The purpose is to set-up the necessary model infrastructure (model matrices, inverse link functions, etc.) and to compute reasonable starting values for the fixed effects. The second call to `mtram()` specifies the random effects structure (here a correlated pair of random intercept for subject and random slope for days) and optimises the likelihood for all model parameters $\vartheta_1, \tilde{\alpha}, \tilde{\beta}$, and γ in the model (here also looking at the conditional model for subject i)

$$\mathbb{P}(\text{Reaction} \leq y \mid \text{day}, i) = \Phi \left(\vartheta_1 y + \tilde{\alpha} - \tilde{\beta} \text{day} - \tilde{\alpha}_i - \tilde{\beta}_i \text{day} \right), \quad (\tilde{\alpha}_i, \tilde{\beta}_i) \sim N_2(\mathbf{0}, \mathbf{\Lambda}(\gamma)\mathbf{\Lambda}(\gamma))$$

that is, all fixed and random effect parameters are divided by the residual standard deviation σ (this is the reparameterisation applied by `Lm()`). Of course, the parameter ϑ_1 , the inverse residual standard deviation, is ensured to be positive via an additional constraint in the optimiser maximising the log-likelihood.

The log-likelihoods of the two models fitted by `lmer()` and `mtram()` are very close

```
R> logLik(sleep_lmer)
```

```
'log Lik.' -875.9697 (df=6)
```

```
R> logLik(sleep_LMmer)
```

```
'log Lik.' -875.9697 (df=6)
```

Looking at the model coefficients, the two procedures lead to almost identical inverse residual standard deviations

```
R> (sdinv <- 1 / summary(sleep_lmer)$sigma)
```

```
[1] 0.03907485
```

```
R> coef(sleep_LMmer)["Reaction"]
```

```
Reaction
0.03907741
```

and fixed effects (the slope can be interpreted as inverse coefficient of variation)

```
R> fixef(sleep_lmer) * c(-1, 1) * sdinv
```

```
(Intercept)      Days
-9.8236175     0.4090077
```

```
R> coef(sleep_LMmer)[c("(Intercept)", "Days")]
```

```
(Intercept)      Days
-9.8243917     0.4089265
```

The random-effect parameters γ are also reasonably close

```
R> sleep_lmer@theta
[1] 0.92919061 0.01816575 0.22264321

R> coef(sleep_LMmer)[- (1:3)]

      gamma1      gamma2      gamma3
0.92901066 0.01843056 0.22280431
```

Consequently, the variance-covariance and correlation matrices

```
R> sleep_LMmer$G * (1 / sdiv)^2

2 x 2 sparse Matrix of class "dsCMatrix"

[1,] 565.2580 11.21410
[2,] 11.2141 32.73513

R> cov2cor(sleep_LMmer$G * (1 / sdiv)^2)

2 x 2 sparse Matrix of class "dsCMatrix"

[1,] 1.00000000 0.08243925
[2,] 0.08243925 1.00000000

R> unclass(VarCorr(sleep_lmer))$Subject

      (Intercept)      Days
(Intercept) 565.47697 11.05512
Days        11.05512 32.68179
attr(,"stddev")
(Intercept)      Days
 23.779760    5.716799
attr(,"correlation")
      (Intercept)      Days
(Intercept) 1.00000000 0.08132109
Days        0.08132109 1.00000000
```

are practically equivalent. This result indicates the correctness of the alternative implementation of normal linear mixed-effects models in the transformation model framework: `mtram()` reuses some infrastructure from `lme4` and `Matrix`, most importantly fast update methods for Cholesky factors, but the likelihood and corresponding optimisation relies on an independent implementation. So why are we doing this? Because `mtram()` is able to deal with models or likelihoods not available in `lme4`, for example the likelihood for interval-censored observations. Let's assume that the timing of the reaction times was less accurate than suggested by the numerical representation of the results. The following code

```
R> library("survival")
R> sleepstudy$Reaction_I <- with(sleepstudy, Surv(Reaction - 20, Reaction + 20,
+                                             type = "interval2"))
R> sleepstudy$Reaction_I[1:5]

[1] [229.5600, 269.5600] [238.7047, 278.7047] [230.8006, 270.8006]
[4] [301.4398, 341.4398] [336.8519, 376.8519]
```

converts the outcome to interval-censored values, where each interval has length 40. The above mixed model can now be estimated by maximising the likelihood corresponding to interval-censored observations:

```
R> sleep_LM_I <- Lm(Reaction_I ~ Days, data = sleepstudy)
R> sleep_LMmer_I <- mtram(sleep_LM_I, ~ (Days | Subject), data = sleepstudy)
```

Of course, the log-likelihood changes (because this is a log-probability and not a log-density of a continuous distribution) but the parameter estimates are reasonably close

```
R> logLik(sleep_LMmer_I)
```

```
'log Lik.' -214.9675 (df=6)
```

```
R> coef(sleep_LMmer_I)
```

```
(Intercept) Reaction_I      Days      gamma1      gamma2      gamma3
-9.78770607  0.03900116  0.41633415  0.83398952  0.07584130  0.19038611
```

```
R> coef(sleep_LMmer)
```

```
(Intercept) Reaction      Days      gamma1      gamma2      gamma3
-9.82439168  0.03907741  0.40892652  0.92901066  0.01843056  0.22280431
```

The next question is if the normal assumption for reaction times is appropriate. In the transformation world, this assumption is simple to assess because we can easily (theoretically and in-silico) switch to the non-normal linear mixed-effects transformation model

$$\mathbb{P}(\text{Reaction} \leq y \mid \text{day}, i) = \Phi \left(h(y) - \tilde{\beta} \text{day} - \tilde{\alpha}_i - \tilde{\beta}_i \text{day} \right), \quad (\tilde{\alpha}_i, \tilde{\beta}_i) \sim N_2(\mathbf{0}, \mathbf{\Lambda}(\boldsymbol{\gamma})\mathbf{\Lambda}(\boldsymbol{\gamma}))$$

where $h(y) = \mathbf{a}(y)^\top \boldsymbol{\vartheta}$ represents a monotone non-decreasing transformation function. The function implementing such a more flexible model is named in honor of the first paper on the analysis of transformed responses by [Box and Cox \(1964\)](#) but it *does not* simply apply what is known as a Box-Cox transformation. Bernstein polynomials $h(y) = \mathbf{a}(y)^\top \boldsymbol{\vartheta}$ under suitable constraints are applied instead by

```
R> sleep_BC <- BoxCox(Reaction ~ Days, data = sleepstudy)
R> sleep_BCmer <- mtram(sleep_BC, ~ (Days | Subject), data = sleepstudy,
+                      Hessian = TRUE)
R> logLik(sleep_BCmer)
```

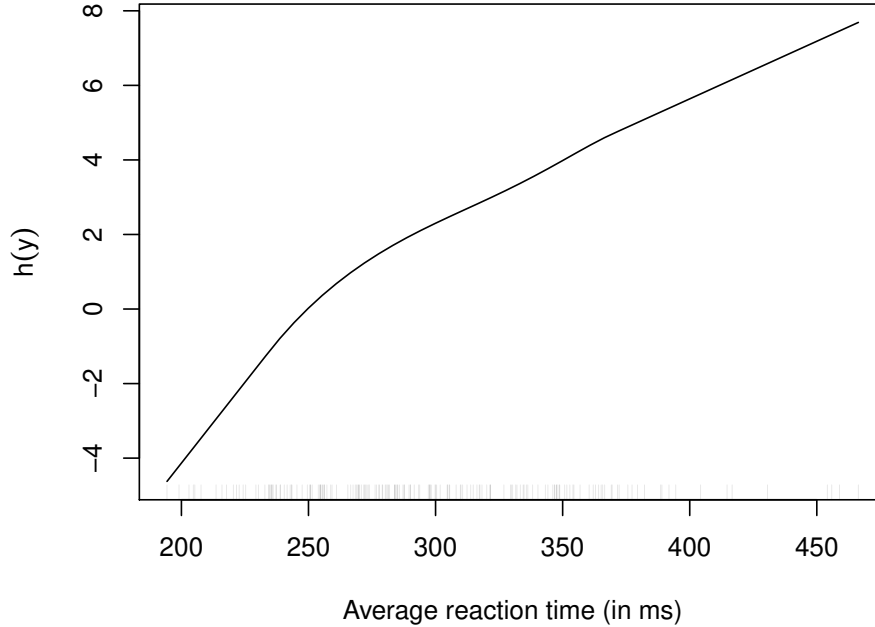


Figure 2: Sleep deprivation: Data-driven transformation \hat{h} of average reaction times to sleep deprivation. The non-linearity induces a non-normal marginal distribution function of reaction times.

'log Lik.' -859.5455 (df=11)

The increase in the log-likelihood compared to the normal model is not a big surprise. Plotting the transformation function $h(y) = \mathbf{a}(y)^\top \boldsymbol{\vartheta}$ as a function of reaction time can help to assess deviations from normality because the latter assumption implies a linear transformation function. Figure 2 clearly indicates that models allowing a certain skewness of reaction times will provide a better fit to the data. This might also not come as a big surprise to experienced data analysts.

However, what does this finding mean in terms of a direct comparison of the model and the data? Looking at the marginal cumulative distribution functions of average reaction time conditional on days of sleep deprivation in Figure 3 one finds that the non-normal marginal transformation models provided a better fit to the marginal empirical cumulative distribution functions than the normal marginal models. Especially for short reaction times in the first week of sleep deprivation, the yellowish marginal cumulative distribution is much closer to the empirical cumulative distribution function representing the marginal distribution of reaction times at each single day of study participation.

It should be noted that the small positive correlation between random intercept and random slope observed in the normal linear mixed-effects model turned into a negative correlation in this non-normal model

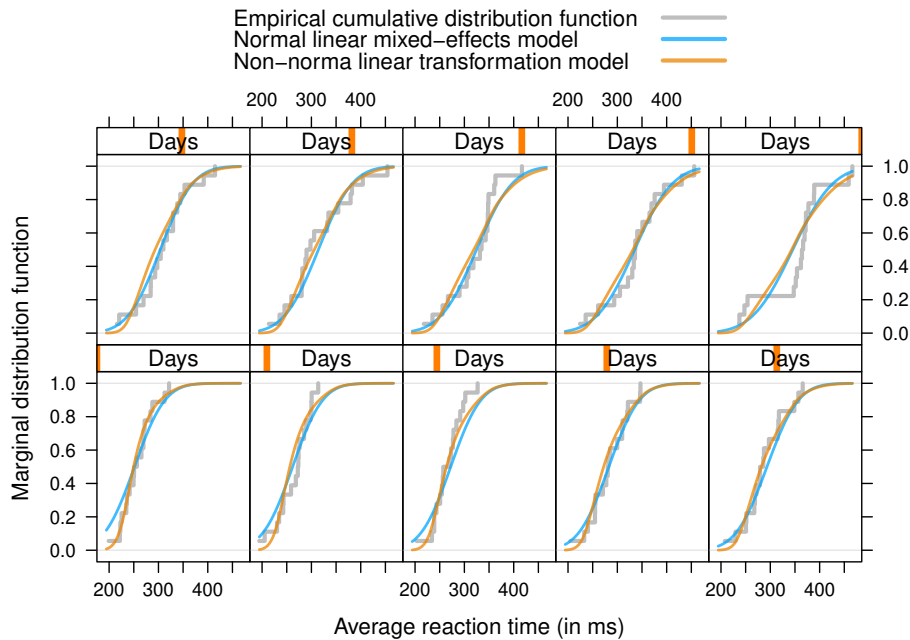


Figure 3: Sleep deprivation: Marginal distribution of reaction times, separately for each day of study participation. The grey step-function corresponds to the empirical cumulative distribution function, the blue line to the marginal cumulative distribution of a normal linear mixed-effects model, and the yellowish line to a non-normal linear mixed-effects transformation model.

```
R> cov2cor(sleep_BCmer$G)
```

```
2 x 2 sparse Matrix of class "dsCMatrix"
```

```
[1,] 1.0000000 -0.1946629
[2,] -0.1946629 1.0000000
```

What is the uncertainty associated with this parameter? The correlation is a non-linear function of γ and therefore the direct computation of confidence intervals questionable. However, we can extract an estimate of the covariance of the estimated model parameters from the model and, relying on the asymptotic normality of the maximum likelihood estimators, we can sample from the asymptotic distribution of the variance of the random intercept $\tilde{\alpha}$, the random slope $\tilde{\beta}$, and their correlation

```
R> VC <- solve(sleep_BCmer$Hessian)
R> idx <- (nrow(VC) - 2):nrow(VC)
R> Rcoef <- rmvnorm(1000, mean = coef(sleep_BCmer), sigma = VC)[,idx]
R> ret <- apply(Rcoef, 1, function(gamma) {
+   L <- matrix(c(gamma[1:2], 0, gamma[3]), nrow = 2)
+   V <- tcrossprod(L)
+   c(diag(V), cov2cor(V)[1,2])
+ })
```

The 95% confidence intervals

```
R> ### variance random intercept
R> quantile(ret[1,], c(.025, .5, .975))
```

```
      2.5%      50%      97.5%
0.9127821 2.5713595 5.2493469
```

```
R> ### variance random slope
R> quantile(ret[2,], c(.025, .5, .975))
```

```
      2.5%      50%      97.5%
0.01890987 0.05348231 0.10594879
```

```
R> ### correlation random intercept / random slope
R> quantile(ret[3,], c(.025, .5, .975))
```

```
      2.5%      50%      97.5%
-0.6193527 -0.1883314  0.4689778
```

indicate rather strong unobserved heterogeneity affecting the intercept and less pronounced variability in the slope. There is only weak information about the correlation of the two random effects contained in the data.

The downside of this approach is that, although the model is nicely interpretable on the scale of marginal or conditional distribution functions, the direct interpretation of the fixed effect $\tilde{\beta}$ is not very straightforward because it corresponds to the conditional mean *after* transforming the outcome. This interpretability issue can be addressed by exchanging the probit link to a logit link in Section 4.

3. Binary Probit Mixed-effects Models

Here we compare different implementations of binary probit mixed models for the notoriously difficult toe nail data (Backer *et al.* 1998). The outcome was categorised to two levels (this being probably the root of all troubles) and a conditional density plot (Figure 4) suggests an improvement in both treatment groups over time, however with a more rapid advance in patients treated with terbinafine.

The random intercept probit model fitted by Laplace and Adaptive Gauss-Hermite Quadrature (AGQ) approximations to the likelihood give quite different results:

```
R> ### Laplace
R> toenail_glmer_RI_1 <-
+   glmer(outcome ~ treatment * time + (1 | patientID),
+         data = toenail, family = binomial(link = "probit"),
+         nAGQ = 1)
R> summary(toenail_glmer_RI_1)
```

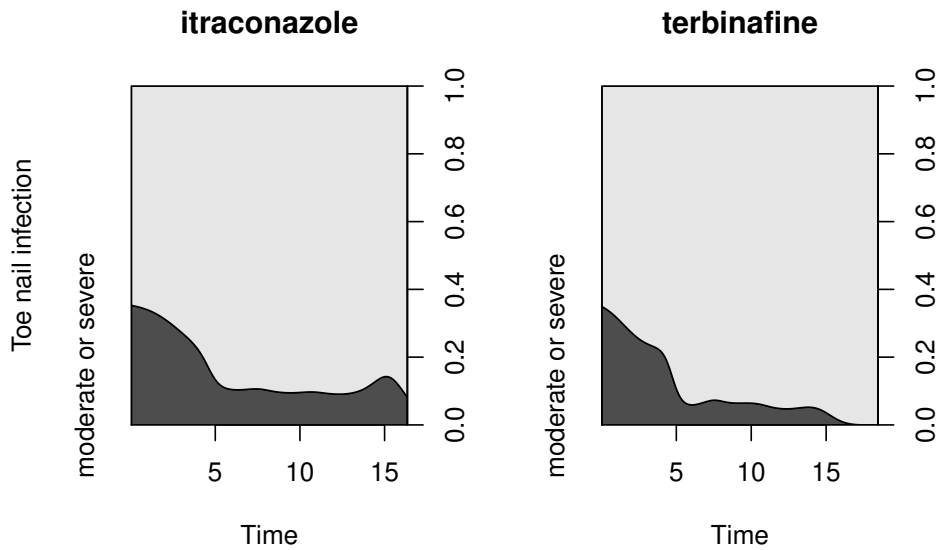



Figure 4: Toe nail data: Conditional density plot of two outcome classes (none or mild vs. moderate or severe) under two treatments.

```

Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) [glmerMod]
Family: binomial (probit)
Formula: outcome ~ treatment * time + (1 | patientID)
Data: toenail
    
```

AIC	BIC	logLik	deviance	df.resid
1279.0	1306.8	-634.5	1269.0	1898

```

Scaled residuals:
  Min      1Q  Median      3Q      Max
-3.507 -0.017 -0.004  0.000  54.046
    
```

```

Random effects:
 Groups   Name      Variance Std.Dev.
patientID (Intercept) 20.68    4.548
Number of obs: 1903, groups: patientID, 289
    
```

```

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.39650    0.22091 -15.375  <2e-16 ***
treatmentterbinafine -0.01532    0.25359  -0.060  0.9518
time          -0.21749    0.02256  -9.639  <2e-16 ***
treatmentterbinafine:time -0.07155    0.03425  -2.089  0.0367 *
---
    
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

```
(Intr) trtmnt time
trtmnttrbnf -0.593
time        -0.009  0.102
trtmnttrbn:  0.093 -0.143 -0.629
```

```
R> toenail_glmer_RI_1@theta
```

```
[1] 4.547891
```

```
R> ### Adaptive Gaussian Quadrature
```

```
R> toenail_glmer_RI_2 <-
+   glmer(outcome ~ treatment * time + (1 | patientID),
+         data = toenail, family = binomial(link = "probit"),
+         nAGQ = 20)
R> summary(toenail_glmer_RI_2)
```

```
Generalized linear mixed model fit by maximum likelihood (Adaptive
  Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
Family: binomial (probit)
Formula: outcome ~ treatment * time + (1 | patientID)
Data: toenail
```

AIC	BIC	logLik	deviance	df.resid
1277.8	1305.6	-633.9	1267.8	1898

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.847	-0.189	-0.078	-0.001	33.997

Random effects:

Groups	Name	Variance	Std.Dev.
patientID	(Intercept)	4.485	2.118

Number of obs: 1903, groups: patientID, 289

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.93061	0.23176	-4.015	5.93e-05 ***
treatmentterbinafine	-0.07609	0.30921	-0.246	0.8056
time	-0.19074	0.02059	-9.263	< 2e-16 ***
treatmentterbinafine:time	-0.06419	0.03099	-2.071	0.0383 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

```

      (Intr) trtmnt time
trtmnttrbnf -0.655
time        -0.186  0.212
trtmnttrbn:  0.193 -0.287 -0.611

```

```
R> toenail_glmer_RI_2@theta
```

```
[1] 2.117846
```

The sequential reduction (SR) algorithm (Ogden 2015) gives results close to AGQ

```

R> library("glmmSR")
R> toenail_glmm_RI_3 <-
+   glmm(outcome ~ treatment * time + (1 | patientID),
+       data = toenail, family = binomial(link = "probit"),
+       method = "SR", control = list(nSL = 3))

```

Fitting the model..... done.

```
R> summary(toenail_glmm_RI_3)
```

Generalized linear mixed model fit by maximum likelihood [glmmFit]
Likelihood approximation: Sequential reduction at level 3

Family: binomial (probit)
Formula: outcome ~ treatment * time + (1 | patientID)

Random effects:

Groups	Name	Estimate	Std.Error
patientID	(Intercept)	2.119	0.1954

Number of obs: 1903, groups: patientID, 289;

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.93105	0.23217	4.0102	6.066e-05
treatmentterbinafine	-0.07618	0.30945	0.2462	8.055e-01
time	-0.19076	0.02060	9.2618	2.010e-20
treatmentterbinafine:time	-0.06420	0.03099	2.0713	3.834e-02

Because of the probit link, this binary generalised linear model is equivalent to a linear transformation model and we can thus use the exact likelihood implemented for the latter model in `mtram()` for parameter estimation (it is still a bit nasty to set-up a constant transformation function $h(y) = \alpha$, I plan to add a more convenient interface later)

```

R> m <- ctm(as.basis(~ outcome, data = toenail),
+         shifting = ~ treatment * time,
+         data = toenail, todistr = "Normal")

```

```
R> toenail_probit <- mlt(m, data = toenail,
+                         fixed = c("outcomemoderate or severe" = 0))
R> toenail_mtram_RI <-
+   mtram(toenail_probit, ~ (1 | patientID),
+         data = toenail, Hessian = TRUE)
R> logLik(toenail_mtram_RI)
```

```
'log Lik.' -633.9638 (df=5)
```

```
R> coef(toenail_mtram_RI)
```

(Intercept)	treatmentterbinafine
0.92946575	0.07699772
time	treatmentterbinafine:time
0.19056617	0.06355426
gamma1	
2.11447751	

For this random intercept model, the exact likelihood is defined as a one-dimensional integral over the unit interval. We use sparse grids (Heiss and Winschel 2008; Ypma 2013) to approximate this integral. The integrand is defined by products of normal probabilities, which are approximated as described by Matić *et al.* (2018). It is important to note that this likelihood can be computed as accurately as necessary whereas Laplace, AGQ, and SR are approximations of limited accuracy.

The results are very close to SR and AGQ, indicating a very good quality of the AGQ and SR approximations. We can also compare the corresponding covariances

```
R> vcov(toenail_glmer_RI_2)
```

```
4 x 4 Matrix of class "dpoMatrix"
```

	(Intercept)	treatmentterbinafine	time
(Intercept)	0.0537125190	-0.046953461	-0.0008877046
treatmentterbinafine	-0.0469534610	0.095609401	0.0013522326
time	-0.0008877046	0.001352233	0.0004239968
treatmentterbinafine:time	0.0013871422	-0.002754670	-0.0003896725
	treatmentterbinafine:time		
(Intercept)		0.0013871422	
treatmentterbinafine		-0.0027546700	
time		-0.0003896725	
treatmentterbinafine:time		0.0009603451	

```
R> solve(toenail_mtram_RI$Hessian)[1:4, 1:4]
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0535092145	-0.046828492	-0.0008863063	0.0013728319
[2,]	-0.0468284922	0.095401203	0.0013454535	-0.0027201601
[3,]	-0.0008863063	0.001345453	0.0004223651	-0.0003889276
[4,]	0.0013728319	-0.002720160	-0.0003889276	0.0009479429

Things get a bit less straightforward when a random slope is added to the model. The two implementations of the Laplace approximation in packages **lme4**

```
R> toenail_glmer_RS <-
+   glmer(outcome ~ treatment * time + (1 + time | patientID),
+         data = toenail, family = binomial(link = "probit"))
R> summary(toenail_glmer_RS)
```

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) [glmerMod]

Family: binomial (probit)

Formula: outcome ~ treatment * time + (1 + time | patientID)

Data: toenail

AIC	BIC	logLik	deviance	df.resid
985.8	1024.7	-485.9	971.8	1896

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.85421	-0.00210	-0.00037	0.00000	2.35828

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
patientID	(Intercept)	118.433	10.883	
	time	3.305	1.818	-0.90

Number of obs: 1903, groups: patientID, 289

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-4.30119	0.26361	-16.316	<2e-16 ***
treatmentterbinafine	0.05419	0.34652	0.156	0.8757
time	-0.06791	0.07846	-0.866	0.3867
treatmentterbinafine:time	-0.23478	0.13885	-1.691	0.0909 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)	trtmnt	time
trtmnttrbnf	-0.662		
time	-0.453	0.342	
trtmnttrbn:	0.270	-0.438	-0.335

```
R> toenail_glmer_RS@theta
```

```
[1] 10.8826888 -1.6359617 0.7930867
```

and **glmmr**

```
R> toenail_glmm_RS_1 <-
+   glmm(outcome ~ treatment * time + (1 + time | patientID),
+       data = toenail, family = binomial(link = "probit"),
+       method = "Laplace")
```

Fitting the model..... done.

```
R> toenail_glmm_RS_1$estim[1:3]
```

```
[1] 4.9992215 -0.5644628 0.4110986
```

```
R> toenail_glmm_RS_1$estim[-(1:3)]
```

```
[1] -3.49232631 0.04197115 -0.06769437 -0.12100940
```

do not quite agree. Note that the standard deviation of the random intercept is twice as large in the `glmer()` output.

The optimisation of the exact discrete likelihood in the transformation framework gives

```
R> toenail_mtram_RS <-
+   mtram(toenail_probit, ~ (1 + time | patientID),
+       data = toenail)
R> logLik(toenail_mtram_RS)
```

```
'log Lik.' -545.1164 (df=7)
```

```
R> coef(toenail_mtram_RS)
```

(Intercept)	treatmentterbinafine
1.5765323	-0.2666840
time	treatmentterbinafine:time
0.5323918	0.1842506
gamma1	gamma2
5.2172376	-0.3723898
gamma3	
0.5285640	

The variance parameters are not too far off the results reported by `glmm()`, but the fixed effects differ quite a bit.

At least in biostatistics, the probit model is less popular than the logit model owing to the better interpretability of the fixed effects as conditional log-odds ratios in the latter. Using a logit link, we can use the transformation approach to compute marginally interpretable time-dependent log-odds ratios from random intercept transformation logit models (`standardise = TRUE()` computes model (M2) instead of the default (M1), see [Hothorn 2019](#)):

```
R> m <- ctm(as.basis(~ outcome, data = toenail),
+          shifting = ~ treatment * time,
+          data = toenail, todistr = "Logistic")
R> toenail_logit <- mlt(m, data = toenail,
+                    fixed = c("outcomemoderate or severe" = 0))
R> toenail_mtram_logit <- mtram(toenail_logit, ~ (1 | patientID),
+                             data = toenail)
R> toenail_mtram_logit_s <- mtram(toenail_logit, ~ (1 | patientID),
+                               data = toenail, standardise = TRUE,
+                               Hessian = TRUE)
```

It is important to note that this model is *not* a logistic mixed-effects model and thus we can't expect to obtain identical results from `glmer()` as it was (partially) the case for the probit model.

From the standardised model, we can compute marginally interpretable probabilities and odds ratios over time

```
R> tmp <- toenail_logit
R> cf <- coef(tmp)
R> cf <- cf[names(cf) != "outcomemoderate or severe"]
R> sdrf <- rev(coef(toenail_mtram_logit_s))[1]
R> cf <- coef(toenail_mtram_logit_s)[names(cf)] / sqrt(sdrf^2 + 1)
R> cf <- c(cf[1], "outcomemoderate or severe" = 0, cf[-1])
R> coef(tmp) <- cf
R> time <- 0:180/10
R> treatment <- sort(unique(toenail$treatment))
R> nd <- expand.grid(time = time, treatment = treatment)
R> nd$prob_logit_s <- predict(tmp, newdata = nd, type = "distribution")[1,]
R> nd$odds <- exp(predict(tmp, newdata = nd, type = "trafo")[1,])
```

We can also sample from the distribution of the maximum likelihood estimators to obtain an idea about the uncertainty (Figure 5).

From the unstandardised logit and probit models, we can also obtain marginally interpretable probabilities as (probit)

```
R> tmp <- toenail_logit
R> cf <- coef(tmp)
R> cf <- cf[names(cf) != "outcomemoderate or severe"]
R> sdrf <- rev(coef(toenail_mtram_logit))[1]
R> cf <- coef(toenail_mtram_logit)[names(cf)]
R> cf <- c(cf[1], "outcomemoderate or severe" = 0, cf[-1])
R> coef(tmp) <- cf
R> pr <- predict(tmp, newdata = nd, type = "distribution")[1,]
R> nd$prob_logit <- pnorm(qnorm(pr) / sdrf)
```

and (logit)

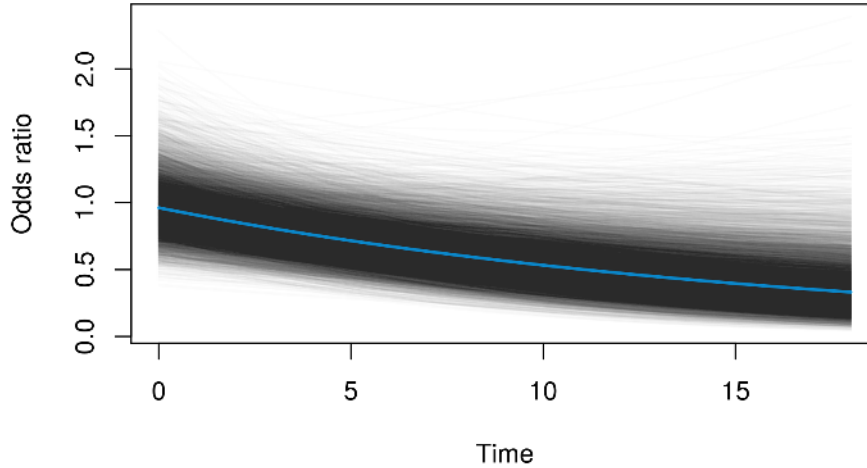


Figure 5: Toe nail data: Marginal odds ratio over time (from a logistic random intercept model). The blue line represents the maximum likelihood estimator, the grey lines are samples from the corresponding distribution.

```
R> tmp <- toenail_probit
R> cf <- coef(tmp)
R> cf <- cf[names(cf) != "outcomemoderate or severe"]
R> sdrf <- rev(coef(toenail_mtram_RI))[1]
R> cf <- coef(toenail_mtram_RI)[names(cf)] / sqrt(sdrf^2 + 1)
R> cf <- c(cf[1], "outcomemoderate or severe" = 0, cf[-1])
R> coef(tmp) <- cf
R> nd$prob_probit <- predict(tmp, newdata = nd, type = "distribution")[1,]
```

The marginal time-dependent probabilities obtained from all three models are very similar as shown in Figure 6.

4. Proportional Odds Models for Bounded Responses

Manuerra and Heller (2010) proposed a mixed-effects model for bounded responses whose fixed effects can be interpreted as log-odds ratios. We fit a transformation model to data from a randomised controlled trial on chronic neck pain treatment (Chow *et al.* 2006). The data are visualised in Figure 7. Subjective neck pain levels were assessed on a visual analog scale, that is, on a bounded interval.

Manuerra and Heller (2010) suggested the conditional model

$$\text{logit}(\mathbb{P}(\text{pain} \leq y \mid \text{treatment}, \text{time}, i)) = h(y) + \beta_{\text{Active}} + \beta_{7 \text{ weeks}} + \beta_{12 \text{ weeks}} + \beta_{7 \text{ weeks, Active}} + \beta_{12 \text{ weeks, Active}} + \alpha_i$$

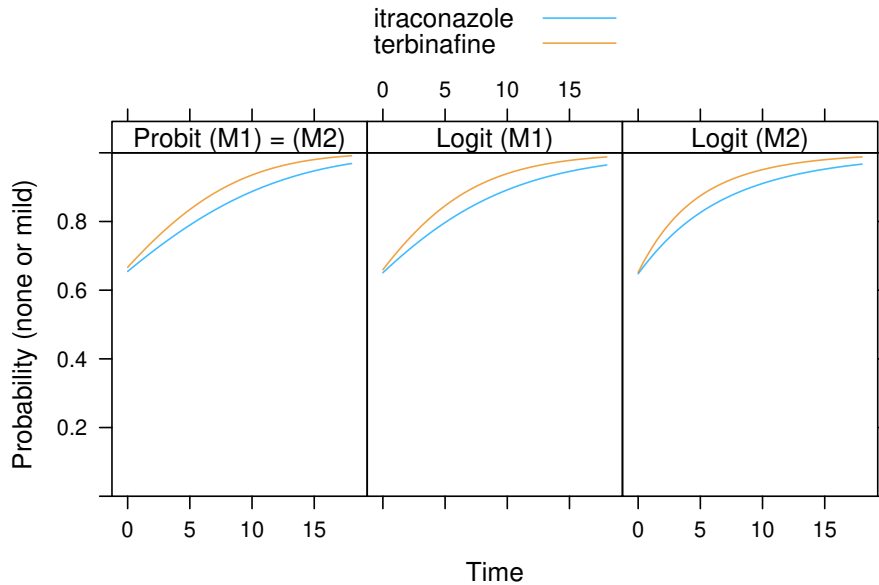


Figure 6: Toe nail data: Comparison of marginal probabilities obtained from a probit linear mixed-effects model and two logistic transformation models (M2: with or M1: without marginal log-odds ratio treatment effect).

with random intercepts $\tilde{\alpha}_i$ such that the odds at baseline, for example, are given by

$$\frac{\mathbb{P}(\text{pain} \leq y \mid \text{Active, baseline}, i)}{\mathbb{P}(\text{pain} > y \mid \text{Active, baseline}, i)} = \exp(\beta_{\text{Active}}) \frac{\mathbb{P}(\text{pain} \leq y \mid \text{Placebo, baseline}, i)}{\mathbb{P}(\text{pain} > y \mid \text{Placebo, baseline}, i)}$$

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

The results

```
R> summary(neck_ocrm)
```

Call:

```
ocrm(formula = vas ~ laser * time + (1 | id), data = pain_df,
      scale = c(0, 1))
```

Random effects:

Name	Variance	Std.Dev.
Intercept id	5.755	2.399

Coefficients:

	Estimate	StdErr	t.value	p.value
laserActive	-2.07922	0.65055	-3.1961	0.001918 **
time7 weeks	-0.60366	0.35744	-1.6889	0.094689 .

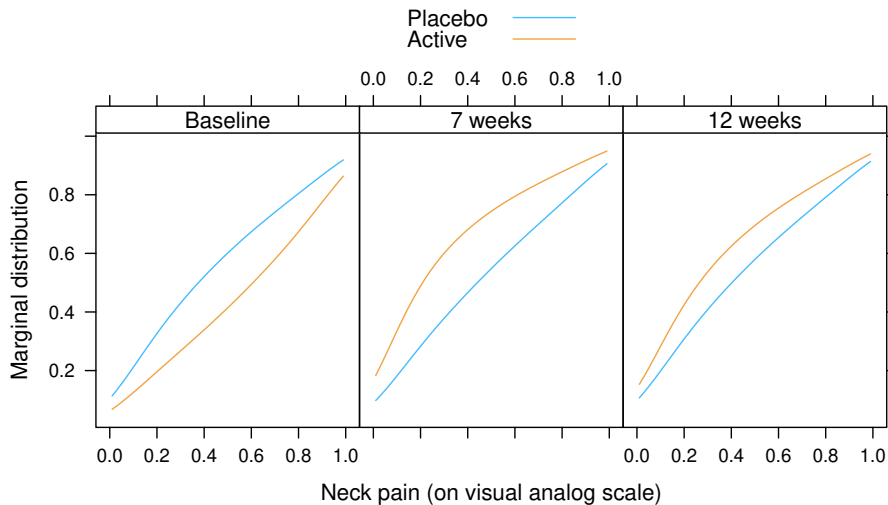


Figure 8: Neck pain: Marginal distribution functions of chronic neck pain evaluated at three different time points under placebo or active low-level laser therapy.

```
+
      extrapolate = TRUE)
R> neck_Colrmer <- mtram(neck_Colr, ~ (1 | id), data = pain_df,
+
      Hessian = TRUE)
R> logLik(neck_Colrmer)
```

```
'log Lik.' 74.35908 (df=13)
```

Based on this model of form (M1), it is possible to derive the marginal distribution functions in the two groups, see Figure 8.

5. Marginally Interpretable Weibull and Cox Models

The CAO/ARO/AIO-04 randomised clinical trial (Rödel *et al.* 2015) compared Oxaliplatin added to fluorouracil-based preoperative chemoradiotherapy and postoperative chemotherapy to the same therapy using fluorouracil only for rectal cancer patients. Patients were randomised in the two treatment arms by block randomisation taking the study center, the lymph node involvement (negative, positive), and tumour grading (T1-3 vs. T4) into account. The primary endpoint was disease-free survival, defined as the time between randomisation and non-radical surgery of the primary tumour (R2 resection), locoregional recurrence after R0/1 resection, metastatic disease or progression, or death from any cause, whichever occurred first. The observed outcomes are a mix of exact dates (time to death or incomplete removal of the primary tumour), right-censoring (end of follow-up or drop-out), and interval-censoring (local or distant metastases). We are interested in a clustered Cox or Weibull model for interval-censored survival times. The survivor functions, estimated separately for each of the four strata defined by lymph node involvement and tumour grading, are given in Figure 9. The implementation of mixed transformation models is currently not able to deal with mixed exact and censored outcomes in the same cluster. We therefore recode exact event times as

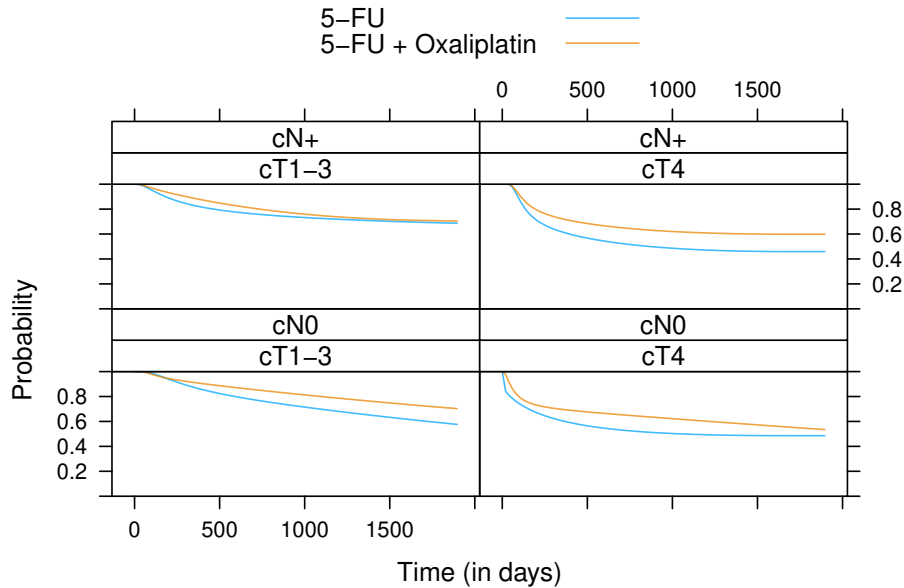


Figure 9: Rectal cancer: Distribution of disease-free survival times for two treatments in the four strata defined by lymph node involvement (negative or positive) and tumour grading (T1-3 or T4).

being interval-censored by adding a 4-day window to each exact event time (variable `iDFS2`).

```
R> ### convert "exact" event dates to interval-censoring (+/- one day)
R> tmp <- CA0surv$iDFS
R> exact <- tmp[,3] == 1
R> tmp[exact,2] <- tmp[exact,1] + 2
R> tmp[exact,1] <- pmax(tmp[exact,1] - 2, 0)
R> tmp[exact,3] <- 3
R> CA0surv$iDFS2 <- tmp
```

We start with the random intercept model

$$\mathbb{P}(Y > y \mid \text{treatment}) = \exp \left(- \exp \left(\frac{\vartheta_1 + \vartheta_2 \log(y) - \beta_{5\text{-FU} + \text{Ox}}}{\sqrt{\gamma_1^2 + 1}} \right) \right)$$

assuming a marginal Weibull model whose effects are scaled depending on the variance γ_1^2 of a block-specific (interaction of lymph node involvement, tumour grading, and study center) random intercept:

```
R> CAO_SR <- Survreg(iDFS2 ~ randarm, data = CA0surv)
R> CAO_SR_mtram_s <- mtram(CAO_SR, ~ (1 | Block), data = CA0surv,
+                          standardise = TRUE, Hessian = TRUE)
R> logLik(CAO_SR_mtram_s)
```

```
'log Lik.' -2081.542 (df=4)
```

```
R> (cf <- coef(CAO_SR_mtram_s))
```

```

              (Intercept)                log(iDFS2)
              -6.2990054                 0.7412855
randarm5-FU + Oxaliplatin                gamma1
              0.2328600                 0.1683613
```

```
R> (OR <- exp(-cf["randarm5-FU + Oxaliplatin"] / sqrt(cf["gamma1"]^2 + 1)))
```

```

randarm5-FU + Oxaliplatin
              0.794829
```

We are, of course, interested in the marginal treatment effect, that is, the odds ratio

$$\exp(-\beta_{5\text{-FU} + \text{Ox}} / \sqrt{\gamma_1^2 + 1}).$$

We simply sample from the joint normal distribution of the maximum likelihood estimators $\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\beta}_{5\text{-FU} + \text{Ox}}, \hat{\gamma}_1$ and compute confidence intervals for the marginal treatment effect 0.79 as

```
R> S <- solve(CAO_SR_mtram_s$Hessian)
R> sqrt(diag(S))
```

```
[1] 0.29019989 0.03872268 0.10722445 0.12433065
```

```
R> rbeta <- rmvnorm(10000, mean = coef(CAO_SR_mtram_s),
+                  sigma = S)
R> s <- rbeta[,ncol(rbeta)]
R> rbeta <- rbeta[, -ncol(rbeta)] / sqrt(s^2 + 1)
R> quantile(exp(-rbeta[, ncol(rbeta)]), prob = c(.025, .5, .975))
```

```

          2.5%          50%          97.5%
0.6478684 0.7953985 0.9747696
```

In a next step, we stratify with respect to lymph node involvement and tumour grading: For each of the four strata, the parameters ϑ_1 and ϑ_2 are estimated separately:

```
R> CAO_SR_2 <- Survreg(iDFS2 | 0 + strat_n:strat_t ~ randarm, data = CAOsurv)
R> CAO_SR_2_mtram_s <- mtram(CAO_SR_2, ~ (1 | Block), data = CAOsurv,
+                            standardise = TRUE, Hessian = TRUE)
R> logLik(CAO_SR_2_mtram_s)
```

```
'log Lik.' -2067.797 (df=10)
```

```
R> (cf <- coef(CAO_SR_2_mtram_s))
```

```

(Intercept):strat_ncN0:strat_tcT1-3  log(iDFS2):strat_ncN0:strat_tcT1-3
                                     -7.8833653                               0.9584499
(Intercept):strat_ncN+:strat_tcT1-3  log(iDFS2):strat_ncN+:strat_tcT1-3
                                     -6.2225174                               0.7198965
  (Intercept):strat_ncN0:strat_tcT4   log(iDFS2):strat_ncN0:strat_tcT4
                                     -3.0467542                               0.3711277
  (Intercept):strat_ncN+:strat_tcT4   log(iDFS2):strat_ncN+:strat_tcT4
                                     -4.8207089                               0.6214653
    randarm5-FU + Oxaliplatin          gamma1
                                     0.2240023                               0.1474685

```

```
R> (OR_2 <- exp(-cf["randarm5-FU + Oxaliplatin"] / sqrt(cf["gamma1"]^2 + 1)))
```

```

randarm5-FU + Oxaliplatin
      0.8012313

```

The corresponding confidence interval for the marginal treatment effect is then

```

[1] 0.68882415 0.09402924 0.34618756 0.04633063 1.01843933 0.13874107
[7] 0.68657931 0.09534206 0.10731700 0.13455556

```

```

      2.5%      50%      97.5%
0.6528915 0.8044045 0.9856882

```

We now relax the Weibull assumption in the Cox model

$$\mathbb{P}(Y > y \mid \text{treatment}, i) = \exp \left(- \exp \left(\frac{\mathbf{a}(\log(y))^{\top} \boldsymbol{\vartheta} + \beta_{5\text{-FU} + \text{Ox}}}{\sqrt{\gamma_1^2 + 1}} \right) \right)$$

(note the positive sign of the treatment effect).

```

R> CAO_Cox_2 <- Coxph(iDFS2 | 0 + strat_n:strat_t ~ randarm, data = CAOSurv,
+                   support = c(1, 1700), log_first = TRUE, order = 4)
R> logLik(CAO_Cox_2)

```

```
'log Lik.' -2021.875 (df=21)
```

```

R> CAO_Cox_2_mtram_s <- mtram(CAO_Cox_2, ~ (1 | Block), data = CAOSurv,
+                             standardise = TRUE, Hessian = TRUE)
R> logLik(CAO_Cox_2_mtram_s)

```

```
'log Lik.' -2031.051 (df=22)
```

```
R> coef(CAO_Cox_2_mtram_s)
```

```

Bs1(iDFS2):strat_ncN0:strat_tcT1-3 Bs2(iDFS2):strat_ncN0:strat_tcT1-3
      -5.832796e+01                      -3.163149e+00
Bs3(iDFS2):strat_ncN0:strat_tcT1-3 Bs4(iDFS2):strat_ncN0:strat_tcT1-3
      -3.161654e+00                      -2.181838e+00
Bs5(iDFS2):strat_ncN0:strat_tcT1-3 Bs1(iDFS2):strat_ncN+:strat_tcT1-3
      -7.656357e-01                      -1.653595e+01
Bs2(iDFS2):strat_ncN+:strat_tcT1-3 Bs3(iDFS2):strat_ncN+:strat_tcT1-3
      -8.143542e+00                      -2.071762e+00
Bs4(iDFS2):strat_ncN+:strat_tcT1-3 Bs5(iDFS2):strat_ncN+:strat_tcT1-3
      -1.792342e+00                      -7.614334e-01
Bs1(iDFS2):strat_ncN0:strat_tcT4   Bs2(iDFS2):strat_ncN0:strat_tcT4
      -2.525002e+00                      -2.519727e+00
Bs3(iDFS2):strat_ncN0:strat_tcT4   Bs4(iDFS2):strat_ncN0:strat_tcT4
      -2.330486e+00                      -3.587793e-01
Bs5(iDFS2):strat_ncN0:strat_tcT4   Bs1(iDFS2):strat_ncN+:strat_tcT4
      -1.456236e-01                      -4.096546e+01
Bs2(iDFS2):strat_ncN+:strat_tcT4   Bs3(iDFS2):strat_ncN+:strat_tcT4
      -2.000965e+00                      -1.987891e+00
Bs4(iDFS2):strat_ncN+:strat_tcT4   Bs5(iDFS2):strat_ncN+:strat_tcT4
      -3.570476e-01                      1.919546e-14
      randarm5-FU + Oxaliplatin                gamma1
      -1.806504e-01                      -1.215448e-06

```

with confidence interval

```

      2.5%      50%      97.5%
0.6911201 0.8325653 1.0084011

```

Because the estimated variance parameter γ_1 is not very large, we would expect to see similar results in a conditional Cox model with normal frailty term

```

R> library("coxme")
R> m <- coxme(DFS ~ randarm + (1 | Block), data = CA0surv)
R> summary(m)

```

Cox mixed-effects model fit by maximum likelihood

```

Data: CA0surv
events, n = 357, 1236
Iterations= 22 91
              NULL Integrated      Fitted
Log-likelihood -2432.971  -2430.475 -2414.482

              Chisq   df      p  AIC   BIC
Integrated loglik  4.99  2.00 0.0823850 0.99  -6.76
Penalized loglik 36.98 16.72 0.0029833 3.53 -61.33

```

Model: DFS ~ randarm + (1 | Block)

Fixed coefficients

	coef	exp(coef)	se(coef)	z	p
randarm5-FU + Oxaliplatin	-0.2310483	0.7937011	0.1067215	-2.16	0.03

Random effects

Group Variable	Std Dev	Variance
Block Intercept	0.21741797	0.04727057

```
R> sd <- sqrt(diag(vcov(m)))
R> exp(coef(m) + c(-1, 0, 1) * qnorm(.975) * sd)
```

```
[1] 0.6438957 0.7937011 0.9783596
```

6. Assessment of Unexplained Variability

Pollet and Nettle (2009) reported on an association between partner wealth and female self-reported orgasm frequency. It was later (Herberich *et al.* 2010) pointed out that the finding was due to an incorrectly implemented variable selection procedure based on a proportional odds (cumulative logit) model for the ordinal variable corresponding to the question “When having sex with your current partner, how often did you have orgasm?” with possible answer categories $y_1 = \text{Always}$, $y_2 = \text{Often}$, $y_3 = \text{Sometimes}$, $y_4 = \text{Rarely}$, or $y_5 = \text{Never}$. The full model explains the conditional distribution of orgasm frequency by $\mathbf{x} =$ partner income, partner height, the duration of the relationship, the respondents age, the difference between both partners regarding education and wealth, the respondents education, health, happiness, and place of living (regions in China) of the form

$$\mathbb{P}(\text{orgasm} \leq y_k | \mathbf{x}) = \text{expit}(\vartheta_k + \mathbf{x}^\top \boldsymbol{\beta})$$

for $i = 1, \dots, N = 1531$ independent heterosexual couples. In this model, the regression coefficients $\boldsymbol{\beta}$ can be interpreted as log-odds ratios and we question the appropriateness of this model here by including a subject-specific random intercept with standard deviation γ_1 . This changes the model to

$$\mathbb{P}(\text{orgasm} \leq y_k | \mathbf{x}) = \Phi \left(\frac{\Phi^{-1}(\text{expit}(\vartheta_k + \mathbf{x}^\top \boldsymbol{\beta}))}{\sqrt{\gamma_1^2 + 1}} \right)$$

A value of γ_1 close to zero corresponds to marginal distributions very similar to the proportional odds model and, consequently, it is appropriate to interpret $\boldsymbol{\beta}$ as log-odds ratios. Larger values of γ_1 indicate a more variable distribution and thus the choice $F = \text{expit}$ might be questionable.

We obtain

```
R> CHFLS_Polr <- Polr(orgasm ~ AincomeSD + AheightSD + RAdurationSD +
+ RageSD + edudiffSD + wealthdiffSD + Redu +
+ Rhealth + Rhappy + Region, data = orgAcc)
R> logLik(CHFLS_Polr)
```



```
'log Lik.' -1852.615 (df=27)
```

```
R> orgAcc$ID <- factor(1:nrow(orgAcc))
R> CHFLS_mtram <- mtram(CHFLS_PoI_r, ~ (1 | ID),
+                       data = orgAcc)
R> logLik(CHFLS_mtram)
```

```
'log Lik.' -1852.829 (df=28)
```

```
R> coef(CHFLS_mtram)
```

orgasm1	orgasm2	orgasm3	orgasm4
-3.00585271	-1.38078485	1.20377067	3.14108781
AincomeSD	AheightSD	RAdurationSD	RageSD
0.02603981	-0.02217060	0.07529263	-0.35310380
edudiffSD	wealthdiffSD	Redujcol	Reduupmid
-0.17413324	-0.03428024	0.13682888	0.17668815
Redulowmid	Reduprimary	Redunoschool	Rhealthnot good
-0.42694677	-0.95912533	-1.80236615	1.36152834
Rhealthfair	Rhealthgood	Rhealthexcellent	Rhappynot too
1.69707889	1.83806835	1.86386980	0.29162606
Rhappyrelatively	Rhappyvery	RegionNortheast	RegionNorth
0.75856526	1.02374623	0.40651849	0.20369845
RegionInlandS	RegionCoastalE	RegionCoastalS	gamma1
0.49050565	0.20059228	0.58338272	0.01678467

and from $\hat{\gamma}_1 = 0.017$ can conclude that the proportional odds model is appropriate here.

References

- Backer MD, Vroey CD, Lesaffre E, Scheys I, Keyser PD (1998). “Twelve Weeks of Continuous Oral Therapy for Toenail Onychomycosis Caused by Dermatophytes: A Double-blind Comparative Trial of Terbinafine 250 mg/day versus Itraconazole 200 mg/day.” *Journal of the American Academy of Dermatology*, **38**(5(2)), S57–S63. doi:10.1016/S0190-9622(98)70486-4.
- Bates D, Mächler M, Bolker B, Walker S (2015). “Fitting Linear Mixed-Effects Models Using lme4.” *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.
- Belenky G, Wesensten NJ, Thorne DR, Thomas ML, Sing HC, Redmond DP, Russo MB, Balkin TJ (2003). “Patterns of Performance Degradation and Restoration During Sleep Restriction and Subsequent Recovery: A Sleep Dose-response Study.” *Journal of Sleep Research*, **12**, 1–12. doi:10.1046/j.1365-2869.2003.00337.x.
- Box GEP, Cox DR (1964). “An Analysis of Transformations.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **26**(2), 211–252.

- Chow RT, Heller GZ, Barnsley L (2006). “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. doi:10.1016/j.pain.2006.05.018.
- Heiss F, Winschel V (2008). “Likelihood Approximation by Numerical Integration on Sparse Grids.” *Journal of Econometrics*, **144**(1), 62–80. doi:10.1016/j.jeconom.2007.12.004.
- Herberich E, Hothorn T, Nettle D, Pollet TV (2010). “A Re-evaluation of the Statistical Model in Pollet and Nettle 2009.” *Evolution and Human Behavior*, **31**(2), 150–151.
- Hothorn T (2019). “Marginally Interpretable Linear Transformation Models for Clustered Observations.” *Technical report*, arXiv 1910.09219. URL <http://arxiv.org/abs/1910.09219>.
- Hothorn T, Barbanti L (2020). *tram: Transformation Models*. R package version 0.5-2, URL <http://ctm.R-forge.R-project.org>.
- Manuguerra M, Heller GZ (2010). “Ordinal Regression Models for Continuous Scales.” *International Journal of Biostatistics*, **6**(1), 14. doi:10.2202/1557-4679.1230.
- Matić I, Radoičić R, Stefanica D (2018). “A Sharp Pólya-Based Approximation to the Normal CDF.” *Applied Mathematics and Computation*, **322**, 111–122. URL <https://ssrn.com/abstract=2842681>.
- Ogden HE (2015). “A Sequential Reduction Method for Inference in Generalized Linear Mixed Models.” *Electronic Journal of Statistics*, **9**, 135–152. doi:10.1214/15-EJS991.
- Pollet TV, Nettle D (2009). “Partner Wealth Predicts Self-Reported Orgasm Frequency in a Sample of Chinese Women.” *Evolution and Human Behavior*, **30**, 146–151.
- Rödel C, Graeven U, Fietkau R, Hohenberger W, Hothorn T, Arnold D, Hofheinz RD, Ghadimi M, Wolff HA, Lang-Welzenbach M, Raab HR, Wittekind C, Ströbel P, Staib L, Wilhelm M, Grabenbauer GG, Hoffmanns H, Lindemann F, Schlenska-Lange A, Folprecht G, Sauer R, Torsten Liersch on behalf of the German Rectal Cancer Study Group (2015). “Oxaliplatin Added to Fluorouracil-based Preoperative Chemoradiotherapy and Postoperative Chemotherapy of Locally Advanced Rectal Cancer (the German CAO/ARO/AIO-04 study): Final Results of the Multicentre, Open-label, Randomised, Phase 3 Trial.” *The Lancet Oncology*, **16**(8), 979–989. doi:10.1016/S1470-2045(15)00159-X.
- Ypma J (2013). *SparseGrid: Sparse grid integration in R*. R package version 0.8.2, URL <https://CRAN.R-project.org/package=SparseGrid>.

Affiliation:

Torsten Hothorn
Institut für Epidemiologie, Biostatistik und Prävention
Universität Zürich
Hirschengraben 84, CH-8001 Zürich, Switzerland
Torsten.Hothorn@uzh.ch
<http://tiny.uzh.ch/bh>