24º SINAPE | Simpósio Nacional de Probabilidade e Estatística

MODELING LONGITUDINAL DATA USING ROBUST MIXED MODELS IN R

Fenanda L. Schumacher - The Ohio State University, USA
Larissa A. Matos - Universidade Estadual de Campinas, Brasil
Victor H. Lachos - University of Connecticut, USA

July, 31 2022

Overview

Motivating datasets

The general linear mixed model (LMM)

The skew-normal linear mixed model

LMM based on scale mixture of skew-normal distributions

Accounting for within-subject serial correlation

Fitting LMM in R

The R package **skewlmm**

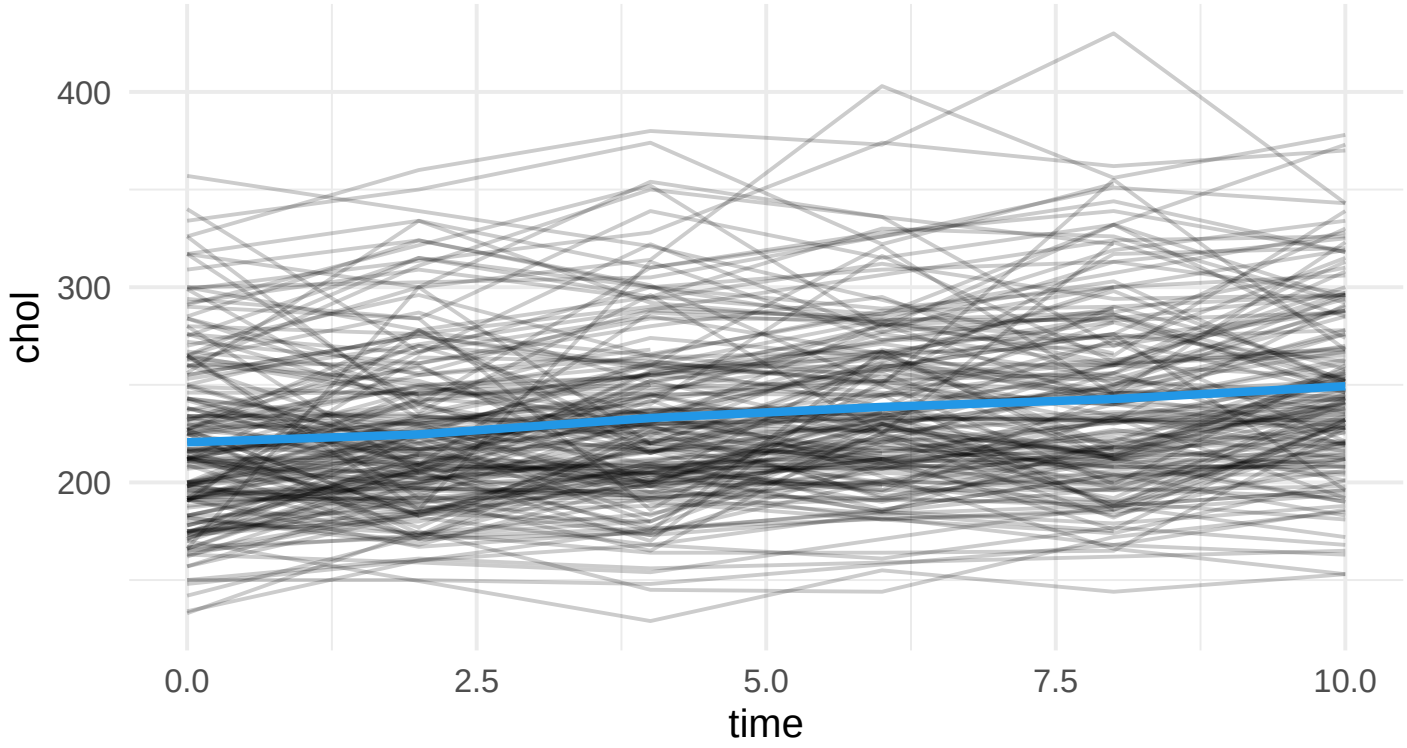
Practical activity

LMM with censored response

Motivating datasets

Motivating Data: Cholesterol levels

- Cholesterol levels collected as part of the famed Framingham heart study.
- The data set includes the cholesterol levels over time, age at baseline and gender for $n = 200$ randomly selected individuals.



Motivating Data: Schizophrenia data

- Schizophrenia is a severe psychiatric disorder characterized by delusions, hallucinations, persistent delusions and sometimes disorganized behavior and speech.

- Lapierre et al. (1990) presented a double-blinded clinical trial with randomization among four treatments: three doses (low, medium and high) of a new therapy (NT) against a standard therapy (ST), for 245 patients with acute schizophrenia.

- The study was conducted at 13 clinical centres, and the primary response variable was assessed using the Brief Psychiatric Rating Scale (BPRS) at baseline (week 0), and at weeks 1, 2, 3, 4 and 6 of treatment.

Motivating Data: Schizophrenia data

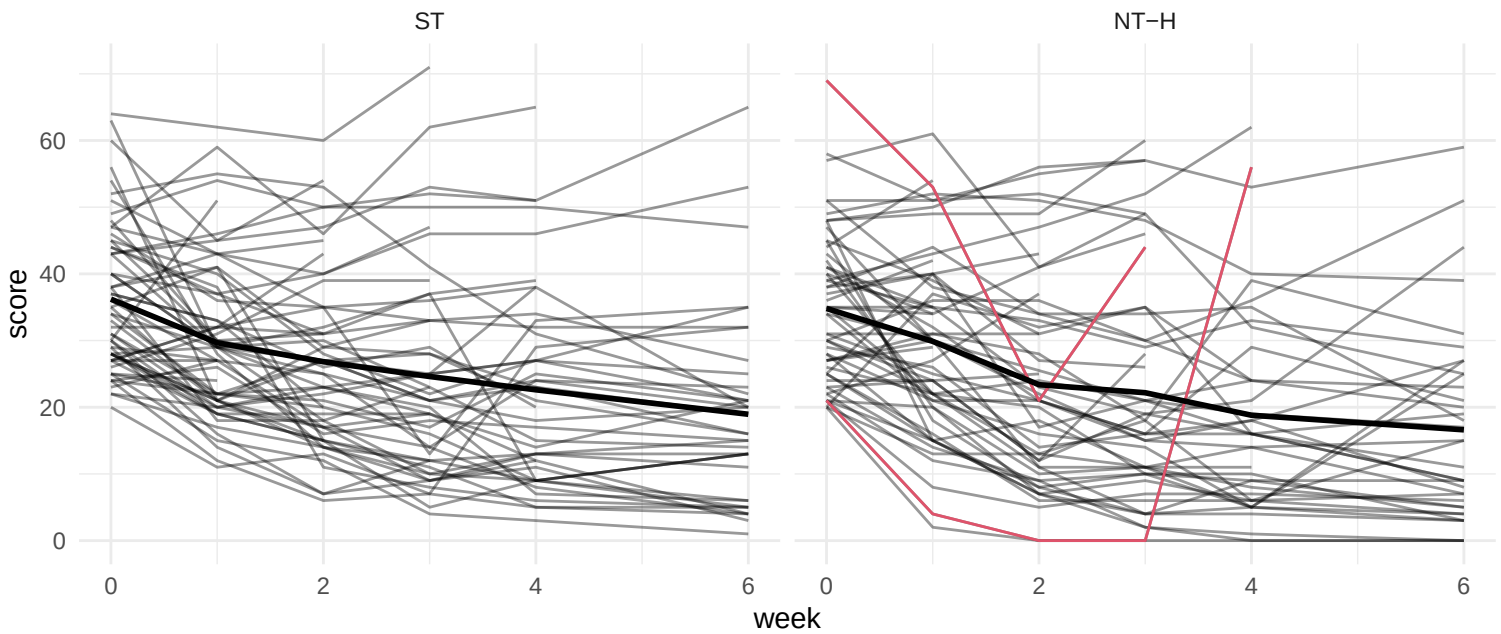


Figure 1: Trajectories of schizophrenia levels for the data.

The general linear mixed model (LMM)

Introduction

- Linear mixed-effects models (LMMs) are an important class of statistical models that can be used to analyze correlated data (repeated measures data, clustered data and longitudinal studies). Such data are encountered in a variety of fields including biostatistics, public health, psychometrics, educational measurement, sociology and geophysics.
- LMMs allow to incorporate parameters associated with an entire population (fixed effects) along with effects associated with individual experimental units drawn at random from a population (random effects).
- The increasing popularity of these models is explained by the flexibility they offer in modeling the within-subject correlation often present in longitudinal data, by the handling of both balanced and unbalanced data, and by the availability of reliable and efficient software for fitting them.

Normal LMM

In general, a normal linear mixed-effects model is defined as

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, n\,, \tag{1}$$

where

- $\mathbf{X}_i$ of dimension $n_i \times l$ is the design matrix corresponding to the fixed effects,
- $\boldsymbol{\beta}$ of dimension $l \times 1$ is a vector of population-averaged regression coefficients called fixed effects,
- $\mathbf{Z}_i$ of dimension $n_i \times q$ is the design matrix corresponding to the $q \times 1$ random effects vector $\mathbf{b}_i$, and
- $\boldsymbol{\epsilon}_i$ of dimension $n_i \times 1$ is the vector of random errors.

Normal LMM

- Usual assumptions:
  1. $\mathbf{b}_i \overset{\text{iid}}{\sim} N_q(\mathbf{0}, \mathbf{D}) \perp \boldsymbol{\epsilon}_i \overset{\text{ind}}{\sim} N_{n_i}(\mathbf{0}, \boldsymbol{\Sigma}_i)$.
  2. $\boldsymbol{\Sigma}_i = \sigma_e^2 \mathbf{I}_{n_i}$.

- Assumption 1. implies that $\mathbf{Y}_i \overset{\text{ind}}{\sim} N_{n_i}(\mathbf{X}_i\boldsymbol{\beta}, \mathbf{Z}_i\mathbf{D}\mathbf{Z}_i^\top + \boldsymbol{\Sigma}_i)$.

- Assumption 2. (together with assumption 1.) implies conditional independence (UNC).

Cholesterol levels: Fitting the same linear mixed model used by Zhang and Davidian (2001), which is given by

$$y_{ij} = \beta_0 + \beta_1 \text{sex}_j + \beta_2 \text{age}_j + \beta_3 t_{ij} + b_{0j} + b_{1j} t_{ij} + \epsilon_{ij},$$

where y_{ij} is cholesterol level divided by 100 at the i*th* time for subject j and t_{ij} is (time - 5)/10, with time measured in years from baseline.

Application: Cholesterol data

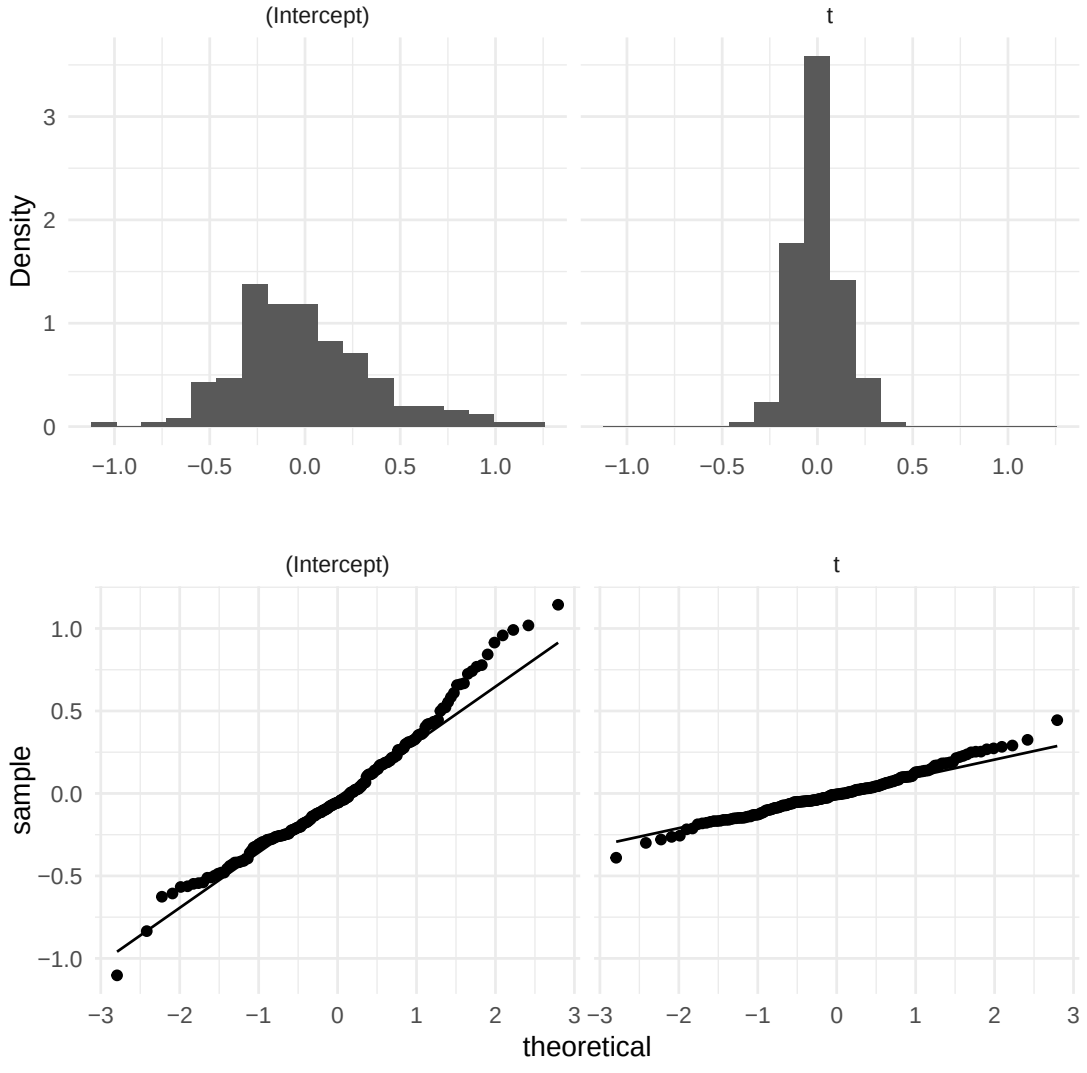


Figure 2: Results from empirical Bayes estimates of random effects (N-LMM).

Introduction

- The misspecification of the distributional assumption may result in invalid statistical inferences, especially when dealing with heavy tails and skewness.
- For instance, substantial bias in the ML estimates of regression parameters can result when the random-effects distribution is misspecified (Drikvandi et al., 2017).
- Some proposals have been made in the literature for relaxing the assumption of normality, for instance:
  - Pinheiro et al. (2001) proposed a multivariate t-LMM and showed that it performed well in the presence of outliers;
  - Davidian and Zhang (2001) proposed the semi-nonparametric LMM.
  - Celeux and Lavergne (2005) proposed the finite mixture of linear mixed models.
  - Arellano-Valle et al. (2005) proposed a skew-normal (SN) LMM.

The skew-normal linear mixed model

The skew-normal distribution

- The $\mathrm{SN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda})$ distribution (Azzalini and Dalla Valle, 1996) is defined as:

$$f(\mathsf{y}) = 2\phi_p(\mathsf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\Phi(\boldsymbol{\lambda}^\top \boldsymbol{\Sigma}^{-1/2}(\mathsf{y} - \boldsymbol{\mu})), \quad \mathsf{y} \in \mathbb{R}^p.$$

- If $\mathsf{W} \sim \mathrm{SN}_p(\mathbf{0}, \mathsf{I}_p, \boldsymbol{\lambda})$, then

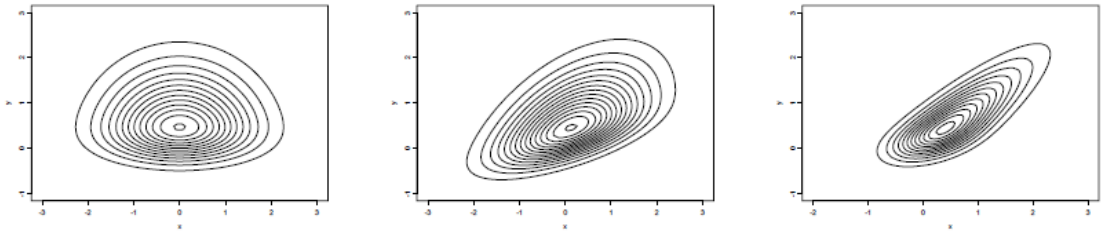
$$\mathsf{W} \stackrel{d}{=} \boldsymbol{\delta}|T_0| + (\mathsf{I}_p - \boldsymbol{\delta}\boldsymbol{\delta}^\top)^{1/2}\mathsf{T}_1, \quad \text{with} \quad \boldsymbol{\delta} = \frac{\boldsymbol{\lambda}}{\sqrt{1 + \boldsymbol{\lambda}^\top \boldsymbol{\lambda}}},$$

  where $T_0 \sim \mathrm{N}_1(0, 1) \perp \mathsf{T}_1 \sim \mathrm{N}_p(\mathbf{0}, \mathsf{I}_p)$.

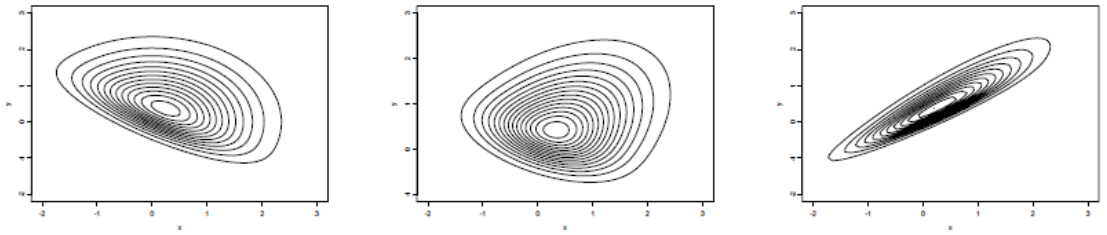
- If $\mathsf{W} \sim \mathrm{SN}_p(\mathbf{0}, \mathsf{I}_p, \boldsymbol{\lambda})$, then $\mathsf{Y} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\mathsf{W} \sim \mathrm{SN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda})$ and $\mathrm{E}\{\mathsf{Y}\} = \boldsymbol{\mu} + \sqrt{\frac{2}{\pi}}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\delta}$, $\mathrm{Var}\{\mathsf{Y}\} = \boldsymbol{\Sigma} - \frac{2}{\pi}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\delta}\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{1/2}$.

Contours of the bi-variate SN distribution

(a) For $\boldsymbol{\lambda} = (0, 3)^T$ and $\rho = 0,\ 0.5,\ 0.9$, respectively.



(b) For $\boldsymbol{\lambda} = (2, 3)^T$ and $\rho = 0,\ 0.5,\ 0.9$, respectively.



(c) For $\boldsymbol{\lambda} = (-2, 2)^T$ and $\rho = 0,\ 0.5,\ 0.9$, respectively.

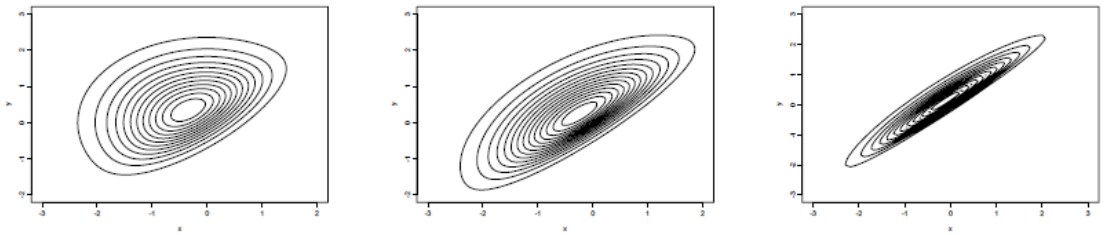


Figure 1: Contour of the bivariate skew-normal distribution in (2.2), with $\boldsymbol{\mu} = (0, 0)^T, \boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^T$ for different values of λ_1, λ_2 and ρ.

The skew-normal linear mixed model (SN-LMM)

Arellano-Valle, Bolfarine and Lachos (2005) [Journal of Data Science], define the SN-LMM as:

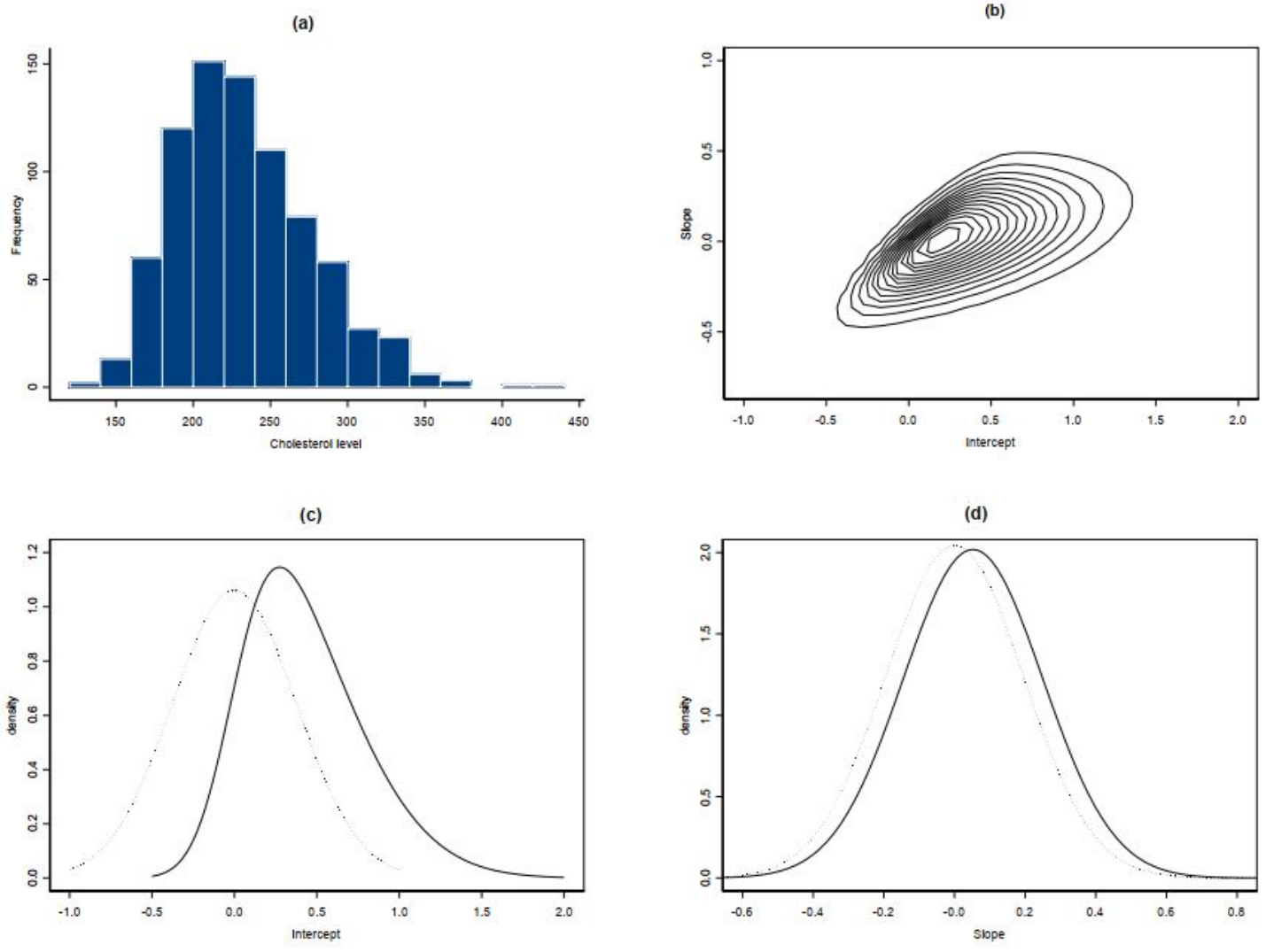
$$\begin{pmatrix} \mathbf{b}_i \\ \boldsymbol{\epsilon}_i \end{pmatrix} \overset{\text{ind}}{\sim} \text{SN}_{q+n_i}\left(\begin{pmatrix} 0 \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_i \end{pmatrix}, \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{0} \end{pmatrix}\right), i = 1, \ldots, n.$$

- An EM-type algorithm is proposed for ML estimation.
- Interesting properties are developed. For instance, the marginal distribution of the response is still skew-normal family.
- So far, this paper has around 250 citations.

Application: Cholesterol data

	N-LMM		SN-LMM	
Parameter	Estimate	SE	Estimate	SE
β_0 (intercept)	1.5969	0.1580	1.7327	0.1530
β_1 (sex)	-0.0630	0.0539	-0.0481	0.0515
β_2 (age)	0.0184	0.0034	0.0151	0.0033
β_3 (year)	0.2817	0.0255	0.2809	0.0266
σ^2	0.0434	0.0017	0.0429	0.0017
d_{11}	0.3716	0.0200	0.5290	0.0479
d_{12}	0.0562	0.0158	0.0022	0.0305
d_{22}	0.1874	0.0281	0.2171	0.0324
λ_{b1}	-	-	7.9313	
λ_{b2}	-	-	-3.7514	
log-like	-160.987		-152.13	
AIC	337.973		324.26	
BIC	377.58		373.768	

Application: Cholesterol data



Some publications related to the SN-LMM

- In a subsequent paper, Arellano-Valle, Bolfarine and Lachos (2007), developed a Bayesian approach for the SN-LMM.
- Lachos, Bolfarine, Arellano-Valle and Montenegro (2007). Likelihood-based inference for multivariate skew-normal regression models (Communication in Statistics: Theory and Methods).
- Mattos, Matos and Lachos (2022). Likelihood-based inference for mixed-effects models with censored response using skew-normal distribution. (Springer: Innovations in multivariate statistical modeling: navigating theoretical and multidisciplinary domains).

Further complications in LMM

- Another complication arise when the data presents skewness and heavy tails behavior, simultaneously.
- Lachos et al. (2010) proposed a parametric robust modeling of LMM based on skew-normal/independent (SNI) distributions. This class of asymmetric distributions is attractive as it simultaneously models the skewness with heavy tails.
- In longitudinal studies, repeated measures are collected over time and hence the error term can be serially correlated.

LMM based on scale mixture of skew-normal distributions

Scale mixture of skew-normal distributions

A random vector Y has a SMSN distribution with location parameter $\boldsymbol{\mu}$, scale parameter $\boldsymbol{\Sigma}$ and skewness parameter $\boldsymbol{\lambda}$, denoted by $\mathrm{SMSN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda; H)$, if it has the following stochastic representation:

$$\mathsf{Y} = \boldsymbol{\mu} + \kappa^{1/2}(U)\mathsf{Z}, \quad U \perp \mathsf{Z},$$

where $\mathsf{Z} \sim SN_p(0, \boldsymbol{\Sigma}, \boldsymbol{\lambda})$, U is a positive random variable with cdf $H(\cdot\,|\boldsymbol{\nu})$.

From the stochastic representation, it is straightforward that $\mathsf{Y}|U = u \sim \mathrm{SN}_p(\boldsymbol{\mu}, \kappa(u)\boldsymbol{\Sigma}, \boldsymbol{\lambda})$.

SMSN - special cases

- When $\boldsymbol{\lambda} = 0$, we get the the symmetric class $\mathrm{SMN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}; H)$;

- When $\kappa(u) = u^{-1}$, we get the skew-normal/independent (SNI) class of distributions:
  - By taking $U \sim \mathrm{Gamma}(\nu/2, \nu/2)$, the $\mathrm{ST}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu)$ can be derived;
  - By taking $U \sim \mathrm{Beta}(\nu, 1)$, the $\mathrm{SSL}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu)$ can be derived;
  - By taking U as a discrete random variable with probability function given by $h(u|\boldsymbol{\nu}) = \nu_1 \mathbb{I}_{\{\nu_2\}}(u) + (1 - \nu_1)\mathbb{I}_{\{1\}}(u)$, the $\mathrm{SCN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu, \rho)$ can be derived, where $\nu_1, \nu_2 \in (0, 1)$.

SMSN-LMM

The SMSN-LMM proposed by Lachos, Ghosh and Arellano-Valle (2010) [Statistica Sinica] can be defined by considering

$$\begin{pmatrix} \mathsf{b}_i \\ \boldsymbol{\epsilon}_i \end{pmatrix} \overset{\text{ind}}{\sim} \text{SMSN}_{q+n_i}\left(\begin{pmatrix} \mathsf{0} \\ \mathsf{0} \end{pmatrix}, \begin{pmatrix} \mathsf{D} & \mathsf{0} \\ \mathsf{0} & \boldsymbol{\Sigma}_i \end{pmatrix}, \begin{pmatrix} \boldsymbol{\lambda} \\ \mathsf{0} \end{pmatrix}; H\right), \quad (2)$$

$i = 1, \ldots, n$, where

- $\mathsf{D} = \mathsf{D}(\boldsymbol{\alpha})$ depends on unknown and reduced parameter vector $\boldsymbol{\alpha}$, and
- $\boldsymbol{\Sigma}_i = \sigma_e^2 \mathsf{R}_i$, with R_i being a known matrix of dimension $n_i \times n_i$.
- When $\mathsf{R}_i = \mathsf{I}_{n_i}$, we have the UNC structure.

Important remarks

1. $\mathbf{b}_i \overset{\text{iid}}{\sim} \text{SMSN}_q(\mathbf{0}, \mathbf{D}, \boldsymbol{\lambda}; H)$ and $\boldsymbol{\epsilon}_i \overset{\text{ind}}{\sim} \text{SMN}_{n_i}(\mathbf{0}, \sigma_e^2 \mathbf{R}_i; H)$.

2. $\mathbf{b}_i$ and $\boldsymbol{\epsilon}_i$ are not independent in general, but $\mathbf{b}_i | U_i \perp \boldsymbol{\epsilon}_i | U_i$, and therefore $\mathbf{b}_i$ and $\boldsymbol{\epsilon}_i$ are uncorrelated.

3. Marginally,

$$\mathbf{Y}_i \overset{\text{ind}}{\sim} \text{SMSN}_{n_i}(\mathbf{X}_i\boldsymbol{\beta}, \boldsymbol{\Psi}_i, \bar{\boldsymbol{\lambda}}_i; H),$$

where $\boldsymbol{\Psi}_i = \boldsymbol{\Sigma}_i + \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i^\top$, $\bar{\boldsymbol{\lambda}}_i = \dfrac{\boldsymbol{\Psi}_i^{-1/2} \mathbf{Z}_i \mathbf{D} \boldsymbol{\zeta}}{\sqrt{1 + \boldsymbol{\zeta}^\top \boldsymbol{\Lambda}_i \boldsymbol{\zeta}}}$, $\boldsymbol{\zeta} = \mathbf{D}^{-1/2} \boldsymbol{\lambda}$ and $\boldsymbol{\Lambda}_i = (\mathbf{D}^{-1} + \mathbf{Z}_i^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{Z}_i)^{-1}$.

Important remarks

4. The SMSN-LMM can be written hierarchically as follows:

$$\begin{aligned}
\mathsf{Y}_i|\mathsf{b}_i, U_i = u_i &\overset{\text{ind}}{\sim} \mathrm{N}_{n_i}\left(\mathsf{X}_i\boldsymbol{\beta} + \mathsf{Z}_i\mathsf{b}_i, u_i^{-1}\sigma_e^2\mathsf{R}_i\right), \\
\mathsf{b}_i|T_i = t_i, U_i = u_i &\overset{\text{ind}}{\sim} \mathrm{N}_q\left(\boldsymbol{\Delta}t_i, u_i^{-1}\boldsymbol{\Gamma}\right), \\
T_i|U_i = u_i &\overset{\text{ind}}{\sim} \mathrm{TN}\left(0, u_i^{-1}, (0,\infty)\right), \text{ and} \\
U_i &\overset{\text{ind}}{\sim} \mathrm{H}(\cdot;\boldsymbol{\nu}),
\end{aligned}$$

   which are all independent, where $\boldsymbol{\Gamma} = \mathsf{D} - \boldsymbol{\Delta}\boldsymbol{\Delta}^\top$.

5. Bayesian approach: Lachos, Cancho and Dey (2009). Robust linear mixed models with skew-normal independent distributions from a Bayesian perspective. (Journal of Statistical Planning and Inference)

Application: Cholesterol data

	SN-LMM		ST-LMM		SCN-LMM		SSL-LMM	
Parameter	Estimate	SE	Estimate	SE	Estimate	SE	Estimate	SE
β_o	1.3520	0.1502	1.3888	0.1311	1.4045	0.1396	1.4089	0.1433
β_1	-0.0488	0.0509	-0.0548	0.0447	-0.0461	0.0468	-0.0430	0.0482
β_2	0.0152	0.0033	0.0149	0.0029	0.0144	0.0030	0.0140	0.0031
β_3	0.3562	0.0667	0.3641	0.0611	0.4006	0.0630	0.3998	0.0638
σ_e^2	0.0430	0.0017	0.0325	0.0025	0.0264	0.0028	0.0228	0.0025
d_{11}	0.5261	0.0474	0.4417	0.0477	0.4079	0.0541	0.3918	0.0472
d_{12}	0.0018	0.0302	-0.0030	0.0305	-0.0246	0.0290	-0.0232	0.0277
d_{22}	0.2166	0.0330	0.2035	0.0370	0.2099	0.0386	0.1953	0.0353
λ_1	13.8050	4.2423	13.7822	4.4242	13.4875	4.6855	14.1171	4.7110
λ_2	-6.3654	4.3984	-8.0691	3.9867	-8.7607	4.0621	-8.4215	4.2099
ν	-	-	8.1799	2.1980	0.2981	0.0865	2.0898	0.4669
γ	-	-	-	-	0.3345	0.0425	-	-
$\ell(\widehat{\boldsymbol{\theta}})$	-152.0090		-127.4155		-125.9182		-130.3672	
AIC	0.1552		0.1326		0.1321		0.1354	

Application: Cholesterol data

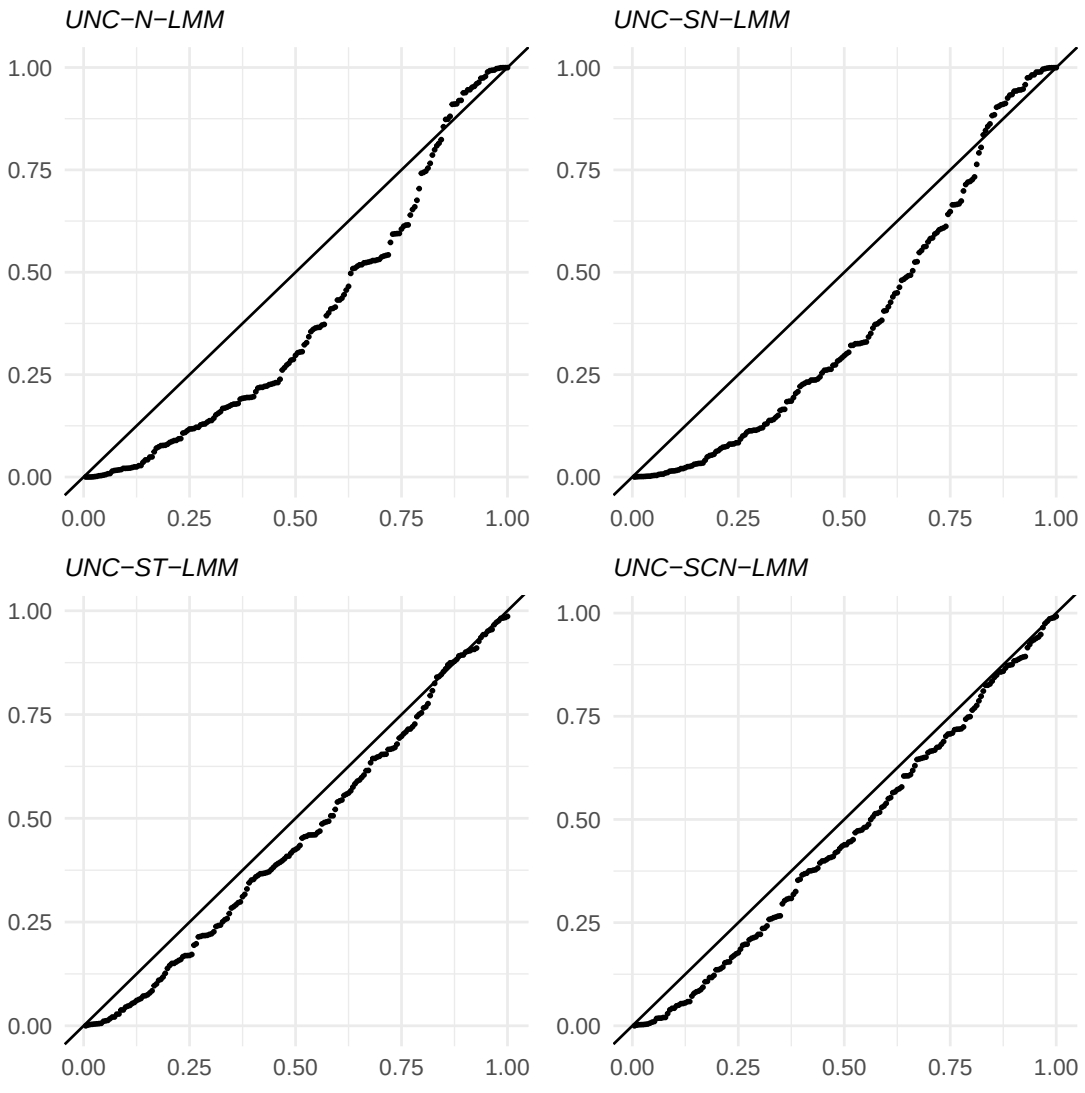


Figure 3: Healy-type plot.

Accounting for within-subject serial correlation

Within-subject dependence structures

An interesting extension is to allow for serial dependence. Schumacher, Lachos and Matos (2021) [Statistics in Medicine] considered SMSN-LMM with the following within-subject dependence structures:

1. Uncorrelated (UNC): $\mathbf{R}_i = \mathbf{I}_{n_i}$.
2. Autoregressive dependence of order p (AR(p)):

$$\mathbf{R}_i = \mathbf{R}_i(\boldsymbol{\phi}) = \frac{1}{1 - \phi_1\rho_1 - \ldots - \phi_p\rho_p}[\rho_{|r-s|}],$$

   where $\rho_1, \ldots, \rho_p$ are the theoretical autocorrelations of the process and functions of $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)^\top$, and they satisfy the Yule-Walker equations.
3. Damped exponential correlation (DEC):

$$\mathbf{R}_i = \mathbf{R}_i(\phi_1, \phi_2, \mathbf{t}_i) = \left[\phi_1^{|t_{ij} - t_{ik}|^{\phi_2}}\right], \quad 0 < \phi_1 < 1, \ \phi_2 > 0.$$

Application: Schizophrenia data

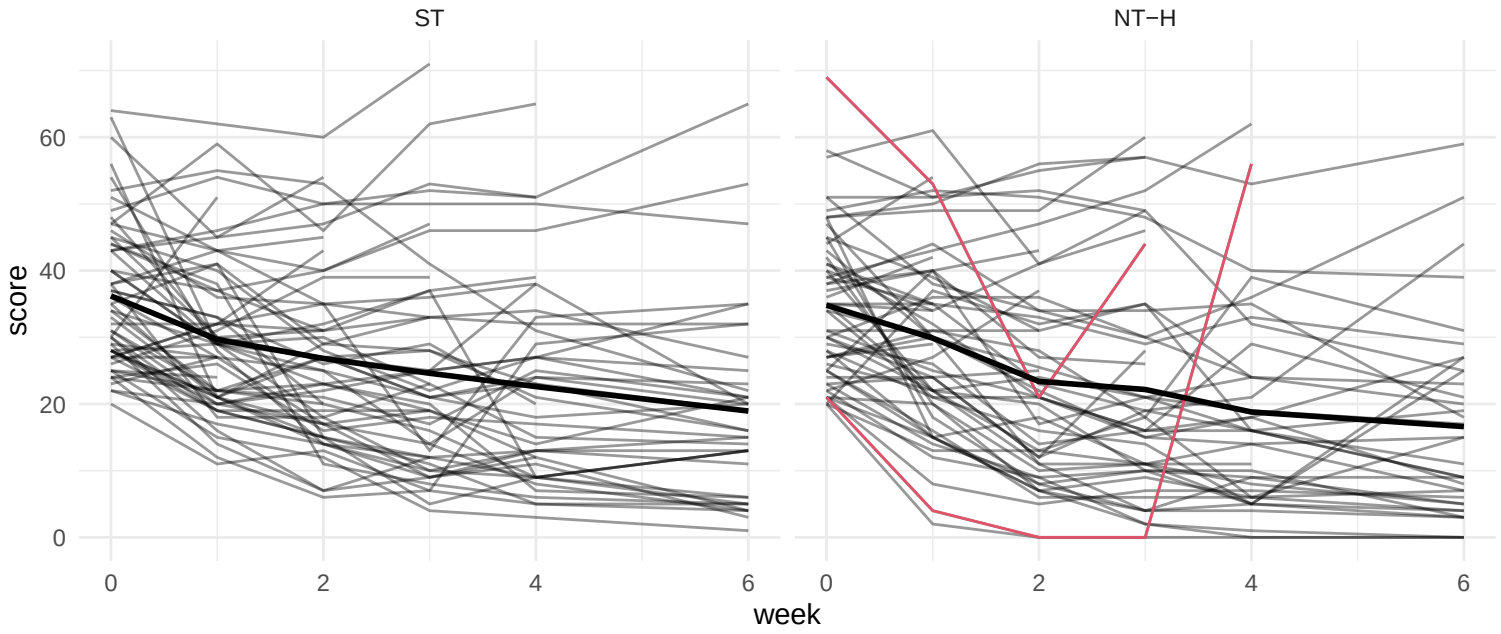


Figure 4: Trajectories of schizophrenia levels for the data.

We propose to fit the model

$$\mathbf{Y}_i = (b_{0i} + \beta_0 + \beta_1 \,\mathrm{NT}_i)\mathbf{1}_{n_i} + (b_{1i} + \beta_2 + \beta_3 \,\mathrm{NT}_i)\mathbf{x}_i + \beta_4 \,\mathbf{x}_i^2 + \boldsymbol{\epsilon}_i,$$

$i = 1, \ldots, 118$, where $\mathbf{Y}_i$ is the BRPS score vector divided by 10, x_{ij} is (time - 3)/10.

Application: Schizophrenia data

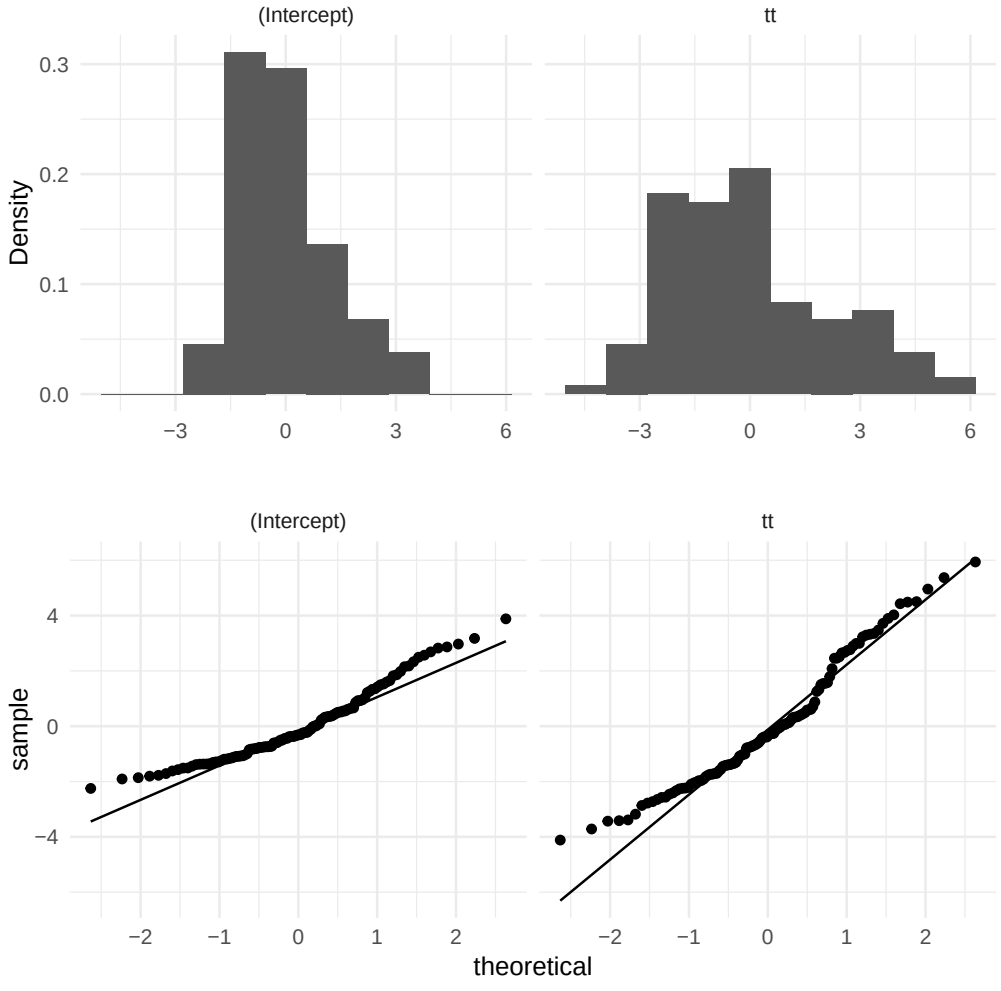


Figure 5: Empirical Bayes estimates of random effects obtained from fitting a LMM to the schizophrenia data.

Application: Schizophrenia data

Table 1: ML results for the schizophrenia data set.

Parameter	UNC-ST-LMM		AR(1)-ST-LMM			
	Estimate	Std. error	Estimate	Std. error	95% CI	
β_0	2.54	0.15	2.59	0.19	2.22	2.96
β_1	-0.22	0.15	-0.19	0.19	-0.57	0.19
β_2	-1.27	0.37	-1.49	0.49	-2.46	-0.53
β_3	-0.30	0.44	-0.15	0.49	-1.11	0.81
β_4	6.62	0.50	6.08	0.75	4.61	7.54
σ_e^2	0.20	0.02	0.27	0.04		
ϕ	-	-	0.60	0.16		
F_{11}	1.36	0.14	1.14	0.38		
F_{12}	1.13	0.14	1.23	0.28		
F_{22}	2.98	0.36	2.36	0.66		
λ_1	65.92	-	12.74	-		
λ_2	75.21	-	16.01	-		
ν	4.14	-	4.07	-		

Application: Schizophrenia data

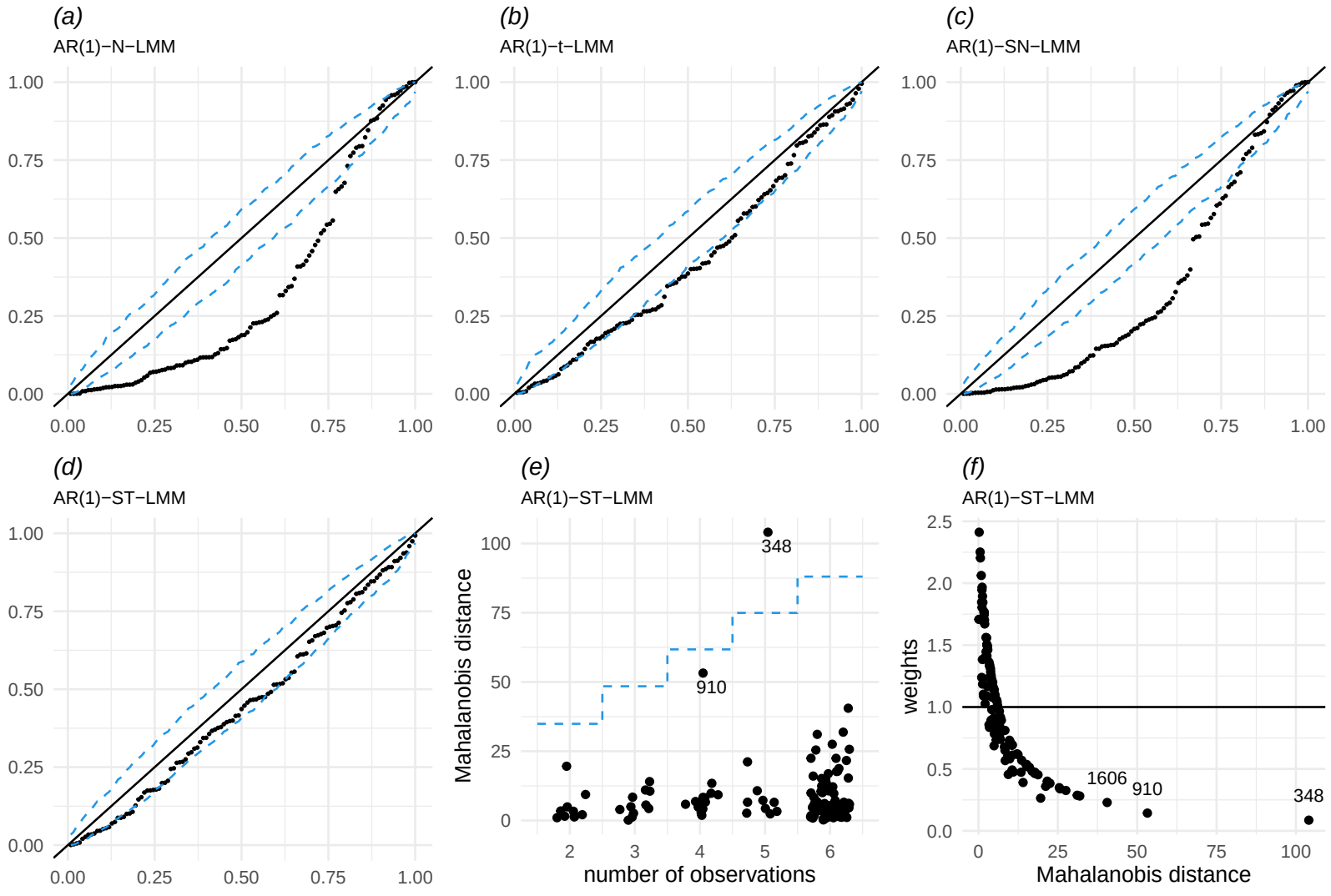


Figure 6: Model evaluation.

Application: Schizophrenia data

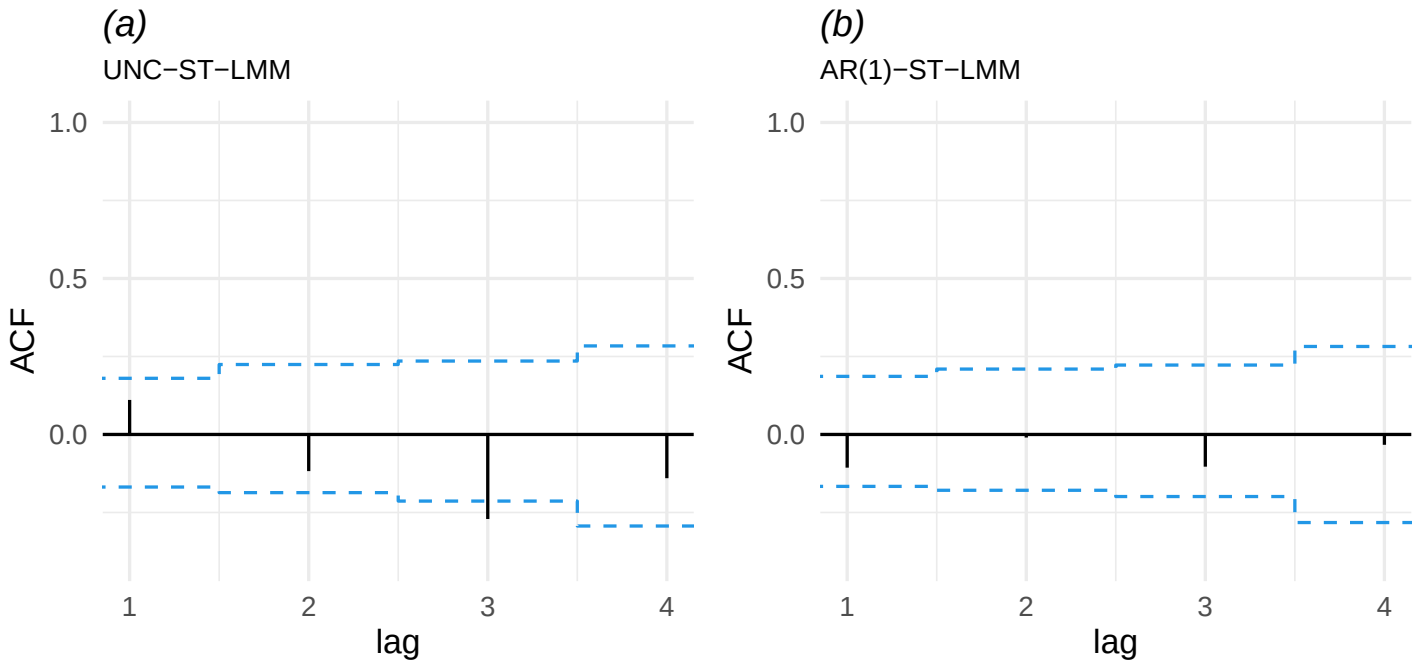


Figure 7: Autocorrelation plots.

Schizophrenia data

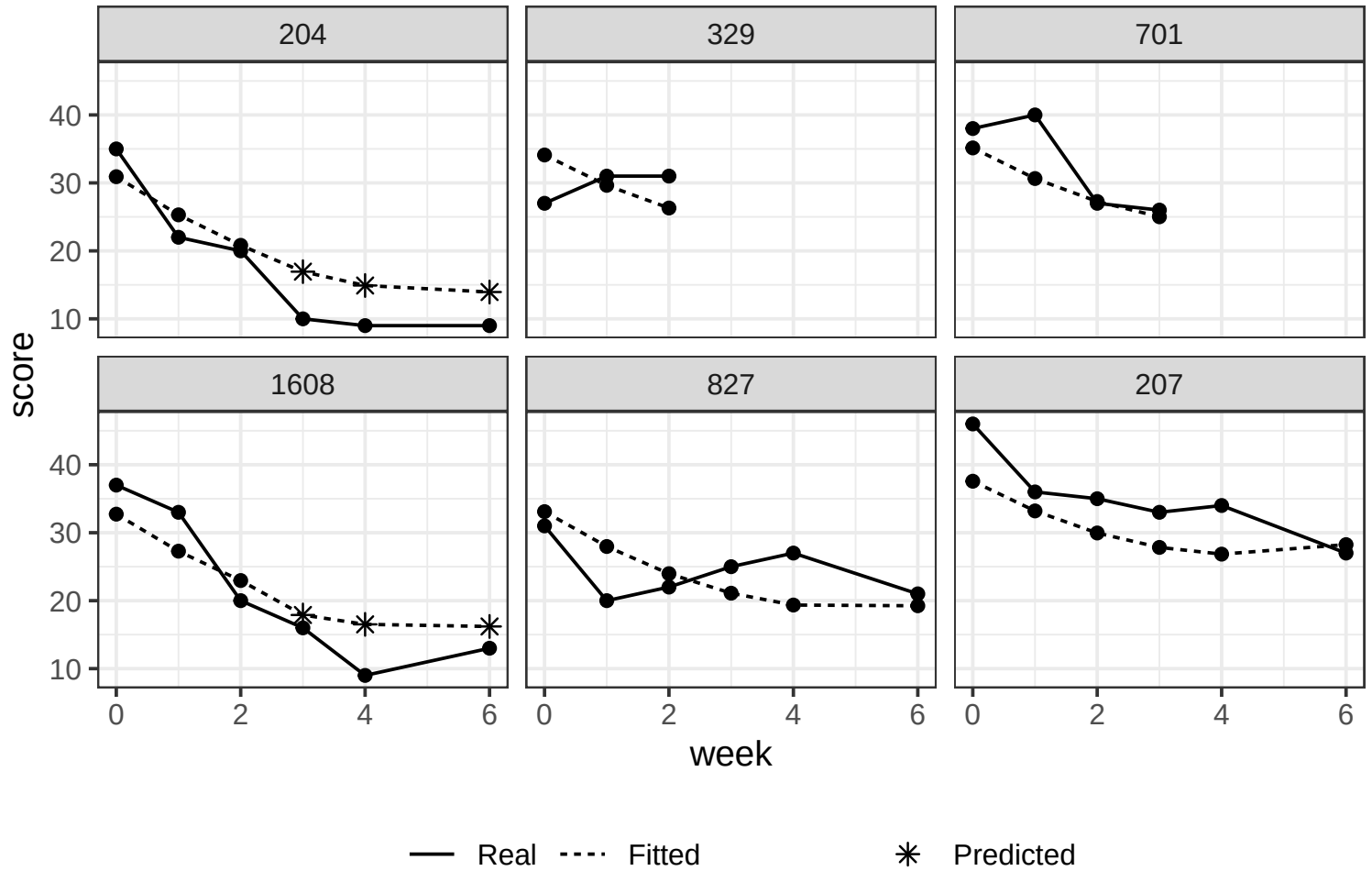


Figure 8: Evaluation of fit and prediction for six random subjects, who are identified by their ID.

Fitting LMM in R

Fitting LMM in R

- Two main popular functions to fit LMMs in R:
  - *lme()* in the **nlme** package: supports several random effects and error level dependence structures.
  - *lmer()* in the **lme4** package: has more efficient linear algebra tools and is more efficient for fitting models with crossed random effects, but does not support special dependence structures.

- Both these functions assume normal distributions for the random terms (although the function *glmer()* in the **lme4** package fits generalized linear mixed effect models).

Fitting LMM in R

R package	Function	Approach	Details/Assumptions
nlme	*lme*	Classic	Optimized for nested hierarchical structures; allows within-subject correlation; symmetric
lme4	*lmer*	Classic	Efficient for crossed random effects; no within-subject correlation; symmetric
lqmm	*lqmm*	Quantile-based	Allows median-type estimates; no within-subject correlation; symmetric
heavy	*heavyLme*	t distributions	No within-subject correlation; symmetric; possible bug in the software
robustlmm	*rlmer*	Huberization of likelihood and DAS-scale estimation	No within-subject correlation; symmetric
nlmm	*nlmm*	Generalized (symmetric) Laplace distribution	Allows heteroscedasticity in the errors; no within-subject correlation; symmetric
ngme	*ngme*	Generalized hyperbolic distributions	Additional random term; allows skewness; allows within-subject correlation; not available on CRAN
skewlmm	*smsn.lmm*	SMSN distributions	Allows within-subject correlation; skewed
	smn.lmm	SMN distributions	Allows within-subject correlation; symmetric

Source: adapted from Table 1 in in Koller (2016)

The R package skewlmm

The R package skewlmm

The **skewlmm** package can be installed from GitHub as follows:

```
devtools::install_github("fernandalschumacher/skewlmm")
```

Or its released version can be downloaded from CRAN as follows:

```
install.packages("skewlmm")
```

The R package skewlmm

Two main functions are available: *smsn.lmm()* and *smn.lmm()*, that fit SMSN-LMMs and SMN-LMMs, respectively.

```
smsn.lmm(data, formFixed, groupVar, formRandom = ~1,
         depStruct = "UNC", timeVar = NULL, distr = "sn",
         covRandom = "pdSymm", skewind, pAR = 1,
         control = lmmControl())

smn.lmm(data, formFixed, groupVar, formRandom = ~1,
        depStruct = "UNC", timeVar = NULL, distr = "norm",
        covRandom = "pdSymm", pAR = 1,
        control = lmmControl())
```

The R package skewlmm

- `data` is a data frame.
- `formFixed` is a two-sided linear formula (e.g. `y ~ x1 + x2`).
- `groupVar` is the name of the group variable.
- `formRandom` is a one-sided linear formula (e.g. `~ x1`).
- `depStruct`: `"UNC"`, `"ARp"`, `"DEC`, `"CS"`, or `"CAR1"`.
- `timeVar` is the name of the time variable.
- `distr` (symmetrical): `"norm"`, `"t"`, `"sl"`, or `"cn"`
- `distr` (skewed): `"sn"`, `"st"`, `"ssl"`, or `"scn"`
- `covRandom` is the random effects scale matrix (`"pdSymm"` or `"pdDiag"`).
- `skewind` indicates which elements of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_q)$ should be estimated.
- `pAR` is the order of the autoregressive process.
- `control` additional control via the function `lmmControl()`.

The R package skewlmm

These functions return objects of the class *SMSN* and *SMN*, respectively, containing a list of elements, and the following methods/functions are available to these classes:

- *update*
- *boot_par*
- *boot_ci*
- *print*
- *summary*
- *fitted*
- *ranef*
- *predict*
- *plot*
- *residuals*
- *acfresid*
- *healy.plot*
- *mahalDist*
- *lr.test*
- *criteria*

Computational aspects

- The required computation time to estimate EM algorithms.
- We use the R package **optimParallel** (Gerber and Furrer, 2019) to (try to) accelerate numerical optimization.
- Further, there are several proposals in the literature to accelerate the often-slow convergence of the EM algorithm.
- We use of the *Damped Anderson Acceleration With Restarts and Monotonicity Control for Accelerating EM (DAAREM)* algorithms (Henderson and Varadhan, 2019), to accelerate the convergence of the EM-type algorithm used in the estimation process.

DAAREM algorithm

- The DAAREM algorithm is built on the *Anderson acceleration* (AA) technique (Anderson, 1965), which is itself an EM accelerator that uses the current and past iterates of the sequence of parameter values and the corresponding EM mappings of these parameter values.

- It incorporates the following modifications:
  1. “restarts” for the order of the extrapolation;
  2. damped extrapolations;
  3. “epsilon-monotonicity”.

Computational aspects

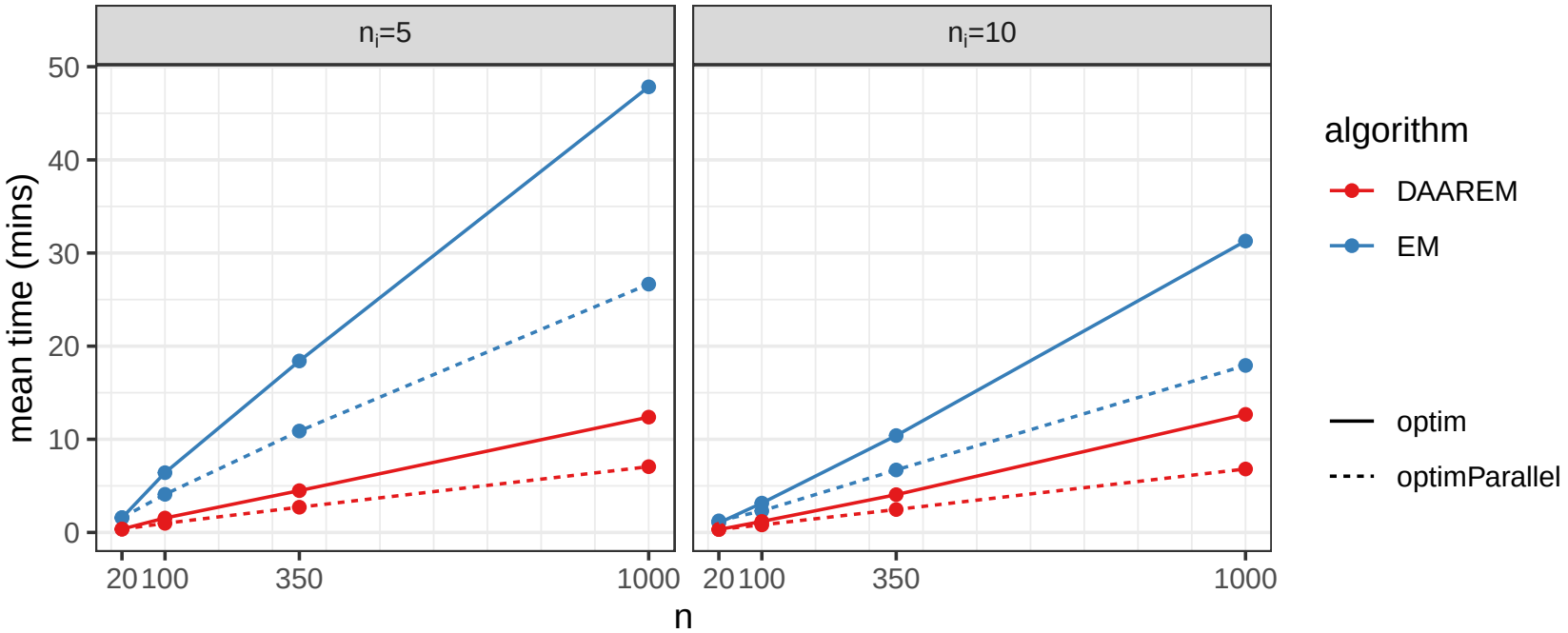


Figure 9: Mean time for fitting AR(2)-ST-LMMs.

Application to sleep deprivation data

We can fit a classic model N-LMM, and, for example, an SL-LMM, and an SSL-LMM using the following code:

```
fit_norm <- smn.lmm(data = sleepstudy,
                    formFixed = Reaction ~ Dayst,
                    formRandom = ~Dayst, groupVar = "Subject")

fit_sl <- update(object = fit_norm, distr = "sl")

fit_ssl <- smsn.lmm(data = sleepstudy,
                    formFixed = Reaction ~ Dayst,
                    formRandom = ~Dayst, groupVar = "Subject",
                    distr = "ssl")
```

Application to sleep deprivation data

We can perform a likelihood ratio test for testing $H_0 : \boldsymbol{\lambda} = \mathbf{0}$ using the function *lr.test()* and the two nested fitted models:

```
lr.test(fit_sl, fit_ssl)
```

```
##
## Model selection criteria:
##           logLik      AIC      BIC
## fit_sl  -685.634 1385.269 1406.057
## fit_ssl -685.402 1388.804 1415.532
##
##     Likelihood-ratio Test
##
## chi-square statistics =  0.464485
## df =  2
## p-value =  0.7927539
##
## The null hypothesis that both models represent the
## data equally well is not rejected at level  0.05
```

Application to sleep deprivation data

To evaluate the adequacy of the distributional assumption, we can produce Healy-type plots as follows:

```
grid.arrange(healy.plot(fit_norm, calcCI = TRUE),
             healy.plot(fit_sl, calcCI = TRUE), nrow=1)
```

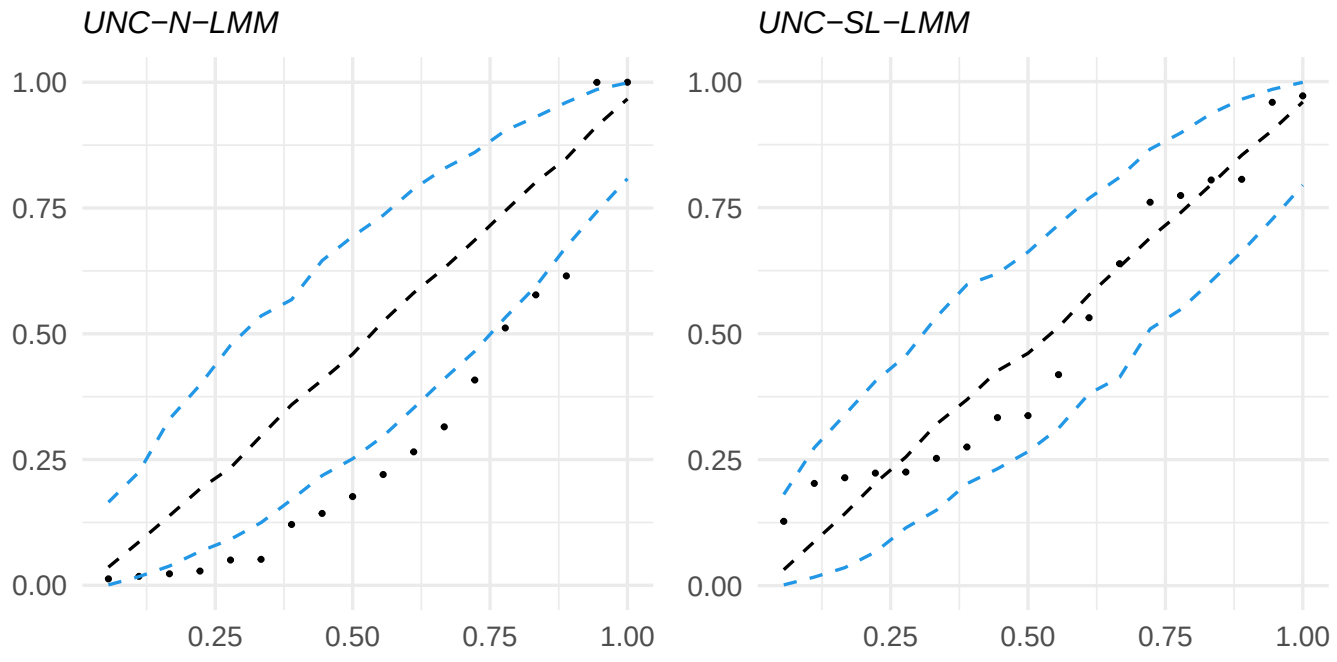


Figure 10: Healy-type plots for the *sleepdata* data set.

Application to sleep deprivation data

We can consider different dependence structures, as follows:

```
fit_sl_ar1 <- update(fit_sl, depStruct = "ARp", pAR = 1)
fit_sl_ar2 <- update(fit_sl, depStruct = "ARp", pAR = 2)
fit_sl_DEC <- update(fit_sl, depStruct = "DEC",
                     timeVar ="Days")
```

```
criteria(list(UNC = fit_sl, AR1 = fit_sl_ar1,
              AR2 = fit_sl_ar2, DEC = fit_sl_DEC))
```

depStruct	logLik	npar	AIC	BIC
UNC	-685.6343	7	1385.269	1406.057
AR1	-682.5440	8	1381.088	1404.846
AR2	-681.1045	9	1380.209	1406.937
DEC	-681.8286	9	1381.657	1408.385

Application to sleep deprivation data

We can extract information about the fit using the method *summary*:

```
summary(fit_sl_ar1)
```

```
## Linear mixed models with distribution sl and dependency structure ARp
## Call:
## smn.lmm(data = sleepstudy, formFixed = Reaction ~ Dayst,
##         groupVar = "Subject", formRandom = ~Dayst,
##         depStruct = "ARp", distr = "sl", pAR = 1)
##
## Distribution sl with nu = 1.223598
##
## Random effects:
##   Formula: ~Dayst
##   Structure: General positive-definite
##   Estimated variance (D):
##             (Intercept)     Dayst
## (Intercept)    987.5287 111.21641
## Dayst          111.2164  18.64026
```

Application to sleep deprivation data

```
##
## Fixed effects: Reaction ~ Dayst
## with approximate confidence intervals
##                 Value Std.error CI 95% lower CI 95% upper
## (Intercept) 303.09140 10.563667   282.386993    323.79581
## Dayst        11.26655  1.659266     8.014453     14.51865
##
## Dependency structure: ARp
##   Estimate(s):
##      sigma2         phi1
## 187.2645087   0.4553367
##
## Model selection criteria:
##    logLik      AIC      BIC
##  -682.544 1381.088 1404.846
##
## Number of observations: 144
## Number of groups: 18
```

Application to sleep deprivation data

When dealing with small samples, it may be helpful to compute bootstrap confidence intervals (based on B simulated samples), which can be done using the functions *boot_par()* and *boot_ci()* as follows:

```
boot_sl_ar1 <- boot_par(fit_sl_ar1, B = 100)
boot_ci(boot_sl_ar1) %>% kable(digits = 2)
```

	(Intercept)	Dayst	sigma2	phiAR1	Dsqrt11	Dsqrt12	Dsqrt22	nu1
2.5%	287.72	8.11	111.30	0.13	19.95	0.77	0.73	0.72
97.5%	323.78	15.05	386.03	0.69	55.02	6.21	5.06	17.79

Application to sleep deprivation data

Plotting the fitted object results in the following figure:

```
plot(fit_sl_ar1,type = "normalized")
```

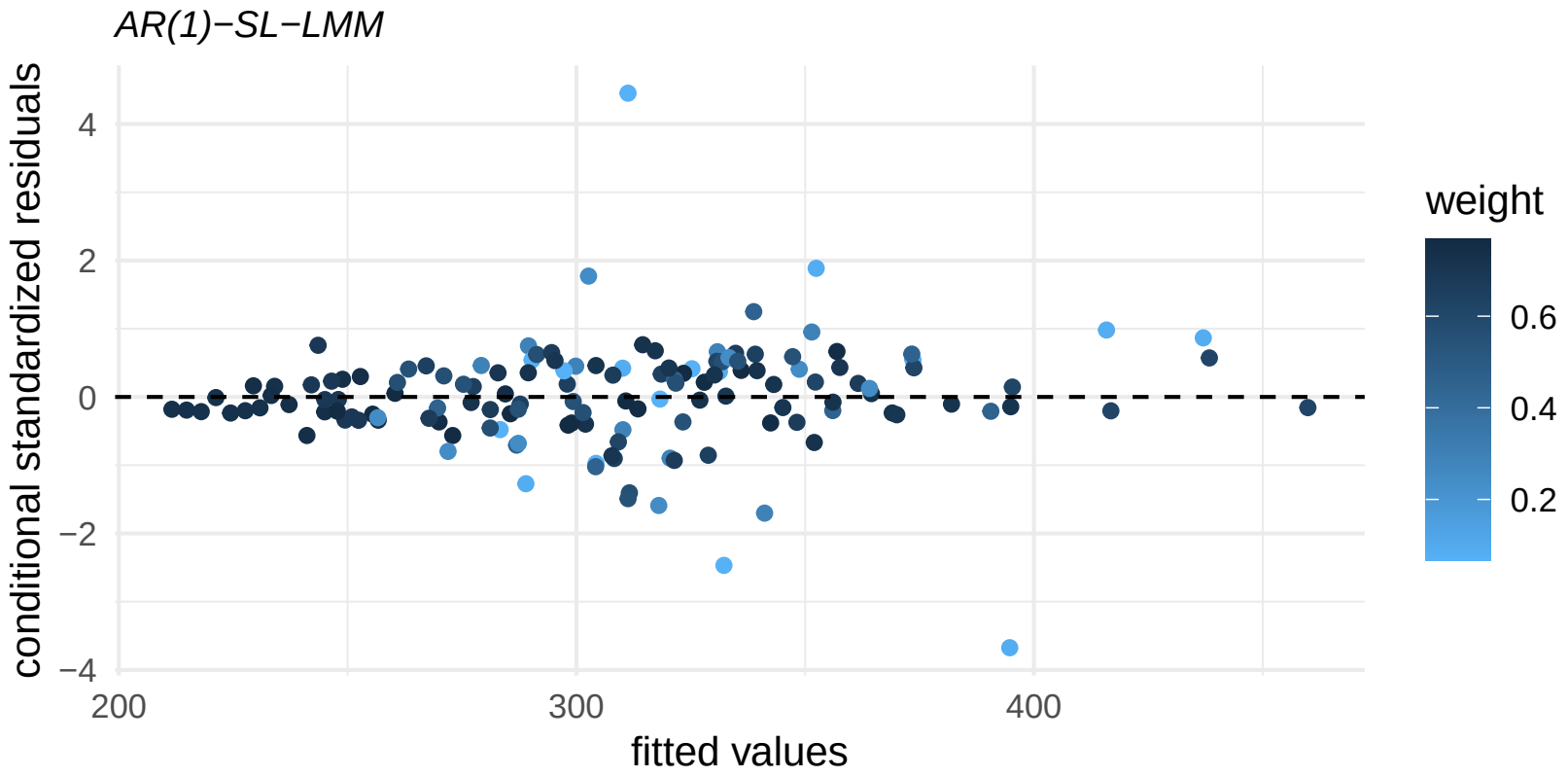


Figure 11: Plotting the `sleepdata` data set fitted object.

Practical activity

Practical activity

To reproduce the example, please make sure that have the following packages:

- *skewlmm*
- *tidyverse*
- *nlme*
- *lme4*
- *gridExtra*
- *knitr*

LMM with censored response

LMM with censored response

We assume that the response Y_{ij} is not fully observed for all i, j.

Let the observed data for the i-th subject be $(\mathbf{V}_i, \mathbf{C}_i)$, where

- $\mathbf{V}_i$ represents the vector of uncensored readings or censoring level,
- $\mathbf{C}_i$ is the vector of censoring indicators,

such that, for left-censoring, we have

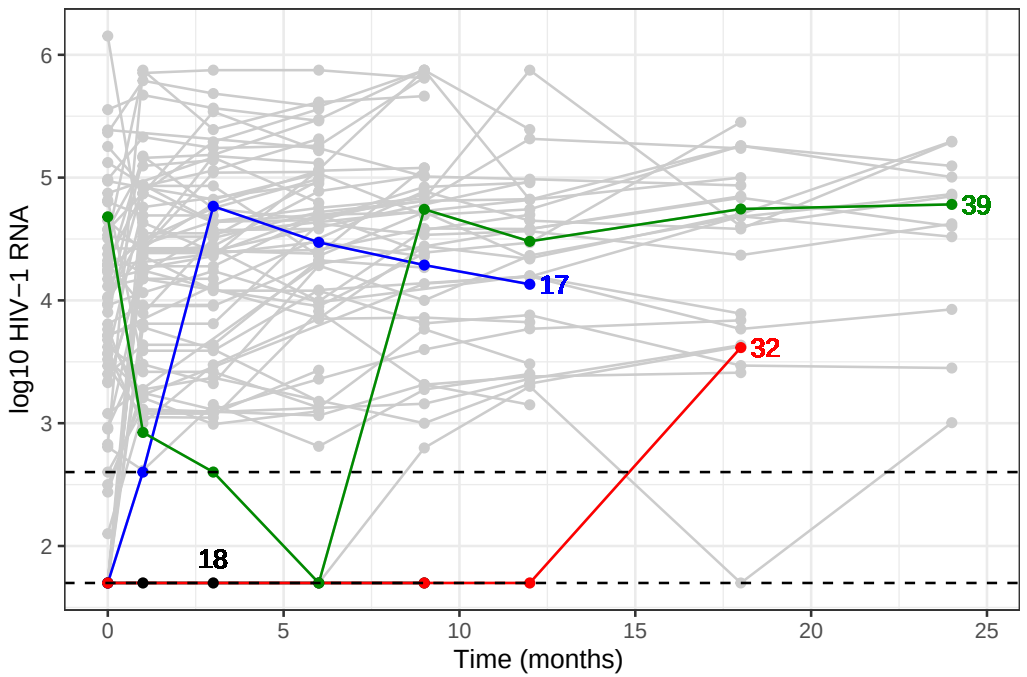
$$\begin{aligned} Y_{ij} &\leq V_{ij} \quad \text{if} \quad C_{ij} = 1, \\ Y_{ij} &= V_{ij} \quad \text{if} \quad C_{ij} = 0. \end{aligned}$$

The extensions to arbitrary censoring are immediate.

Motivating data - UTI data

Unstructured treatment interruption - UTI data

- 72 perinatally HIV-infected children (Saitoh et al. 2008);
- The viral loads were monitored at 0, 1, 3, 6, 9, 12, 18, and 24 months after the treatment interruption;
- 7% of the data (26 observations) were below the detection limits (50 or 400 copies/mL) and considered left-censored at these values.



Some publications related to the LMM with censored response

- Vaida, F. and Liu, L., 2009. Fast implementation for normal mixed effects models with censored response. Journal of Computational and Graphical Statistics, 18(4), pp.797-817.
- Matos, L.A., Prates, M.O., Chen, M.H. and Lachos, V.H., 2013. Likelihood-based inference for mixed-effects models with censored response using the multivariate-t distribution. Statistica Sinica, pp.1323-1345.
- Matos, L.A., Castro, L.M. and Lachos, V.H., 2016. Censored mixed-effects models for irregularly observed repeated measures with applications to HIV viral loads. Test, 25(4), pp.627-653.
- Olivari, R.C., Garay, A.M., Lachos, V.H. and Matos, L.A., 2021. Mixed-effects models for censored data with autoregressive errors. Journal of Biopharmaceutical Statistics, 31(3), pp.273-294.

The R package ARpLMEC

```
library(ARpLMEC)
```

Imports: *numDeriv*, *stats*, *MASS*, *mnormt*, *tclt2k*, *expm*, *relliptical*, *TruncatedNormal*, *LaplacesDemon*.

Two functions:

- `ARpMMEC.est`: Fits left, right or intervalar censored mixed-effects linear model, within-subject dependence structures;
- `ARpMMEC.sim`: Generate data from LMM with censored response.

UTI data

Reading the data:

```
library(tlmec) # UTIdata
data(UTIdata)
data1 <- subset(UTIdata, !is.na(RNA))
data1 <- data1 %>% add_count(Patid) %>% arrange(Patid,Fup)
data1 <- data1 %>% filter(n > 2) %>% droplevels()

data1 <- data1 %>%
  mutate(Patid.1 = as.numeric(Patid),.after=Patid) %>%
  mutate(y = log10(RNA)) %>%
  mutate(cc = if_else(RNAcens==1, 1, 0)) %>%
  mutate(tt = Fup)

subjects <- data1 %>% distinct(Patid) %>% pull(Patid)
```

The *ARpMMEC.est* function

```
ARpMMEC.est(y, x, z, tt, cc, nj, struc = "UNC", order = 1,
            nu.fixed = TRUE, typeModel = "Normal",
            cens.type = "left", LI = NULL, LS = NULL,
            initial = NULL, MaxIter = 200, error = 1e-04,
            Prev = FALSE, step = NULL, isubj = NULL, xpre = NULL, zpre = NULL)
```

Value

returns list of class “ARpMMEC”:

`FixEffect`	Data frame with: estimate, standar errors and confidence intervals of the fixed effects.
`Sigma2`	Data frame with: estimate, standar errors and confidence intervals of the variance of the white noise process.
`Phi`	Data frame with: estimate, standar errors and confidence intervals of the autoregressive parameters.
`RandEffect`	Data frame with: estimate, standar errors and confidence intervals of the random effects.
`nu`	the parameter "nu" for the t-student distribution
`Est`	Vector of parameters estimate (fixed Effects, sigma2, phi, random effects).
`SE`	Vector of the standard errors of (fixed Effects, sigma2, phi, random effects).
`Residual`	Vector of the marginal residuals.
`loglik`	Log-likelihood value.
`AIC`	Akaike information criterion.
`BIC`	Bayesian information criterion.
`AICc`	Corrected Akaike information criterion.
`iter`	Number of iterations until convergence.

UTI data

We will consider a profile LME model with random intercepts b_i as

$$y_{ij} = b_i + \beta_j + \epsilon_{ij},$$

where y_{ij} is the $\log_{10}$ HIV RNA for subject i at time t_j, $t_1 = 0$, $t_2 = 1$, $t_3 = 3$, $t_4 = 6$, $t_5 = 9$, $t_6 = 12$, $t_7 = 18$, $t_8 = 24$.

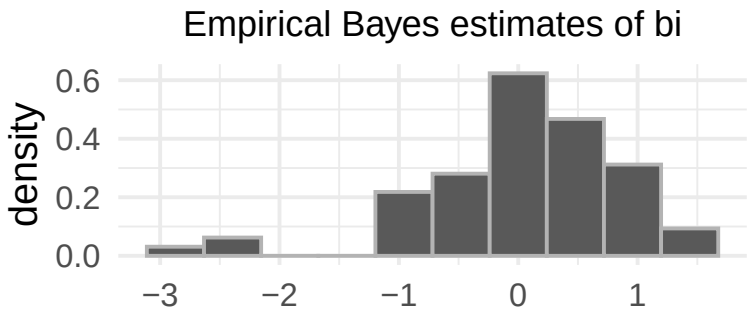
```
y  <- data1$y
cc <- data1$cc
tt <- data1$tt
nj <- data1 %>% group_by(Patid) %>% filter(row_number()==1) %>% pull(n)
x  <- as.matrix(dummy_columns(data1$Fup, remove_selected_columns=TRUE))
z  <- matrix(rep(1,length(y)), ncol=1)
```

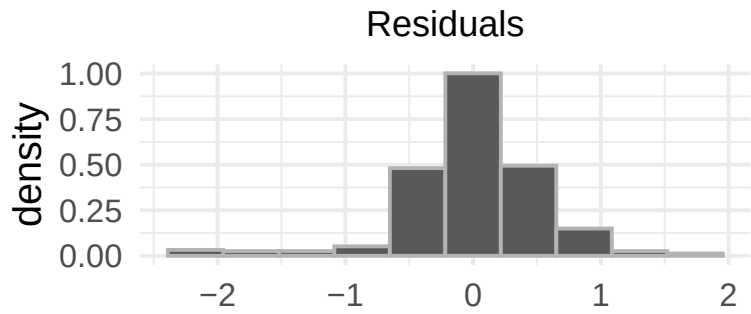
```r
fitNunc <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj)
```

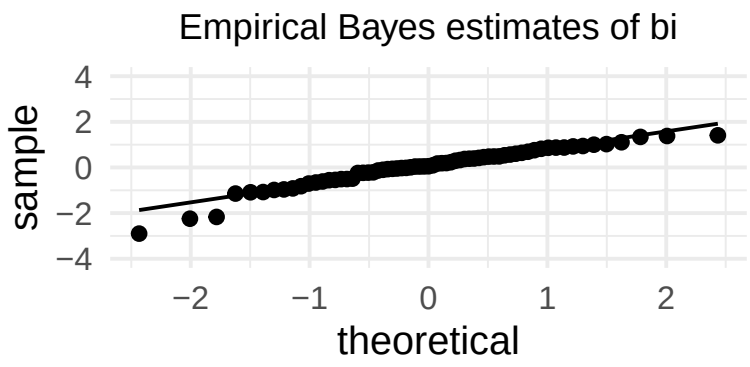
```
## ------------------------------------------------
## DEC censored mixed-effects models
## ------------------------------------------------
## Case = UNC
## Distribution = Normal
## Subjects = 67 ; Observations = 354
##
## -----------
## Estimates
## -----------
##
## - Fixed effects
##          Est    SE        IConf(95%)
## beta 1 3.516 0.137 < 3.247 , 3.785 >
## beta 2 4.101 0.177 < 3.754 , 4.448 >
## beta 3 4.179 0.212 < 3.763 , 4.595 >
## beta 4 4.299 0.203 < 3.901 , 4.697 >
## beta 5 4.503 0.221  < 4.07 , 4.936 >
## beta 6 4.505 0.245 < 4.025 , 4.985 >
## beta 7 4.613 0.215 < 4.192 , 5.034 >
## beta 8 4.730 0.361 < 4.022 , 5.438 >
##
## - Sigma^2
##           Est    SE        IConf(95%)
## Sigma^2 0.345 0.021 < 0.304 , 0.386 >
##
## - Random effects
##            Est    SE        IConf(95%)
## Alpha 11 0.755 0.127 < 0.506 , 1.004 >
```

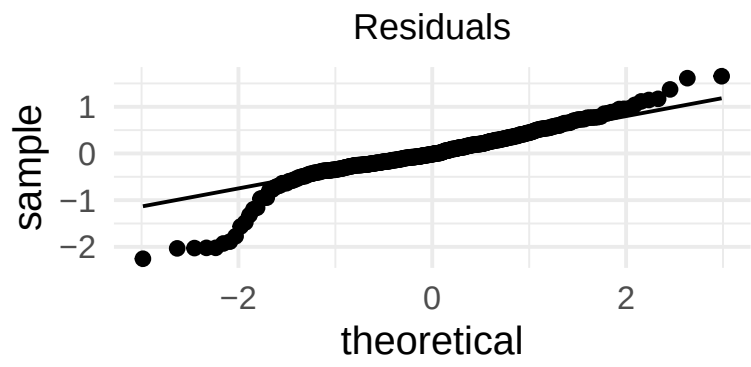
```
## ------------------------
## Model selection criteria
## ------------------------
##
##         Loglik     AIC     BIC
## Value -402.001 824.002 862.695
##
## -------
## Details
## -------
##
## Convergence reached? = TRUE
## Iterations = 6 / 200
## Processing time = 2.267847 secs
```

```
resid <- data.frame(resid=apply(fitNunc$others$uyi,1,sum)-fitNunc$Yfit)
bi <- data.frame(bi=apply(fitNunc$others$ubi,1,sum))
```









```r
fitNdec <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="DEC")

fitNdAR <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="DEC(AR)")

fitNsym <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="SYM")

fitNAR1 <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="ARp",order=1)

fitNAR2 <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="ARp",order=2)
```

depStruct	log.lik	AIC	BIC
UNC	-402.001	824.002	862.695
SYM	-401.996	825.993	868.555
DEC	-401.936	827.872	874.304
DEC(AR)	-402.015	826.029	868.592
AR(1)	-399.815	821.630	864.193
AR(2)	**-394.538**	**813.075**	**859.507**

Student-t

```r
fitTdec <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="DEC",
                       typeModel="Student",nu.fixed=FALSE)
```

```
## ------------------------------------------------
## DEC censored mixed-effects models
## ------------------------------------------------
## Case = DEC
## Distribution = Student
## nu = 2.01263
## Subjects = 67 ; Observations = 354
##
## -----------
## Estimates
## -----------
##
## - Fixed effects
##          Est    SE        IConf(95%)
## beta 1 3.921 0.151 < 3.275 , 4.567 >
## beta 2 4.209 0.146 < 3.585 , 4.833 >
## beta 3 4.247 0.146 < 3.623 , 4.871 >
## beta 4 4.431 0.144 < 3.815 , 5.047 >
## beta 5 4.574 0.149 < 3.937 , 5.211 >
## beta 6 4.575 0.152 < 3.925 , 5.225 >
## beta 7 4.594 0.166 < 3.884 , 5.304 >
## beta 8 4.774 0.196 < 3.936 , 5.612 >
##
## - Sigma^2
##          Est
## Sigma^2 0.22
##
## - Autoregressives parameters
##         Est
## Phi 1 0.421
## Phi 2 0.115
##
## - Random effects
##            Est
## Alpha 11 0.507
##
## ------------------------
## Model selection criteria
## ------------------------
##
##         Loglik     AIC     BIC
## Value -309.068 644.135 694.436
##
## -------
## Details
## -------
##
## Convergence reached? = TRUE
## Iterations = 15 / 200
## Processing time = 1.128532 mins
```

```
fitTunc <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="UNC",
                       typeModel="Student",nu.fixed=FALSE)
fitTdAR <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="DEC(AR)",
                       typeModel="Student",nu.fixed=FALSE)
fitTsym <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="SYM",
                       typeModel="Student",nu.fixed=FALSE)
fitTAR1 <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="ARp",order=1,
                       typeModel="Student",nu.fixed=FALSE)
fitTAR2 <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc,nj=nj,struc="ARp",order=2,
                       typeModel="Student",nu.fixed=FALSE)
```

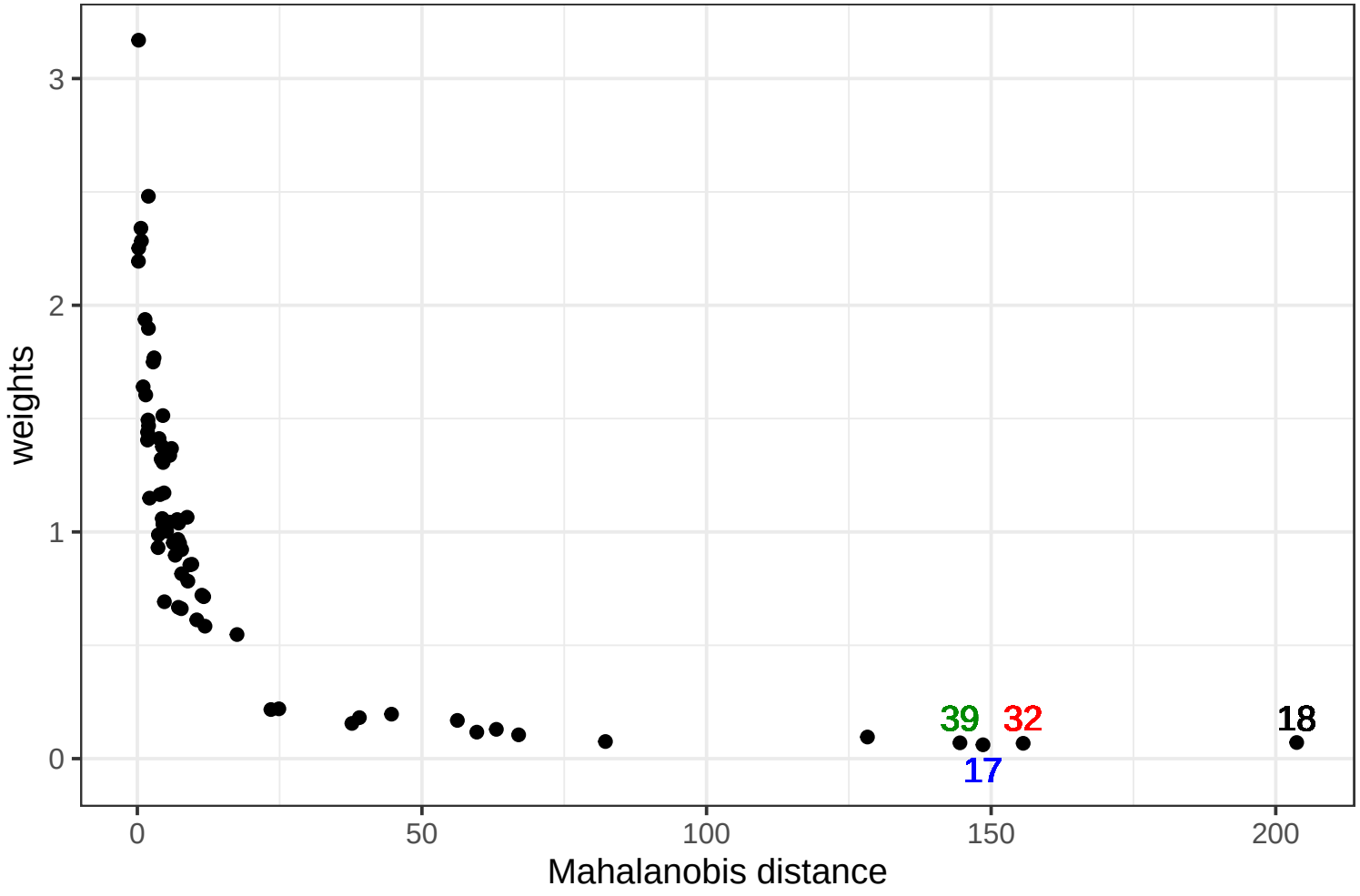
depStruct	log.lik	AIC	BIC
UNC	-308.838	641.676	688.108
SYM	-310.553	645.106	691.538
DEC	-309.068	644.135	694.436
DEC(AR)	-310.686	645.372	691.803
AR(1)	-305.647	635.294	681.726
AR(2)	**-291.823**	**609.646**	**659.947**

UTI data: t-AR(2) model

Best model estimates:

	Estimate	SE
β_1	3.961	0.110
β_2	4.247	0.138
β_3	4.275	0.136
β_4	4.470	0.136
β_5	4.604	0.140
β_6	4.584	0.141
β_7	4.582	0.165
β_8	4.779	0.194
σ^2	0.076	-
ϕ_1	0.153	-
ϕ_2	0.773	-
α	0.203	-
ν	2.016	-

UTI data: t-AR(2) model



The *ARpMMEC.sim* function

```
ARpMMEC.sim(m, x = NULL, z = NULL, tt = NULL, nj,
            beta, sigmae, D, phi,
            struc = "ARp", order = 1,
            typeModel = "Normal", nu = NULL,
            p.cens = NULL, n.cens = NULL, cens.type = "left")
```

Value

returns list:

`cc` Vector of censoring indicators.

`y_cc` Vector of responses censoring.

Simulated data

We will consider the model as

$$y_{ij} = (b_{0i} + \beta_0) + (b_{1i} + \beta_1)t_{ij} + \epsilon_{ij},$$

where $t_i = (0, 1, 2, 3, 4, 5, 6, 7, 8)$.

```
set.seed(310822)
m <- 150
t <- 1:8
D <- matrix(c(2,0,0,0.5),2,2)
sigma2 <- 2
phi <- 0.5
beta <- c(10,1.5)

tt <- rep(t,m)
nj <- rep(length(t),m)
x <- cbind(rep(1,length(tt)),tt)
z <- cbind(rep(1,length(tt)),tt)

p.cens  = 0.1
data=ARpMMEC.sim(m, x, z, tt, nj, beta, sigma2, D, phi,
                 struc="ARp", order=1, typeModel="Normal",
                 p.cens=p.cens, cens.type = "right")
```

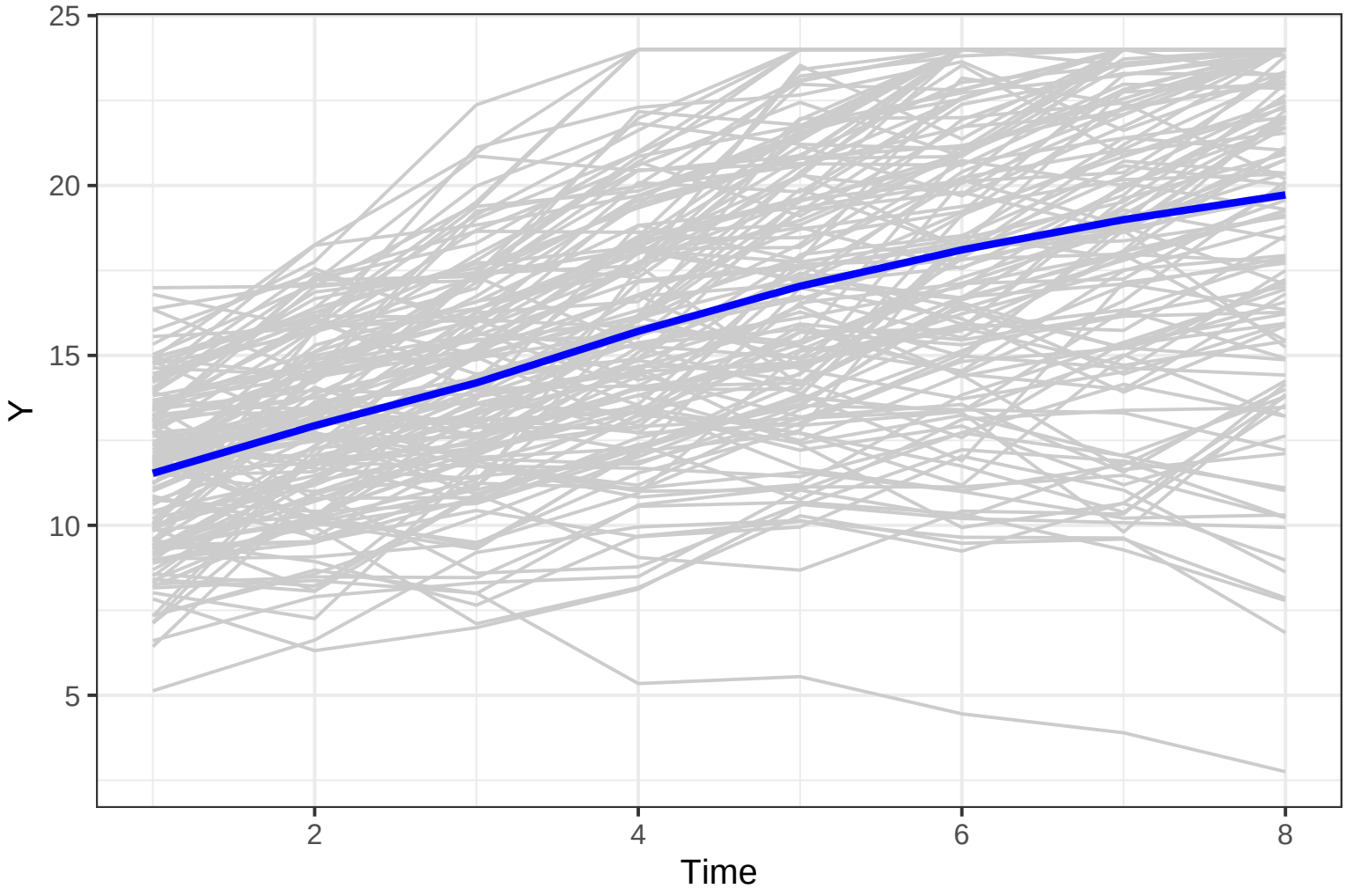


Figure 12: Simulated data with 10% of right-censoring.

Fitting the simulated data

```
y =  matrix(data$y_cc, ncol = 1)
cc1 = matrix(data$cc, ncol = 1)
tt <- rep(t,m)
nj <- rep(length(t),m)
x <- cbind(rep(1,length(tt)),tt)
z <- cbind(rep(1,length(tt)),tt)

fitN <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc1,nj=nj,
                    struc="ARp",order=1,cens.type="right")
```

```
## ------------------------------------------------------
## Autoregressive censored mixed-effects models
## ------------------------------------------------------
## Autoregressive order = 1
## Distribution = Normal
## Subjects = 150 ; Observations = 1200
##
## -----------
## Estimates
## -----------
##
## - Fixed effects
##           Est    SE         IConf(95%)
## beta 1 10.166 0.178 < 9.817 , 10.515 >
## beta 2  1.294 0.075  < 1.147 , 1.441 >
##
## - Sigma^2
##           Est    SE        IConf(95%)
## Sigma^2 2.209 0.112 < 1.989 , 2.429 >
##
## - Autoregressives parameters
##         Est    SE       IConf(95%)
## Phi 1 0.658 0.047 < 0.566 , 0.75 >
## - Random effects
##
##             Est    SE         IConf(95%)
## Alpha 11 0.133 0.280      < 0 , 0.682 >
## Alpha 12 0.097 0.239 < -0.371 , 0.565 >
## Alpha 22 0.574 0.150   < 0.28 , 0.868 >
##
##
## ------------------------
## Model selection criteria
## ------------------------
##
##          Loglik      AIC      BIC
## Value -2226.942 4467.884 4503.514
##
## -------
## Details
## -------
##
## Convergence reached? = TRUE
## Iterations = 4 / 200
## Processing time = 22.97717 secs
```

Intervalar censoring:

```
LI = y
LS = matrix(rep(40,length(y)), ncol = 1)
fitN_I <- ARpMMEC.est(y=y,x=x,z=z,tt=tt,cc=cc1,nj=nj,struc="ARp",order=1,
                      cens.type="interval",LI=LI,LS=LS)
```

```
## ------------------------------------------------
## Autoregressive censored mixed-effects models
## ------------------------------------------------
## Autoregressive order = 1
## Distribution = Normal
## Subjects = 150 ; Observations = 1200
##
## -----------
## Estimates
## -----------
##
## - Fixed effects
##           Est    SE         IConf(95%)
## beta 1 10.170 0.178 < 9.821 , 10.519 >
## beta 2  1.293 0.075   < 1.146 , 1.44 >
##
## - Sigma^2
##          Est    SE      IConf(95%)
## Sigma^2 2.21 0.112 < 1.99 , 2.43 >
##
## - Autoregressives parameters
##         Est    SE        IConf(95%)
## Phi 1 0.659 0.047 < 0.567 , 0.751 >
## - Random effects
##             Est    SE        IConf(95%)
## Alpha 11 0.133 0.279      < 0 , 0.68 >
## Alpha 12 0.096 0.238 < -0.37 , 0.562 >
## Alpha 22 0.568 0.149  < 0.276 , 0.86 >
##
##
## ------------------------
## Model selection criteria
## ------------------------
##
##          Loglik      AIC      BIC
## Value -2227.251 4468.501 4504.132
##
## -------
## Details
## -------
##
## Convergence reached? = TRUE
## Iterations = 4 / 200
## Processing time = 23.12733 secs
```

Main references

Arellano-Valle R.B., Bolfarine H., & Lachos V. (2005). Skew-normal linear mixed models. *J Data Sci 3*, 415–438.

Azzalini, A. & Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika 83(4)*.

Belenky, G., Wesensten, N.J., Thorne, D.R., Thomas, M.L., Sing, H.C., Redmond, D.P., Russo, M.B., & Balkin, T.J. (2003). Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study. *J Sleep Res* 12(1), 1–12.

Drikvandi R., Verbeke G., Molenberghs G. (2017). Diagnosing misspecification of the random-effects distribution in mixed models. *Biometrics 73(1)*.

Lachos, V.H., Ghosh, P., & Arellano-Valle, R.B. (2010). Likelihood based inference for skew–normal independent linear mixed models. *Stat Sin 20*.

Pinheiro, J.C., & Bates, D.M.(2000). *Mixed-Effects Models in S and S-PLUS*. Springer.

Pinheiro, J.C., Liu, C.H., & Wu, Y.N. (2001). Efficient algorithms for robust estimation in linear mixed-effects models using a multivariate t-distribution. *J Comput Graph Stat 10*.

Schumacher, F.L., Lachos, V.H., & Matos, L.A. (2021) Scale mixture of skew-normal linear mixed models with within-subject serial dependence. *Stat Med 40*.

github.com/fernandalschumacher/SINAPE2022-tutorial

Fernanda Lang Schumacher
schumacher.313@osu.edu

Larissa Avila Matos
larissam@unicamp.br

Victor Hugo Lachos
hlachos@uconn.edu