Robust estimation in linear mixed models using the R package *skewlmm*

Fernanda Lang Schumacher

joint work with Larissa A. Matos and Victor H. Lachos

Universidade Estadual de Campinas

23 de novembro de 2020

Introduction

- Linear mixed models are frequently used to analyze repeated measures data.
- Usual assumption: both random effect and error term follow normal distributions.
- Some proposals have been made in the literature for relaxing the assumption of normality.

Introduction

- Linear mixed models are frequently used to analyze repeated measures data.
- Usual assumption: both random effect and error term follow normal distributions.
- Some proposals have been made in the literature for relaxing the assumption of normality.
- Frequently, classes of LMMs consider that the error terms are conditionally independent.
- However, in longitudinal studies, repeated measures are collected over time and hence the error term tends to be serially correlated.

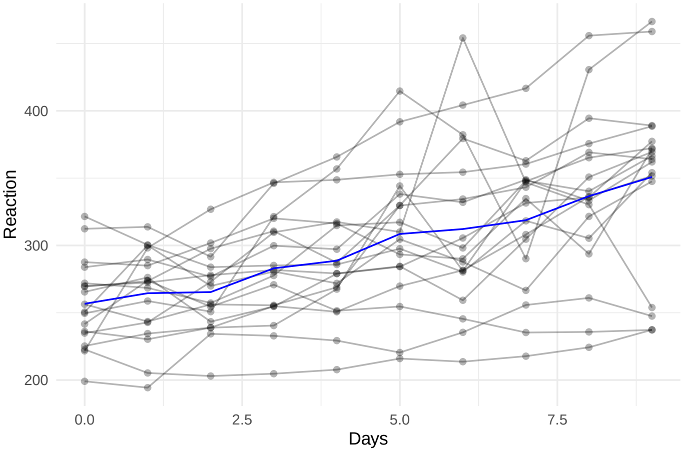
Motivation: sleepstudy data

- The average reaction time per day for subjects was evaluated by Gregory et al. (2003) in a sleep deprivation study.

- On day 0 the subjects had their normal amount of sleep and starting that night they were restricted to 3 hours of sleep per night for 9 days, and the reaction time basead on a series of tests was measured on each day for each subject.

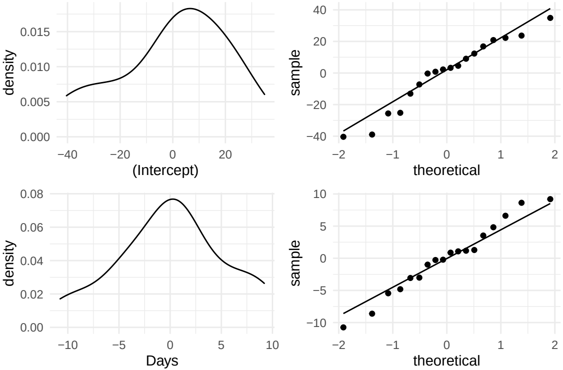
- The data are available at the R package *lme4*.

Motivation: sleepstudy data



Fitting a LMM to the sleepstudy dataset

```
library(nlme)
fitlme <- lme(Reaction~Days,data=sleepstudy,
              random=~Days|Subject)
```



Scale mixture of skew-normal (SMSN) distributions

The $\mathrm{SN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda})$ distribution (Azzalini and Valle, 1996) can be defined from:

$$f(\mathbf{y}) = 2\phi_p(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\Phi(A), \quad \mathbf{y} \in \mathbb{R}^p,$$

where $A = \boldsymbol{\lambda}^\top \boldsymbol{\Sigma}^{-1/2}(\mathbf{y} - \boldsymbol{\mu})$.

Scale mixture of skew-normal (SMSN) distributions

The $\text{SN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda})$ distribution (Azzalini and Valle, 1996) can be defined from:

$$f(\mathbf{y}) = 2\phi_p(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\Phi(A), \quad \mathbf{y} \in \mathbb{R}^p,$$

where $A = \boldsymbol{\lambda}^\top \boldsymbol{\Sigma}^{-1/2}\,(\mathbf{y} - \boldsymbol{\mu})$.

The $\text{SMSN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}; H)$ class of distributions can then be defined through the following pdf:

$$f(\mathbf{y}) = 2\int_0^\infty \phi_p(\mathbf{y}; \boldsymbol{\mu}, \kappa(u)\boldsymbol{\Sigma})\ \Phi(\kappa(u)^{-1/2}A)dH(u; \boldsymbol{\nu}),$$

$\mathbf{y} \in \mathbb{R}^p$, for some positive weight function $\kappa(u)$.

SMSN - special cases

- When $\boldsymbol{\lambda} = 0$, we get the $\text{SMN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}; H)$;
- When $\kappa(u) = u^{-1}$, we get the skew-normal/independent (SNI) class of distributions:
  - By taking $U \sim \text{Gamma}(\nu/2, \nu/2)$, the $\text{ST}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu)$ can be derived;
  - By taking $U \sim \text{Beta}(\nu, 1)$, the $\text{SSL}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu)$ 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 $\text{SCN}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}, \nu, \rho)$ can be derived, where $\nu_1, \nu_2 \in (0, 1)$.

SMSN linear mixed model (SMSN-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.

SMSN-LMM

Usual assumptions:

- $\mathbf{b}_i \overset{\text{iid}}{\sim} N_q(\mathbf{0}, \mathbf{D}) \perp \boldsymbol{\epsilon}_i \overset{\text{ind}}{\sim} N_{n_i}(\mathbf{0}, \boldsymbol{\Sigma}_i)$,
- The $q \times q$ random effects covariance matrix $\mathbf{D}$ may be unstructured or structured,
- the $n_i \times n_i$ error covariance matrix $\boldsymbol{\Sigma}_i$ is commonly written as $\sigma_e^2 \mathbf{R}_i$, where $\mathbf{R}_i$ can be a known matrix or a structured matrix depending on a vector of parameter, say $\boldsymbol{\phi}$.

SMSN-LMM

Likewise, the SMSN-LMM can be defined by considering

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

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

- $c = c(\boldsymbol{\nu}) = -\sqrt{\frac{2}{\pi}}k_1$, $k_1 = \mathrm{E}\{U^{-1/2}\}$, $\boldsymbol{\Delta} = \mathbf{D}^{1/2}\boldsymbol{\delta}$,
- $\mathbf{D} = \mathbf{D}(\boldsymbol{\alpha})$ depends on unknown and reduced parameter vector $\boldsymbol{\alpha}$, and
- $\boldsymbol{\Sigma}_i = \sigma_e^2\mathbf{R}_i$, with $\mathbf{R}_i = \mathbf{R}_i(\boldsymbol{\phi})$, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)^\top$, being a structured matrix.

Within-subject dependence structures

1. Conditional independence (CI): $\mathbf{R}_i = \mathbf{I}_{n_i}$.
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):

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

Important remark

The SMSN-LMM can be written hierarchically as follows:

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

which are all independent, and $\mathrm{TN}(\mu, \tau, (a, b))$ denotes the univariate normal distribution $(\mathrm{N}(\mu, \tau))$ truncated on the interval (a, b).

This representation is useful for the implementation of an EM-type algorithm, for details see Schumacher et al. (2020).

Tools for model evaluation

- Likelihood ratio (LR) test;
- Mahalanobis distance (known distribution);
- Healy-type plot;
- Empirical autocorrelation function (ACF) for standardized marginal residuals, which at lag l can be defined as

$$\widehat{\rho}(l) = \frac{\sum_{i=1}^{n} \sum_{\{(j,k)|t_k - t_j = l\}} r_{it_j} r_{it_k} / N(l)}{\sum_{i=1}^{n} \sum_{j=1}^{n_i} r_{it_j}^2 / N(0)},$$

where $\mathbf{r}_i = \widehat{\boldsymbol{\Upsilon}}_i^{-1/2} \left(\mathbf{y}_i - \mathbf{X}_i \widehat{\beta}\right)$ is the standardized marginal residual vector for subject i, with $\boldsymbol{\Upsilon}_i = \mathrm{Var}(\mathbf{Y}_i)$, and $N(\cdot)$ is the number of pairs used in the respective numerator summation.

The R package *skewlmm*

- The package *skewlmm* implemets an EM-type algorithm in R using S3 class, containing methods for estimating and predicting the SM(S)N-LMM.

- It has an user-friendly interface with generic R functions `print`, `summary`, `plot`, `fitted`, `residuals` and `predict` implemented.

- The main functions in the package are `smsn.lmm()` and `smn.lmm()`, which fit a SMSN-LMM and a SMN-LMM, respectively.

The R package *skewlmm*

The basic syntax of these functions is as follows:

```
smsn.lmm(data, formFixed, groupVar, formRandom,
         depStruct, distr, ...)
smn.lmm(data, formFixed, groupVar, formRandom,
        depStruct, distr, ...)
```

where

- `data`: A data frame containing all the variables to be used in the model.
- `formFixed`: A two-sided linear formula object describing the fixed effects part of the model.
- `groupVar`: A character containing the name of the variable which represents the subjects or groups in data.

- `formRandom`: A one-sided linear formula object describing the random effects part of the model.
- `depStruct`: A character indicating which dependence structure should be used.
- `distr`: A character indicating which distribution should be used.

- `formRandom`: A one-sided linear formula object describing the random effects part of the model.
- `depStruct`: A character indicating which dependence structure should be used.
- `distr`: A character indicating which distribution should be used.

Some other useful arguments:

- `timeVar`: A character containing the name of the variable which represents the time in data.
- `pAR`: The order of the autoregressive process that should be used (if depStruct="ARp").
- `initialValues`: A named list containing initial parameter values.

The functions return an object of the class `SMSN` and `SMN`, respectively, containing a list of elements, and the following methods/ functions are available to these classes:

- `print`
- `summary`
- `fitted`
- `plot`
- `predict`
- `residuals`
- `ranef`
- `acfresid`
- `healy.plot`
- `lr.test`
- `mahalDist`

Example: sleepstudy data

```
#fitting a (CI)-SMN-LMM, default is distr='norm'
fit0<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
              formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
#fitting a (CI)-SMSN-LMM, default is distr='sn'
fitskew0<-smsn.lmm(data=sleepstudy,formFixed = Reaction~Days,
              formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
fit0
```

```
## Linear mixed models with distribution norm and dependency structure CI
## Call:
## smn.lmm(data = sleepstudy, formFixed = Reaction ~ Days, groupVar = "Subject",
##     formRandom = ~Days, quiet = T)
##
## Fixed:Reaction ~ Days
## Random:~Days
##   Estimated variance (D):
##             (Intercept)     Days
## (Intercept)   560.68002 11.91827
## Days           11.91827 32.52525
##
## Estimated parameters:
##      (Intercept)    Days   sigma2  Dsqrt1 Dsqrt2 Dsqrt3
##         251.4051 10.4673 655.4400 23.6752 0.4059 5.6886
## s.e.      7.2257  1.5541  31.3784 10.0856 1.7514 1.4270
##
## Model selection criteria:
##   logLik     AIC      BIC
##  -875.97 1763.94 1783.098
##
## Number of observations: 180
## Number of groups: 18
```

```
fitskew0
```

```
## Linear mixed models with distribution sn and dependency structure CI
## Call:
## smsn.lmm(data = sleepstudy, formFixed = Reaction ~ Days, groupVar = "Subject",
##     formRandom = ~Days, quiet = T)
##
## Fixed:Reaction ~ Days
## Random:~Days
##   Estimated variance (D):
##             (Intercept)     Days
## (Intercept)  1432.65701 35.22718
## Days           35.22718 33.76855
##
## Estimated parameters:
##      (Intercept)    Days   sigma2  Dsqrt1 Dsqrt2 Dsqrt3 lambda1 lambda2
##         251.4073 10.4657 652.6598 37.8418 0.8080 5.7546 -4.2917 -0.2477
## s.e.     11.6328  2.1610  32.7878 28.9642 5.2063 1.6311      NA      NA
##
## Model selection criteria:
##    logLik      AIC      BIC
##  -875.354 1766.709 1792.253
##
## Number of observations: 180
## Number of groups: 18
```

Changing the distribution

```
fit1<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’t’,
              formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
fitskew1<-smsn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’st’,
                   formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
fit2<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’sl’,
              formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
fitskew2<-smsn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’ssl’,
                   formRandom = ~Days,groupVar = "Subject",quiet = T)
```

```
fit3<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’cn’,
              formRandom = ~Days,groupVar = "Subject",quiet = T)
```

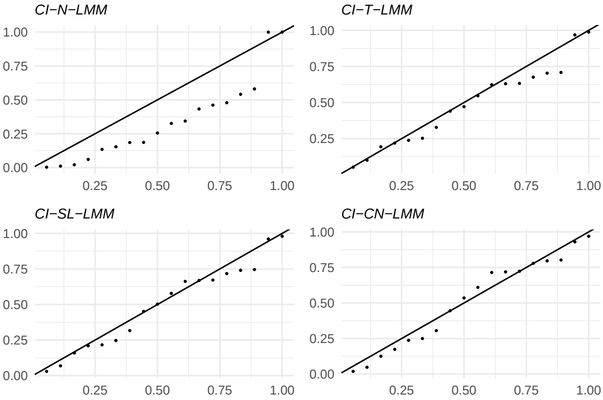
```
fitskew3<-smsn.lmm(data=sleepstudy,formFixed = Reaction~Days,distr = ’scn’,
                   formRandom = ~Days,groupVar = "Subject",quiet = T)
```

Comparing the fitted models

distr	AIC	BIC
norm	1763.9	1783.1
sn	1766.7	1792.3
t	1737.5	1759.8
st	1739.8	1768.6
sl	1736.2	**1758.5**
ssl	1738.4	1767.2
cn	**1733.7**	1759.3
scn	1735.9	1767.8

Assessing the goodness of fit using a Healy-type plot

```
grid.arrange(healy.plot(fit0),healy.plot(fit1),
             healy.plot(fit2),healy.plot(fit3))
```



LR test for $H_0 : \lambda = 0$ in the CN-LMM

```r
lr.test(fitskew3,fit3)
```

```
## Model selection criteria:
##             logLik      AIC      BIC
## fitskew3 -857.926 1735.851 1767.781
## fit3     -858.861 1733.722 1759.266
##
##     Likelihood-ratio Test
##
## chi-square statistics =  1.871248
## df =  2
## p-value =  0.392341
##
## The null hypothesis that both models represent the
## data equally well is not rejected at level  0.05
```

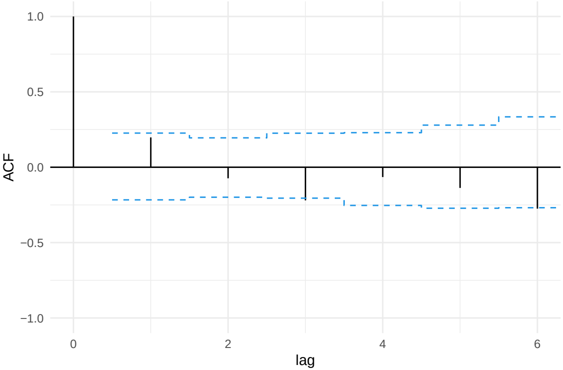
Computing the ACF of the residuals from CN-LMM

```
acfresid(fit3)
```

```
##     lag         ACF n.used
## 1     0  1.00000000    180
## 2     1  0.19793878    162
## 3     2 -0.07329748    144
## 4     3 -0.21972223    126
## 5     4 -0.06519806    108
## 6     5 -0.13723727     90
## 7     6 -0.27485055     72
## 8     7 -0.08778128     54
## 9     8  0.19912222     36
## 10    9  0.58707921     18
```

Plotting the ACF (CI-CN-LMM)

```
plot(acfresid(fit3,calcCI = T,maxLag = 6))
```



Fitting an AR(p)-SMN-LMM

```
#sl
fit2ar1<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="sl",depStruct = "ARp",pAR=1,quiet=T)
```

```
fit2ar2<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="sl",depStruct = "ARp",pAR=2,quiet=T)
```

```
#cn
fit3ar1<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="cn",depStruct = "ARp",pAR=1,quiet=T)
```

```
fit3ar2<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="cn",depStruct = "ARp",pAR=2,quiet=T)
```

Fitting a DEC-SMN-LMM

```
#sl
fit2dec<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="sl",depStruct = "DEC",quiet=T,
                 timeVar = "Days")
```

```
#cn
fit3dec<-smn.lmm(data=sleepstudy,formFixed = Reaction~Days,
                 formRandom = ~Days,groupVar = "Subject",
                 distr="cn",depStruct = "DEC",quiet=T,
                 timeVar = "Days")
```

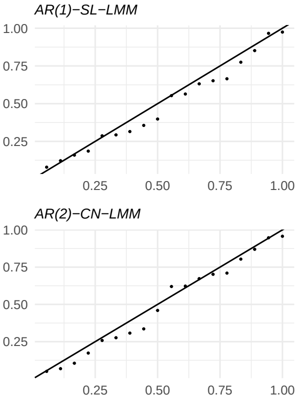
Since the data are equally spaced and sorted by time, the use of `timeVar` in here is optional (the function will use the position if `timeVar` is not provided).

Comparing the fitted models

distr	depStruct	AIC	BIC
sl	CI	1736.2	1758.5
sl	AR1	1716.9	**1742.5**
sl	AR2	1717.3	1746.0
sl	DEC	1718.2	1746.9
cn	CI	1733.7	1759.3
cn	AR1	1715.5	1744.3
cn	AR2	**1714.9**	1746.9
cn	DEC	1716.3	1748.2

Healy plot for the AR(p)-SMN-LMM

```
grid.arrange(healy.plot(fit2ar1),healy.plot(fit3ar1),
             healy.plot(fit3ar2),ncol=2)
```



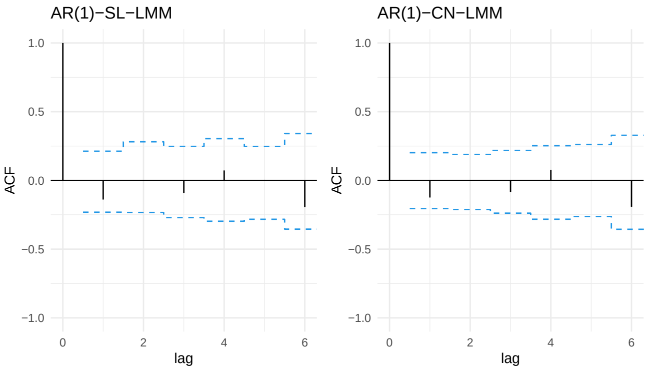
LR test for $H_0 : \phi_2 = 0$ in the CN-LMM

```
lr.test(fit3ar1,fit3ar2)
```

```
##
## Model selection criteria:
##           logLik      AIC      BIC
## fit3ar1 -848.760 1715.520 1744.257
## fit3ar2 -847.463 1714.925 1746.855
##
##     Likelihood-ratio Test
##
## chi-square statistics =  2.594949
## df =  1
## p-value =  0.1072049
##
## The null hypothesis that both models represent the
## data equally well is not rejected at level  0.05
```

ACF plot for AR(1)-SMN-LMM

```
grid.arrange(plot(acfresid(fit2ar1,calcCI = T,maxLag = 6))+
               ggtitle("AR(1)-SL-LMM"),
             plot(acfresid(fit3ar1,calcCI = T,maxLag = 6))+
               ggtitle("AR(1)-CN-LMM"), ncol=2)
```



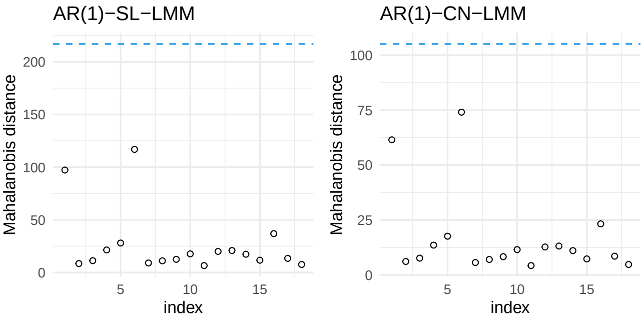
Fitted models

```
cbind(fit2ar1$theta,fit2ar1$std.error)
cbind(fit3ar1$theta,fit3ar1$std.error)
```

	SL		CN	
	estimate	std.error	estimate	std.error
(Intercept)	251.55	7.56	252.55	7.02
Days	9.81	1.66	9.99	1.57
sigma2	266.24	63.43	417.84	78.77
phiAR1	0.57	0.14	0.56	0.14
Dsqrt1	14.21	10.15	16.90	11.82
Dsqrt2	2.58	4.16	2.82	4.76
Dsqrt3	2.17	6.45	2.89	6.31
nu1	1.52	-	0.12	-
nu2	-	-	0.16	-

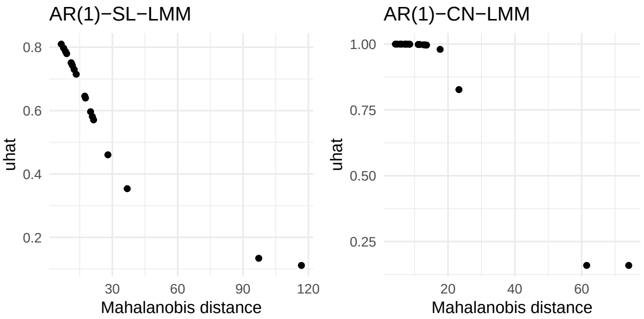
Mahalanobis distance for AR(1)-SMN-LMM

```
grid.arrange(plot(mahalDist(fit2ar1),fitobject = fit2ar1,nlabels = 0)+
               ggtitle(’AR(1)-SL-LMM’),
             plot(mahalDist(fit3ar1),fitobject = fit3ar1,nlabels = 0)+
               ggtitle(’AR(1)-CN-LMM’), ncol=2)
```



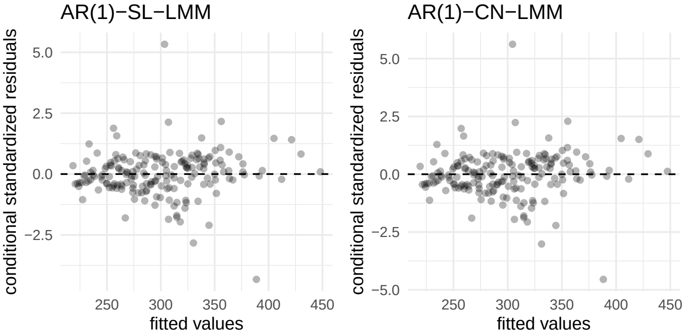
Mahalanobis distance versus $\hat{u}$ for AR(1)-SMN-LMM

```
grid.arrange(qplot(mahalDist(fit2ar1),fit2ar1$uhat)+theme_minimal()+
               ylab("uhat")+xlab("Mahalanobis distance")+
               ggtitle(’AR(1)-SL-LMM’),
             qplot(mahalDist(fit3ar1),fit3ar1$uhat)+theme_minimal()+
               ylab("uhat")+xlab("Mahalanobis distance")+
               ggtitle(’AR(1)-CN-LMM’), ncol=2)
```



Plotting fitted models

```
grid.arrange(plot(fit2ar1,type = "normalized")+ggtitle('AR(1)-SL-LMM'),
             plot(fit3ar1,type = "normalized")+ggtitle('AR(1)-CN-LMM'),
             ncol=2)
```



Prediction of future measurements

```r
tail(sleepstudy,n=3)
```

```
##     Reaction Days Subject
## 178 343.2199    7     372
## 179 369.1417    8     372
## 180 364.1236    9     372
```

```r
predData <- data.frame(Reaction=NA,Days=10:11,Subject="372")
predict(fit2ar1,newData = predData)
```

```
##   groupVar Days    ypred
## 1      372   10 377.0998
## 2      372   11 389.5745
```

```r
predict(fit3ar1,newData = predData)
```

```
##   groupVar Days    ypred
## 1      372   10 377.1056
## 2      372   11 389.5576
```

Concluding remarks

Some additional features are currently under development:

- Estimation of some important special cases, such as when $\mathbf{D}$ is diagonal or in blocks;
- Use of parallel optimization to improve performance;
- Use of a method for acceleration of the convergence rate of the EM-type algorithm used in the estimation procedure.

Main references

[1] Azzalini, A. & A. D. Valle (1996).
The multivariate skew-normal distribution.
Biometrika 83(4), 715–726.

[2] 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.
Journal of Sleep Research 12, 1–12.

[3] Box, G. E., G. M. Jenkins, G. C. Reinsel & G. M. Ljung (2015).
Time series analysis: forecasting and control.
John Wiley & Sons.

[4] Dempster, A., Laird, N. & Rubin, D. (1977).
Maximum likelihood from incomplete data via the EM algorithm.
Journal of the Royal Statistical Society, Series B 39, 1–38.

[5] Lachos, V. H., P. Ghosh & R. B. Arellano-Valle (2010).
Likelihood based inference for skew–normal independent linear mixed models.
Statistica Sinica 20, 303–322.

[6] Muñoz, A., V. Carey, J. P. Schouten, M. Segal & B. Rosner (1992).
A parametric family of correlation structures for the analysis of longitudinal data.
Biometrics 48, 733–742.

[7] Schumacher, F. L., Matos, L. A., & Lachos, V. H. (2020).
Scale mixture of skew-normal linear mixed models with within-subject serial dependence.
arXiv preprint arXiv:2002.01040.

Preprint

GitHub

Acknowledgments

CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico

CAPES

Fernanda Lang Schumacher
PhD student, IMECC – UNICAMP
fernandalschumacher@gmail.com