

---

# Implementing Mixed Models with Repeated Measures (MMRM) in R

*Daniel Sabanés Bové*



# Acknowledgements

*Thanks to many in Product Development Data Sciences at Roche*

## Statistical Programmers

Francois Collin  
Jennifer Li  
Nick Paszty  
Jana Stoilova

## Core Developers

Nikolas Burkoff  
Reinhold Koch  
Ondrej Slama  
Pawel Rucki

## Methods

Gian Thanei  
Marcel Wolbers  
Godwin Yung

## Technical Lead NEST

Adrian Waddell

# Transformation to an R based workflow is ongoing

2019

2020

2021

2022

**Regulatory SAS platform**

**Exploratory SAS platform**

**Regulatory R platform**

*As the usage of SAS diminishes also the need for these platforms will fade out*

*Validated transition solution*



*Illustration of overall R, Python, ... usage*



*Illustration of overall SAS usage*

**DevOps Approach**  
Professional Statistical Software Engineering through Collaboration and Automation

**R-packages (NEST platform)**

**Other (e.g. Python, SAS, ..)**



# Mixed Models with Repeated Measures (MMRM)

## *Theory and Application in One Slide*

### Theory

MMRM = Special case of a Linear Mixed Model (LMM):

$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\varepsilon}_i$  for patient  $i$ , where

- $\boldsymbol{\beta}$  are the fixed effects, including visit-specific treatment effects
- $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$  are the random effects,  $\boldsymbol{\varepsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$  are the residuals
- Resulting marginal covariance matrix is  $\mathbf{V}_i = \mathbf{Z}_i\mathbf{D}\mathbf{Z}_i^\top + \boldsymbol{\Sigma}$
- Marginal model is  $\mathbf{Y}_i \sim N(\mathbf{X}_i\boldsymbol{\beta}, \mathbf{V}_i)$
- Restricted Maximum Likelihood Estimation of parameters

### Application

Primary analysis of continuous endpoints in longitudinal clinical trials

- The time variable is categorical with a small number of “visits”
- Other covariates can be age, gender, baseline value of outcome, etc.
- Correlation between outcomes of the same patient is accounted for
- Typically want the marginal covariance matrix to be “unstructured”

# Challenge: How to implement MMRM in R to match SAS?

## *MMRM as one example where the transformation is non-trivial*

**SAS:** PROC MIXED as gold standard traditional proprietary software. Has many options for MMRM.

But: We want to move to R also for MMRM analysis.

**R-package nlme:** Classic mixed effects package with some transition guidance. Has options for covariance matrix specification.

But: Does not give Satterthwaite adjusted degrees of freedom and p-values.

**R-packages lme4 and lmerTest:** Modern mixed effects packages that are actively being developed and give Satterthwaite adjustment.

But: Seemed like covariance matrix cannot be unstructured.

### lmer for SAS PROC MIXED Users

Douglas Bates  
Department of Statistics  
University of Wisconsin – Madison  
Bates@wisc.edu

#### 1 Introduction

The `lmer` function from the `lme4` package for R is used to fit linear mixed-effects models. It is similar in scope to the SAS procedure PROC MIXED described in Littell et al. (1996).

A file on the SAS Institute web site (<http://www.sas.com>) contains all the data sets in the book and all the SAS programs used in Littell et al. (1996). We have converted the data sets from the tabular representation used for SAS to the data frames objects used for R. To help those familiar with SAS

### Fitting Linear Mixed-Effects Models Using lme4

Douglas Bates Martin Michler Benjamin M. Bolker Steven C. Walker  
University of Wisconsin-Madison ETH Zurich McMaster University McMaster University

#### Abstract

Maximum likelihood or restricted maximum likelihood (REML) estimates of the parameters in linear mixed-effects models can be determined using the lmer function in the lme4 package for R. As for most model-fitting functions in R, the model is described in an R formula and is fit by a function, in this case including both fixed- and random-effects terms. The formula and data together determine a statistical representation of the model from which the profile deviance or the profile REML criterion can be evaluated as a function of one or more model parameters. The appropriate criterion is optimized, using one of the constrained optimization functions in R, to provide the parameter estimates. We describe the structure of the model, the steps in evaluating the profile deviance or REML criterion, and the structure of classes or types that represent such a model. Sufficient detail is included to allow specialization of these structures by users who wish to write

# Step 1: Fit model with unstructured covariance with `lme4`

**Problem** It seemed that `lme4` does not support unstructured covariance matrices.

**Solution**

- Look in detail whether this assumption is really true.
- Accept that we need to use random effects to model the covariance.
- However, if we nest the visit within the subject ID variable and disable some check, we can get it to work:

```
lmer(y ~ treatment + visit + (0 + visit | id), ...,  
      control = lmerControl(check.nobs.vs.nRE = "ignore"))
```

(thanks to [Brice Ozenne](#) for his LMM vignette!)

- Not perfect, as we have one variance parameter too much here. But the covariance matrix as a whole is still identifiable. Just don't look at the "sigma" estimate separately.

## Step 2: Multiple optimization algorithms improve convergence

**Problem** Just using the default optimizer often failed convergence when fitting the LMM.

**Solution**

- First try default optimizer and see if it leads to convergence.
- If not, then try 6 additional optimizers:
  - If multiple of them work, then take the result which gives highest restricted likelihood.
  - If none of them work, only then fail.
- We can run these in parallel (multi-core) easily (at least on Unix):
  - Load percentage check can give us number of “free cores” on our RStudio server instance - so we can just say “parallel = TRUE”

## Step 3: Calculate covariance estimate and model diagnostics

### Problem

1. The default AIC, BIC, etc. in lme4 do not match SAS equivalents.
2. We need the marginal covariance matrix estimate for residual plots.

### Solution

- Search mailing lists etc. to get starting point for covariance matrix calculations from fit internals
  - Take a patient with maximum number of visits
  - Helper function `lme4::getME()` allows access to fit internals
- Implement custom AIC, BIC, etc.
  - Needs effective number of variance parameters (from covariance)
- Lots of comparisons with SAS documentation and results
- Unit tests that make sure our results keep matching gold standard results



# Results: Static production of MMRM tables and plots

## *Leveraging our own rtables for tables and ggplot2 for plots*

```
library(dplyr)
library(tern)
library(random.cdisc.data)

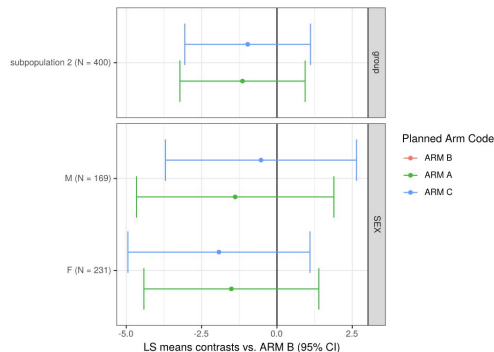
adsl <- radsl(cached = TRUE)
adqs <- radqs(cached = TRUE)
adqs_f <- adqs %>% ...

mmrm_results <- s_mmrn(
  vars = list(
    response = "AVAL",
    covariates = c("STRATA1", "BMRKR2"),
    id = "USUBJID",
    arm = "ARM",
    visit = "AVISIT"
  ),
  data = adqs_f,
  cor_struct = "unstructured",
  weights_emmeans = "proportional",
  optimizer = "automatic"
)
```

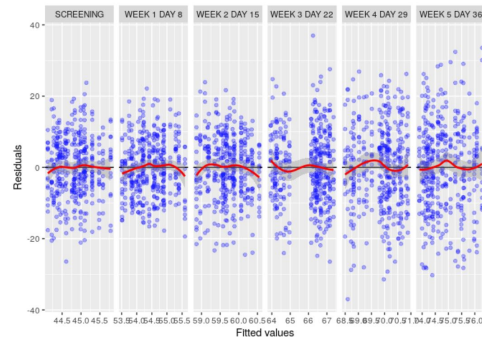
*Code usage example*

B: Placebo (N=134)	A: Drug X (N=134)
134	134
45.190 (0.738)	44.578 (0.737)
(43.739, 46.64)	(43.129, 46.028)
	-0.611 (1.044)
	(-2.664, 1.441)
	-1.4%
	0.5583
134	134
54.306 (0.689)	55.032 (0.688)
(52.952, 55.661)	(53.679, 56.385)
	0.725 (0.975)
	(-1.191, 2.642)
	1.3%
	0.4572

*Least square means table  
(only one part shown)*



*Forest plot for subgroups*



*Residuals vs. fitted values*

*~1900 LoC*

*4 Tables*

*5 Plots*

*2 Months*

**700 for model fit etc.**  
**900 for tables**  
**300 for plots**  
**(+similar for unit and regression tests)**

1. Least square means
2. Fixed effect estimates
3. Covariance matrix
4. Model diagnostic stats

1. Least square means
2. Contrasts of LS means
3. Q-Q residuals
4. Residuals vs. fitted values
5. Forest plot for subgroups

First discussions to release  
 (part of our internal NEST  
 R-package "tern" for statistical  
 analyses)

# Results: Interactive exploration via Shiny framework module

## *Via our teal framework - increases efficiency during study readout*

▼ Model Settings

Analysis data: **ADQS**

Select Response

AVAL

Select Parameter

FKSI-FWB Function/Well-Being (GF1,G)

Arm Variable

ARM Description of Planned Arm

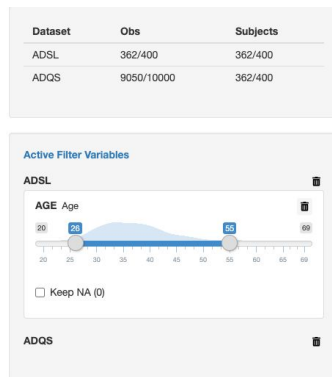
Visit Variable

AVISIT Analysis Visit

Covariate Variables

- Nothing selected -

*Model specification*



*Filtering panel*

Subject Identifier

USUBJID

Weights for LS means

proportional

Correlation Structure

unstructured

Confidence Level

0.95

Optimization Algorithm

automatic

Parallel Computing

Optimizer used: bobyqa

Fit Model

*Fit controls*



```
fit <- s_mmr(
  vars = list(
    response = "AVAL",
    covariates = NULL,
    id = "USUBJID",
    arm = "ARM",
    visit = "AVISIT"
  ),
  data = ANL,
  conf_level = 0.95,
  cor_struct = "unstructured",
  weights_emeans = "proportional",
  optimizer = "automatic",
  parallel = TRUE
)
fit[["col_N"]] <- table(ADSL_P[["ARM"]])
tbl <- t_mmr_lsmeans(
  fit,
  col_N = fit$col_N,
  table_tree = FALSE,
  show_relative = "reduction"
)
print(tbl)
```

Copy to Clipboard Dismiss

*Static R code production*

## *Encodings*

3 Encodings on left-hand side of the module:

1. Model specification
2. Output type
3. Output settings

## *Filtering*

Standard data set filtering on right-hand side of the module.

## *Controls*

- Parallel computing enabled (~50 cores RStudio server).
- Delayed reactivity via “Fit Model” button, as each fit takes at least 5 seconds.

## *Reproducible*

“Show R Code” button produces the R code which can reproduce currently shown output in a static R session.

# Outlook

## *Next steps for leveraging the full value of our MMRM implementation*



**Validation:** As part of next validation cycle, the MMRM implementation will be installed on our regulatory R platform, too.

Opens the door to submissions including the MMRM results from R.



**Further convergence improvements:** Discussing e.g. with Ben Bolker from lme4 team on best practice for handling row ordering, convergence warnings, etc. ([issue](#))

This will increase the robustness and user experience further.



**Methods Extensions:** Additional covariance structures (that are between unstructured and compound symmetry) would be good as fallback solution. Random slope analysis module could leverage backbone pieces.

Will take more time but cover more use cases.

THANK YOU!

## References

*Please reach out to [daniel.sabanes\\_bove@roche.com](mailto:daniel.sabanes_bove@roche.com) for questions!*

- NEST**
- UseR2020 on rtables:  
<https://www.youtube.com/watch?v=CBQzZ8ZhXLA> by G. Becker
  - Phuse 2019 and R in Pharma 2018: “Analysing Clinical Trials Data with R” by A. Waddell  
<https://rinpharma.github.io/website2018/program/analyzing-clinical-trials-data-with-r.html>
- MMRM**
- Mallinckrod, C. H., Lane, P. W., Schnell, D., Peng, Y., & Mancuso, J. P. (2008). Recommendations for the Primary Analysis of Continuous Endpoints in Longitudinal Clinical Trials. Drug Information Journal, 42(4), 303–319.  
<https://doi.org/10.1177/009286150804200402>

*Doing now what patients need next*