

A6. Linear Mixed Models

Fixed and random effects, REML, and practical inference

Dr. Paul Schmidt

To install and load all the packages used in this chapter, run the following code:

```
for (pkg in c("lme4", "lmerTest", "nlme", "glmmTMB", "emmeans",
             "broom.mixed", "agridat", "tidyverse")) {
  if (!require(pkg, character.only = TRUE)) install.packages(pkg)
}

library(lme4)
library(lmerTest)
library(nlme)
library(glmmTMB)
library(emmeans)
library(broom.mixed)
library(agridat)
library(tidyverse)
```

Linear mixed models (LMMs) extend the ordinary linear model by introducing a second type of effect: random effects. They are indispensable whenever the data contain grouped, clustered, nested, or otherwise non-independent observations - which covers the vast majority of designed experiments in agriculture and the life sciences. This chapter explains the underlying concepts and shows how to fit mixed models in R with `lme4`, `lmerTest`, and related packages. Several of the design chapters in this section rely on these ideas: incomplete block designs such as the Alpha Design and the Augmented Design both fit random block effects, and the Row-Column Design uses random row and column effects.

The Quick Version

You are fitting an ANOVA or regression and something in the experimental design tells you that not all observations are truly independent. Plants come from the same block, measurements come from the same animal, plots share a row in the field. Whenever a factor describes such a grouping structure and you are not primarily interested in the specific levels that happen to appear in your data, that factor is a natural candidate for a random effect. The model is then called a linear mixed model because it mixes fixed and random effects.

The most widely used R package for this is `lme4`. The syntax looks almost like `lm()`, with an extra term in parentheses that describes the random effect:

```
mod <- lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy)
summary(mod)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: Reaction ~ Days + (1 | Subject)
Data: sleepstudy

REML criterion at convergence: 1786.5

Scaled residuals:
   Min       1Q   Median       3Q      Max
-3.2257 -0.5529  0.0109  0.5188  4.2506
```

```

Random effects:
Groups   Name             Variance Std.Dev.
Subject  (Intercept) 1378.2   37.12
Residual                    960.5   30.99
Number of obs: 180, groups: Subject, 18

Fixed effects:
              Estimate Std. Error      df t value Pr(>|t|)
(Intercept) 251.4051      9.7467  22.8102  25.79 <2e-16 ***
Days         10.4673      0.8042 161.0000  13.02 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr)
Days -0.371

```

We are using the built-in `sleepstudy` dataset from `lme4`, which contains reaction times for 18 subjects over 10 consecutive days of sleep deprivation. The fixed part `Reaction ~ Days` describes the average effect of sleep deprivation on reaction time. The term `(1 | Subject)` adds a random intercept per subject: each of the 18 subjects is allowed to have their own baseline reaction time, and these baselines are modelled as draws from a normal distribution.

Quick Decision Rule

If a factor describes the randomization structure of the experiment (blocks, plots, animals, measurement occasions), treat it as random. If a factor describes a treatment whose specific levels you want to compare or name in your report, treat it as fixed.

What Is a Random Effect?

In every linear model we have seen so far, the error term `e` is itself a random variable. Strictly speaking, almost every model therefore contains a stochastic component. What distinguishes a random effect from the error is that the random effect structures the variation between groups of observations, not between individual observations.

A useful mental model is to ask where the levels of a factor come from. For a fixed factor, the levels in the data are exactly the levels you care about. If the treatment factor has three varieties A, B, C, then A, B, C are the entire story. For a random factor, the levels in the data are a sample from a larger population of possible levels. The 18 subjects in `sleepstudy` are not 18 specifically chosen humans whose reaction times we care about - they stand in for the population of potential subjects. Blocks in a field trial are another classic example: the experiment could just as well have used a different arrangement of blocks, and the inference is supposed to generalize beyond the particular blocks that happened to be used.

A common question is how many levels a factor needs before it can reasonably be treated as random. A rule of thumb in the literature places the minimum somewhere between 5 and 12 levels, because the variance component that describes the spread of random effects is poorly estimated with very few levels. This rule is a guideline, not a hard threshold. Two qualifications matter in practice:

- Factors that describe the design structure (blocks, plots, subjects) are almost always treated as random regardless of how many levels they have, because the scientific interpretation requires it. Fitting an RCBD with only three blocks as random is standard practice.
- The 5 to 12 range is a convention, not a theorem. What actually matters is the stability of the variance estimate, which depends on sample size, balance, and the size of the variance component itself.

H. P. Piepho, A. Büchse, and K. Emrich [1] give a detailed discussion of fixed versus random decisions in the context of plant breeding, where the same factor (genotype) can reasonably be modelled either way depending on the scientific question.

General Representation

A classical linear model can be written compactly as

$$y = X\beta + e$$

where y is the vector of observations, X is the design matrix of the fixed effects, β is the vector of fixed-effect coefficients, and e is the vector of residuals. A linear mixed model adds one more term:

$$y = X\beta + Zu + e$$

Here Z is the design matrix for the random effects and u is the vector of random effects. The distributional assumptions are that u comes from a multivariate normal distribution with mean zero and some variance-covariance matrix G , and that e is multivariate normal with mean zero and variance-covariance matrix R . Under this setup, the observations y are themselves multivariate normal with mean $X\beta$ and a combined variance $V = ZGZ' + R$ that captures both the random-effects variation and the residual variation.

The important consequence is that observations sharing a random-effect level are correlated. Two reaction times from the same `sleepstudy` subject are more similar than two reaction times from different subjects. The mixed model accounts for this correlation automatically, rather than pretending it does not exist.

A small numerical example

A tiny dataset helps to see what the design matrices actually look like. Consider three varieties with two observations each:

```
toy <- data.frame(
  var = factor(c("A", "A", "B", "B", "C", "C")),
  block = factor(c(1, 2, 1, 2, 1, 2)),
  yield = c(3.2, 3.6, 2.8, 2.9, 4.1, 4.0)
)
toy
```

```
  var block yield
1  A     1   3.2
2  A     2   3.6
3  B     1   2.8
4  B     2   2.9
```

```
5 C 1 4.1
6 C 2 4.0
```

A model with variety as fixed and block as random can be fitted as follows:

```
mod_toy <- lmer(yield ~ var + (1 | block), data = toy)
```

```
boundary (singular) fit: see help('isSingular')
```

```
summary(mod_toy)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: yield ~ var + (1 | block)
Data: toy

REML criterion at convergence: 0.1

Scaled residuals:
   Min       1Q   Median       3Q      Max
-1.1547 -0.2887  0.0000  0.2887  1.1547

Random effects:
 Groups   Name                Variance Std.Dev.
 block    (Intercept)          0.00      0.0000
 Residual                            0.03      0.1732
Number of obs: 6, groups: block, 2

Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)   3.4000     0.1225   3.0000  27.761 0.000103 ***
varB          -0.5500     0.1732   3.0000  -3.175 0.050270 .
varC           0.6500     0.1732   3.0000   3.753 0.033053 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr) varB
varB -0.707
varC -0.707  0.500
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

The fixed-effects design matrix X has six rows (one per observation) and three columns (intercept plus two variety contrasts). The random-effects design matrix Z also has six rows, but two columns - one per block - and consists of zeros and ones indicating which observation belongs to which block. The output of `summary()` reports the fixed-effect estimates, the estimated variance of the block effects, and the residual variance.

Fixed or Random: How to Decide

The decision between fixed and random is occasionally philosophical, but in most applied work a handful of guiding questions settle it quickly.

Treat a factor as **random** when at least one of the following holds:

- The factor represents a sample from a larger population that you want to generalize to. Selecting a handful of locations from a target region, or a panel of subjects from a population of potential subjects, are textbook cases.

- The factor represents a randomization unit in the experimental design. Blocks in an RCBD, main plots in a split-plot design, and incomplete blocks in an alpha design all fall under this rule.
- The factor represents sub-samples of an experimental unit: multiple plants per plot, multiple leaves per plant, repeated measures on the same animal.
- The factor is crossed with another random factor. If `Location` is random, then the interaction `Year:Location` inherits the random label.
- You want to impose a specific covariance structure on the levels, for example genetic relatedness between genotypes or spatial proximity between plots.

Treat a factor as **fixed** when the specific levels are the scientific target of the analysis. Treatment levels in a controlled experiment almost always qualify: if the point of the study is to compare three fertilizers, those three fertilizers are fixed even if they were drawn from a catalogue of dozens of products.

Packages for Mixed Models in R

Three packages cover the overwhelming majority of mixed-model work in R.

The `lme4` package [2] is the de facto standard for linear and generalized linear mixed models with independent residuals and simple random-effect structures. Its main function is `lmer()` for Gaussian responses and `glmer()` for non-Gaussian responses. It is fast, battle-tested, and well documented.

The `nlme` package [3] is older but still widely used. Its function `lme()` supports models with correlated residuals and heterogeneous variances out of the box (via the `correlation =` and `weights =` arguments), which `lme4` does not. If you need AR(1) residuals for repeated measures, compound symmetry, or variance functions like `varIdent` or `varPower`, `nlme::lme()` is often the most direct route.

The `glmmTMB` package combines much of the syntax of `lme4` with the flexibility of `nlme`. It handles a wide range of response distributions (zero-inflated, negative binomial, beta, tweedie), supports correlated random effects and residuals, and scales well to larger datasets. It is the natural choice when `lme4` is too restrictive but you want to keep the `lme4`-style formula syntax.

For the purposes of this chapter we focus on `lme4` because its formula syntax has become the lingua franca of mixed-model specification in R.

Random-effect syntax in `lme4`

The term in parentheses on the right-hand side of the formula specifies the random effects. A few common patterns:

- `(1 | group)` - random intercept per group. Each level of `group` gets its own baseline, drawn from a normal distribution.
- `(1 | group1) + (1 | group2)` - two separate (crossed or nested) random intercepts. Subject and Item in psycholinguistic experiments are the canonical case.

- `(1 | group1/group2) - group2` nested in `group1`. Equivalent to `(1 | group1) + (1 | group1:group2)`.
- `(slope | group)` - random intercept and random slope of `slope` per level of `group`, with an estimated correlation between them.
- `(0 + slope | group)` - random slope without random intercept.

The `sleepstudy` data are a good testbed for random slopes. Subjects likely differ not only in baseline reaction time but also in how strongly sleep deprivation affects them:

```
mod_slope <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)
summary(mod_slope)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: Reaction ~ Days + (Days | Subject)
Data: sleepstudy

REML criterion at convergence: 1743.6

Scaled residuals:
   Min       1Q   Median       3Q      Max
-3.9536 -0.4634  0.0231  0.4634  5.1793

Random effects:
 Groups   Name                Variance Std.Dev. Corr
 Subject (Intercept)    612.10    24.741
           Days              35.07     5.922  0.07
 Residual                    654.94    25.592
Number of obs: 180, groups: Subject, 18

Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)  251.405      6.825  17.000  36.838 < 2e-16 ***
Days         10.467      1.546  17.000   6.771 3.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
 (Intr)
Days -0.138
```

The random-effects section of the output now shows a variance for the intercept, a variance for the `Days` slope, and a correlation between them. The fixed-effect estimate of `Days` is the average slope across subjects.

ML and REML

Ordinary linear models estimate their parameters by least squares. Mixed models cannot: the presence of random effects means that the variance components themselves need to be estimated from the data, and least squares has nothing to say about them. The standard approach is **maximum likelihood (ML)**, which chooses parameter values that make the observed data as likely as possible under the assumed distribution. For a purely fixed-effects Gaussian model, ML and least squares give identical estimates for `beta`.

ML has one unpleasant side effect for mixed models: it underestimates variance components, often substantially in small samples. The intuition is that ML does not correct for the degrees of freedom consumed by estimating the fixed effects, so the leftover variance

looks smaller than it really is. This is analogous to dividing the residual sum of squares by n rather than $n - p$ when estimating the residual variance in a linear model.

Restricted maximum likelihood (REML) fixes this problem. REML estimates the variance components from a version of the likelihood that has been purged of the fixed effects, yielding unbiased or at least much less biased variance estimates. The cost is that REML log-likelihoods depend on the fixed-effects structure, which has a practical consequence: **likelihood-ratio tests and information criteria (AIC, BIC) based on REML fits are only meaningful when the fixed-effects structure is identical across the models being compared.**

The practical rule is:

- Use **REML** (the default in `lmer()` and `lme()`) for reporting variance components, standard errors, confidence intervals, and predictions.
- Use **ML** when comparing models with different fixed-effects structures via likelihood-ratio tests or AIC/BIC.

Switching between the two in `lme4` is a single argument:

```
mod_reml <- lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy) #
REML (default)
mod_ml <- lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy, REML = FALSE)
c(REML = logLik(mod_reml), ML = logLik(mod_ml))
```

	REML	ML
	-893.2325	-897.0393

BLUEs and BLUPs

Mixed models produce two conceptually different kinds of estimates. Fixed-effect estimates are called **best linear unbiased estimators (BLUEs)**. They are unbiased in the classical sense: the expected value of the estimator equals the true parameter value.

Random-effect predictions are called **best linear unbiased predictors (BLUPs)**. They predict the (unobserved) realized values of the random effects. BLUPs are not unbiased in the same sense as BLUEs; they are “shrunk” toward zero relative to what you would get if you fitted the same factor as fixed. The shrinkage is stronger when the group-level data are sparse or noisy, and weaker when the group is well-represented. This borrowing-of-strength across groups is one of the key practical advantages of mixed models, especially in unbalanced or incomplete designs such as alpha designs and augmented designs.

The `raneff()` function extracts BLUPs, and `fixef()` extracts BLUEs:

```
fixef(mod) # BLUEs
```

	(Intercept)	Days
	251.40510	10.46729

```
head(raneff(mod)$Subject) # BLUPs for the first few subjects
```

	(Intercept)
308	40.783710
309	-77.849554
310	-63.108567

```
330 4.406442
331 10.216189
332 8.221238
```

In plant breeding, the BLUE of a genotype (from a fixed-effects model) is the right quantity for variety registration or a direct head-to-head comparison, while the BLUP (from a random-effects model) is the right quantity for selection, because shrinkage correctly down-weights noisy genotype estimates. The decision between fixed and random for genotypes is therefore dictated by the scientific question, not by a general preference.

Inference and p-values

A well-known quirk of `lme4` is that `summary()` does not report p-values for fixed effects, and `anova()` does not either:

```
anova(mod)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
      Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
Days 162703  162703     1   161   169.4 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This is not a bug. Douglas Bates, the author of `lme4`, removed p-values deliberately because there is no universally correct way to compute degrees of freedom for the t- and F-statistics in a general linear mixed model. For balanced, orthogonal designs the usual textbook formulas apply, but for unbalanced data, crossed random effects, or complex variance structures the “right” denominator degrees of freedom is an open problem. Reporting a p-value with the wrong degrees of freedom can be seriously misleading, so `lme4` refuses to guess.

Several reasonable approximations exist, and the `lmerTest` package adds them on top of `lme4`. Loading `lmerTest` changes the behaviour of `lmer()` so that `summary()` and `anova()` start reporting p-values:

```
# lmerTest was loaded with the packages block above
mod_lt <- lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy)
anova(mod_lt)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
      Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
Days 162703  162703     1   161   169.4 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By default, `lmerTest` uses the **Satterthwaite approximation** for the degrees of freedom. A more conservative alternative is the **Kenward-Roger approximation**, which also adjusts the variance-covariance matrix of the fixed effects for small-sample bias:

```
anova(mod_lt, ddf = "Kenward-Roger")
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
      Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
Days 162703  162703     1   161   169.4 < 2.2e-16 ***
```

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

Kenward-Roger is generally preferred for small or unbalanced datasets, at the cost of being slower to compute. Satterthwaite is a reasonable default for larger datasets.

! Why `anova(lmer_model)` without `lmerTest` has no p-values

Without `lmerTest` loaded, `anova()` on an `lmerMod` object only reports F-statistics and no p-values, because `lme4` does not supply denominator degrees of freedom. This is the deliberate design decision described above, not a missing feature. Loading `lmerTest` - or using `pbkrtest::KRmodcomp()` directly - is the standard workaround.

For comparisons of estimated marginal means, the `emmeans` package plugs into both `lme4` and `lmerTest` and computes p-values using Satterthwaite or Kenward-Roger degrees of freedom automatically:

```
emmeans(mod_lt, ~ Days, at = list(Days = c(0, 5, 9)))
```

Days	emmean	SE	df	lower.CL	upper.CL
0	251	9.75	22.8	231	272
5	304	9.06	17.1	285	323
9	346	9.75	22.8	325	366

```
Degrees-of-freedom method: kenward-roger
Confidence level used: 0.95
```

A Realistic Example: Alpha Design

The abstract `sleepstudy` example is useful for syntax, but mixed models shine on designed experiments with blocking. The `agridat` package ships with the classic alpha-design dataset of J. John and E. Williams [4]:

```
dat <- agridat::john.alpha
str(dat)
```

```
'data.frame':  72 obs. of  7 variables:
 $ plot : int  1 2 3 4 5 6 7 8 9 10 ...
 $ rep  : Factor w/ 3 levels "R1","R2","R3": 1 1 1 1 1 1 1 1 1 1 ...
 $ block: Factor w/ 6 levels "B1","B2","B3",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ gen  : Factor w/ 24 levels "G01","G02","G03",..: 11 4 5 22 21 10 20 2 23 14 ...
 $ yield: num  4.12 4.45 5.88 4.58 4.65 ...
 $ row  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ col  : int  1 1 1 1 1 1 1 1 1 1 ...
```

The design has 24 genotypes in 2 replicates, subdivided into incomplete blocks of 4 plots each. A naive RCBD analysis would ignore the incomplete-block structure:

```
mod_rcbd <- lm(yield ~ gen + rep, data = dat)
anova(mod_rcbd)
```

```
Analysis of Variance Table
```

```
Response: yield
```

```

      Df Sum Sq Mean Sq F value    Pr(>F)
gen     23 14.0765  0.61202   4.5475 6.259e-06 ***
rep      2  6.1355  3.06774  22.7939 1.322e-07 ***
Residuals 46  6.1910  0.13459
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

A proper alpha-design analysis treats the incomplete blocks (nested within replicate) as a random effect, which recovers information from the within-block comparisons and typically increases precision:

```

mod_alpha <- lmer(yield ~ gen + rep + (1 | rep:block), data = dat)
anova(mod_alpha)

```

```

Type III Analysis of Variance Table with Satterthwaite's method
      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
gen 10.6786  0.46429     23  34.736   5.4478 4.376e-06 ***
rep  1.5703  0.78513      2  10.394   9.2124 0.004992 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The variance component for `rep:block` quantifies how much of the total variation is explained by the incomplete blocks:

```

VarCorr(mod_alpha) %>% as.data.frame()

```

```

      grp          var1 var2          vcov          sdcov
1 rep:block (Intercept) <NA> 0.06194388 0.2488853
2 Residual          <NA> <NA> 0.08522511 0.2919334

```

For the full analysis including genotype means and letter displays, see the dedicated Alpha Design chapter.

Practical Workflow

When fitting a mixed model for a designed experiment, a workflow that holds up in practice is:

1. Write down the design: what are the treatments, what are the blocking factors, what is the unit of randomization?
2. Decide fixed versus random for each factor using the criteria above. Treatments are usually fixed, design-structure factors are usually random.
3. Fit the model with `lmer()` (or `lme()` / `glmmTMB()` if the situation demands it). Use REML for variance components and BLUPs.
4. Check the model with diagnostic plots. The same residual plots from A1. Model Diagnostics apply, with the addition that `ranef()` values should be roughly normal.
5. Load `lmerTest` and run `anova()` with Kenward-Roger degrees of freedom for small or unbalanced data, Satterthwaite otherwise.
6. Compare treatment means with `emmeans` and report BLUEs.
7. Report variance components alongside the fixed-effect inference, because they describe the study's generalizability.

💡 Additional Resources

General

- `lme4` JSS paper: D. Bates, M. Mächler, B. Bolker, and S. Walker [2]
- `nlme` book: J. C. Pinheiro and D. M. Bates [3]
- GLMM FAQ by Ben Bolker - a regularly updated troubleshooting reference
- Douglas Bates' R-help post on why `lme4` has no p-values: search for "lmer, p-values and all that"

Fixed versus random

- H. P. Piepho, A. Büchse, and K. Emrich [1] for a breeding-focused discussion

Inference

- `lmerTest` package documentation for Satterthwaite
- `pbkrtest` package documentation for Kenward-Roger and parametric bootstrap

Extensions

- `glmmTMB` vignettes for zero-inflation, non-Gaussian responses, and correlated residuals
- `emmeans` vignettes for marginal means and contrast families

i Key Takeaways

1. **Mixed models add random effects to fixed-effects models.** Random effects describe grouping structure and are modelled as draws from a normal distribution.
2. **Design-driven factors** (blocks, plots, subjects) are random regardless of how many levels they have. The 5 to 12 rule is a guideline for sample-based factors, not a hard threshold.
3. `lme4::lmer()` is the default workhorse. `nlme::lme()` is preferred when correlated residuals or variance functions are needed. `glmmTMB` covers everything else.
4. **REML** for reporting variance components and predictions; **ML** for likelihood-ratio tests comparing different fixed-effects structures.
5. **BLUES** are fixed-effect estimates (unbiased). **BLUPs** are random-effect predictions (shrunk toward zero, borrowing strength across groups).
6. `lme4` **deliberately omits p-values**. Load `lmerTest` for Satterthwaite (fast) or Kenward-Roger (more accurate in small samples) approximations.
7. `anova()` **without** `lmerTest` **returns no p-values** - this is by design, not a bug.

Bibliography

- [1] H. P. Piepho, A. Büchse, and K. Emrich, "A Hitchhiker's Guide to Mixed Models for Randomized Experiments," *Journal of Agronomy and Crop Science*, vol. 189, no. 5, pp. 310–322, 2003, doi: 10.1046/j.1439-037X.2003.00049.x.

- [2] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015, doi: 10.18637/jss.v067.i01.
- [3] J. C. Pinheiro and D. M. Bates, *Mixed-Effects Models in S and S-PLUS*. New York: Springer, 2000. doi: 10.1007/b98882.
- [4] J. John and E. Williams, "Cyclic and Computer Generated Designs," *Biometrical Journal*, vol. 38, no. 7, p. 778, 1995, doi: 10.1002/bimj.4710380703.