varTestnlme: An **R** Package for Variance Components Testing in Linear and Nonlinear Mixed-Effects Models

Charlotte Baey ⬡
Université de Lille, CNRS, UMR 8524
Laboratoire Paul Painlevé

Estelle Kuhn ⬡
Université Paris-Saclay, INRAE,
UR1404 MaIAGE

Abstract

The issue of variance components testing arises naturally when building mixed-effects models, to decide which effects should be modeled as fixed or random or to build parsimonious models. While tests for fixed effects are available in R for models fitted with **lme4**, tools are missing when it comes to random effects. The **varTestnlme** package for R aims at filling this gap. It allows to test whether a subset of the variances and covariances corresponding to a subset of the random effects, are equal to zero using asymptotic property of the likelihood ratio test statistic. It also offers the possibility to test simultaneously for fixed effects and variance components. It can be used for linear, generalized linear or nonlinear mixed-effects models fitted via **lme4**, **nlme** or **saemix**. Numerical methods used to implement the test procedure are detailed and examples based on different real datasets using different mixed models are provided. Theoretical properties of the used likelihood ratio test are recalled.

Keywords: generalized mixed-effects models, nonlinear mixed-effects models, variance components, likelihood ratio test, random effects, R.

1. Introduction

Mixed-effects models are widely used in many fields of applications, ranging from agronomy, social science, medical science, to biology. One can distinguish three main types of mixed models: linear mixed models (LMMs), generalized linear mixed models (GLMMs) and nonlinear mixed models (NLMMs). Several packages and software are available to fit these types of model, for example the SAS (SAS Institute Inc. 2020) procedures MIXED and NLMIXED for linear, generalized linear and nonlinear mixed models, or the dedicated software **NONMEM**

(ICON plc 2022) and **Monolix** (Lixoft 2021) for nonlinear mixed models, with an emphasis on pharmacodynamic and pharmacokinetic models for the former. In R (R Core Team 2023), three main packages are available to deal with these models: **lme4** (Bates, Mächler, Bolker, and Walker 2015, for LMMs, GLMMs and NLMMs), **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2022, for LMMs and NLMMs) and **saemix** (Comets, Lavenu, and Lavielle 2017, for LMMS and NLMMs). The way they handle nonlinear mixed models differs: both **lme4** and **nlme** rely on an approximation of the likelihood function based on a linearization of the model, while **saemix** uses an EM-type algorithm to derive the maximum likelihood estimate, without resorting to any approximation or linearization of the model.

When building a mixed-effects model, one is facing the delicate choice of which effects should be modeled as random. The objective can be twofold: on the one hand, being able to compare a simpler model with less random effects to a larger one can help choosing a parsimonious model, on the other hand, it can help identifying the sources of variability. This is the case for example in the context of nonlinear mixed-effect models applied to mechanistic models, where the model parameters are considered as random variables. In plant growth modeling for example (see Baey, Mathieu, Jullien, Trevezas, and Cournède 2018), identifying parameters that vary across individuals or genotypes and those that are constant allows to identify key biological processes exhibiting genotypic variability, the other being considered as species-specific. Understanding this variability is crucial to enhance our knowledge of genotype by environment interactions and then, to better understand plant adaptation to environmental constraints.

This issue has been studied from both a theoretical and a practical point of view. On the theoretical side, two main approaches have been explored. The first one follows the idea of variable selection. Chen and Dunson (2003) focused on linear mixed models and proposed a Bayesian formulation using mixture priors with a point mass at zero for the random effects. More recently, Ibrahim, Zhu, Garcia, and Guo (2011); Groll and Tutz (2014) used a penalized likelihood approach to simultaneously select fixed and random effects. Selection criteria adapted to the context of mixed-effects models were also suggested by Vaida and Blanchard (2005); Gurka (2006) and Delattre, Lavielle, and Poursat (2014). The second approach involves hypothesis testing based on variance components, with the seminal work of Stram and Lee (1994, 1995) in linear mixed models (see also the review by Molenberghs and Verbeke 2007). Recently, Baey, Cournède, and Kuhn (2019) exhibited the limiting distribution of the likelihood ratio test (LRT) statistic, for testing that the variances of any subset of the random effects are equal to zero.

On the practical side, several of the aforementioned methods have been implemented, e.g., in R. Among others, we can cite for example the **glmmLasso** (Groll 2022) package, computing an ℓ_1-penalized maximum likelihood estimator in the context of generalized linear mixed models, or the **cAIC4** (Säfken, Rügamer, Kneib, and Greven 2021) package, implementing the conditional AIC criterion. As far as hypothesis testing is concerned, most of the available tools are designed for linear or generalized linear mixed-effects models, and are mostly adapted to outputs from **lme4** package. The **lmerTest** package (Kuznetsova, Brockhoff, and Christensen 2017) provides enhanced versions of functions `anova()` and `summary()` from **lme4** package by computing p values associated to the tests for fixed effects. It also implements both the Satterthwaite's and the Kenward-Roger's approximations to correct p values in unbalanced or small sample size situations. Finally, it provides a `step()` function to simplify the random effects structure using a step-down approach. Each step is based on the computation of

p values from the LRT for testing that the variance of each random effect is equal to zero. However, the asymptotic distribution used to compute these p values is a χ^2, with the degree of freedom computed as the difference between the number of parameters under the alternative hypothesis and the number of parameters under the null hypothesis. In the context of mean components testing in linear and generalized linear models, the **pbkrtest** package (Halekoh and Højsgaard 2014) provides two alternatives to the LRT in small sample size situations, based either on the Kenward-Roger approximation or on parametric bootstrap. The two aforementioned packages can only be used with models fitted with the **lme4** package. The **RLRSim** package (Scheipl, Greven, and Küchenhoff 2008) implements the tests proposed by Crainiceanu and Ruppert (2004), based on the exact finite sample distribution of the LRT statistic. It can be used with a larger variety of packages and can handle not only models fitted from packages **lme4** and **nlme**, but also from package **gamm4** (Wood and Scheipl 2020) and from function `gamm` of package **mgcv** (Wood 2017). However, only linear mixed models with one single random effect are currently covered by this package. In SAS, the option `COVTEST` can be used for linear mixed models in `PROC MIXED` and provides Wald tests for the variance components. More recently, a SAS macro `%COVTEST` performing a bootstrap parametric test has been proposed for linear mixed models.

To the best of our knowledge, there is no software or R package implementing the LRT for variance components in LMMS, GLMMs and NLMMs, and for an arbitrary subset of variance components corresponding to any subset of the random effects.

In this paper, we present the **varTestnlme** R package (Baey and Kuhn 2023) dedicated to test that the variances of a subset of the random effects are equal to zero in LMMs, GLMMs and NLMMs with one level of random effects, using models that were fitted using either **lme4**, **nlme** or **saemix**. Moreover it is possible to test simultaneously mean and variance components. The **varTestnlme** package takes as inputs the two competitor models, fitted with the same package, and computes the asymptotic distribution of the LRT statistic associated with the two nested models corresponding to the null and alternative hypotheses. It provides the test statistic value, the limiting distribution and the associated p value. The theory behind this test has been established in Baey *et al.* (2019). The purpose of the current paper is to provide hints for a practical implementation and a complete description of the different numerical methods used for the computation of each quantity involved. In particular, we detailed the algorithms used to sample from the limiting $\bar{\chi}^2$ distribution and to estimate the weights of each component of the $\bar{\chi}^2$ distribution. We also extend results of Baey *et al.* (2019), where only the cases of a diagonal or full covariance matrix for the random effects were exemplified, to the general case of a block-diagonal covariance matrix, covering a large range of covariance structures. The package **varTestnlme** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=varTestnlme`.

Mixed-effects models and the LRT procedure are recalled in Section 2. Practical implementations of the package are presented through various examples detailed in Section 3. Technical tools for the identification of the limiting distribution of the LRT statistic are detailed in Section 4. This section can be skipped if one is interested in the practical implementation of the functions of the package. Finally, current limitations and possible extensions are discussed in Section 5.

2. Mixed-effects models and likelihood ratio test procedure

2.1. Description of the mixed-effects models

We consider the following nonlinear mixed-effects model (Davidian and Giltinan 1995, p. 98; Pinheiro and Bates 2000, p. 306; Lavielle 2014, p. 24):

$$y_i = g(\varphi_i, x_i) + \varepsilon_i, \tag{1}$$

where y_i denotes the vector of observations of size m for individual i, $1 \leq i \leq n$, g a nonlinear function, φ_i the vector of individual parameters, x_i the vector of covariates, and ε_i the random error term. The vectors of individual parameters $(\varphi_i)_{1 \leq i \leq n}$ are modeled as:

$$\varphi_i = U_i\beta + V_i b_i \ , \ 1 \leq i \leq n, \tag{2}$$

where β is the vector of fixed effects taking values in $\mathbb{R}^p$, U_i and V_i are covariates matrices of appropriate dimensions and b_i is the centered vector of random effects. We assumed that the random vectors $(b_i)_{1 \leq i \leq n}$ are independent and identically distributed centered Gaussian vectors with covariance matrix Γ of size $q \times q$, and that $(\varepsilon_i)_{1 \leq i \leq n}$ are independent and identically distributed centered Gaussian vectors with covariance matrix $\sigma^2 I_m$, where I_m is the identity matrix of size $m \times m$. Note that such models can handle non identically distributed observations (y_i), for example due to the presence of covariates as illustrated in Section 3. However, multilevel random effects are not yet covered by the current version of the package. Finally, we assume that the sequences (ε_i) and (b_i) are mutually independent. We denote by c the total number of parameters and by $\theta = (\beta, \Gamma, \sigma^2)$ the vector of parameters taking value in the set $\Theta = \{\theta \in \mathbb{R}^c \mid \ \beta \in \mathbb{R}^p, \Gamma \in \mathbb{S}_+^q, \sigma^2 > 0\}$ where $\mathbb{S}_+^q$ denotes the set of positive definite matrices of size q.

2.2. Testing variance components

We consider general test hypotheses of the following form, to test for the nullity of r variances among q:

$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta, \tag{3}$$

where $\Theta_0 \subset \Theta$. Up to permutations of rows and columns of Γ, we can assume that we are testing for the nullity of the last r variances. We then write Γ in blocks as follows:

$$\Gamma = \left(\begin{array}{c|c} \Gamma_1 & \Gamma_{12}^\top \\ \hline \Gamma_{12} & \Gamma_2 \end{array} \right),$$

with Γ_1 a $(q-r) \times (q-r)$ matrix, Γ_2 a $r \times r$ matrix, Γ_{12} a $r \times (q-r)$ matrix and where $A^\top$ denotes the transposition of matrix A, for any matrix A.

We thus have:

$$\begin{aligned} \Theta_0 &= \{\theta \in \mathbb{R}^c \mid \beta \in \mathbb{R}^p, \Gamma_1 \in \mathbb{S}_+^{q-r}, \Gamma_{12} = \mathbb{0}_{r,q-r}, \Gamma_2 = \mathbb{0}_{r,r}, \sigma^2 > 0\} \\ \Theta &= \{\theta \in \mathbb{R}^c \mid \beta \in \mathbb{R}^p, \Gamma \in \mathbb{S}_+^q, \sigma^2 > 0\}. \end{aligned} \tag{4}$$

The likelihood ratio test (LRT) statistic is defined by:

$$LRT_n = -2 \ \log \left(\frac{\sup_{\theta \in \Theta_0} L(y_1, \ldots, y_n; \theta)}{\sup_{\theta \in \Theta} L(y_1, \ldots, y_n; \theta)} \right),$$

where $L(y_1, \ldots, y_n; \theta)$ is the likelihood function.

As shown in Baey *et al.* (2019), assuming conditions (C1) and (C2) required for Theorem 1 to hold are fulfilled, the limiting distribution of the LRT statistic associated with the two hypotheses H_0 and H_1 defined in Equation 3 is:

$$LRT_n \xrightarrow[n \to \infty]{} \bar{\chi}^2(I_*^{-1}, T(\Theta, \theta^*) \cap T(\Theta_0, \theta^*)^\perp), \tag{5}$$

where I_* is the Fisher information matrix, $T(\mathcal{A}, \theta)$ is the tangent cone to space $\mathcal{A}$ at point θ, and where $\bar{\chi}^2(V, \mathcal{C})$ denotes the $\bar{\chi}^2$ distribution parametrized by the positive definite matrix V and the cone $\mathcal{C}$. The $\bar{\chi}^2$ distribution is a mixture of χ^2 distributions, where the degrees of freedom and the weights involved in the mixture depend both on V and $\mathcal{C}$. More details about the computation of the $\bar{\chi}^2$ distribution parameters are given in Section 4.

2.3. Testing simultaneously fixed effects and variance components

Results of the previous section can be extended to the case where one is testing simultaneously that a subset of the mean parameters and a subset of the covariance parameters are null. In this case, if we denote by $\mathcal{I}_f := \{k_1, \ldots, k_{r_f}\}$ the set of indices corresponding to the fixed effects which means are tested equal to 0, the parameter space Θ_0 is slightly modified as follows:

$$\Theta_0 = \{\theta \in \mathbb{R}^c \mid \beta_k = 0, k \in \mathcal{I}_f, \beta_k \in \mathbb{R}, k \notin \mathcal{I}_f, \Gamma_1 \in \mathbb{S}_+^{q-r}, \Gamma_{12} = \mathbb{0}_{r,q-r}, \Gamma_2 = \mathbb{0}_{r,r}, \sigma^2 > 0\} \tag{6}$$

The theoretical result also holds in this test setting, leading to a $\bar{\chi}^2$ distribution.

3. Practical implementation through examples

The **varTestnlme** package provides a unified framework for likelihood ratio tests of fixed and random effects of a linear, generalized linear or nonlinear mixed-effects models fitted either with the **nlme**, **lme4** or **saemix** package. The main function `varCompTest` takes, in its simplest form, two arguments: `m1` the fitted model under the alternative hypothesis H_1 and `m0` the fitted model under the null hypothesis H_0. It gives as outputs the test statistic value, the limiting distribution (i.e. the weights and degrees of freedom associated to each χ^2 component of the $\bar{\chi}^2$ distribution), as well as the p value of the test.

In this section, we give an overview of how the package works, through three examples involving linear, generalized linear and nonlinear mixed-effects models. These examples gather cases where all the quantities characterising the limiting distribution can be obtained explicitly, as well as cases where the Fisher information matrix (FIM), but also the weights of the $\bar{\chi}^2$ distribution should be approximated. In the following, we describe the datasets used to illustrate the package (Section 3.1), the different test hypotheses and the implementation of the models under each hypothesis (Section 3.2), and finally the results obtained using the `varCompTest()` function (Section 3.3). In Section 3.4, we illustrate the added-value of the FIM estimation proposed by the package, on two examples based on a linear and a nonlinear mixed models.
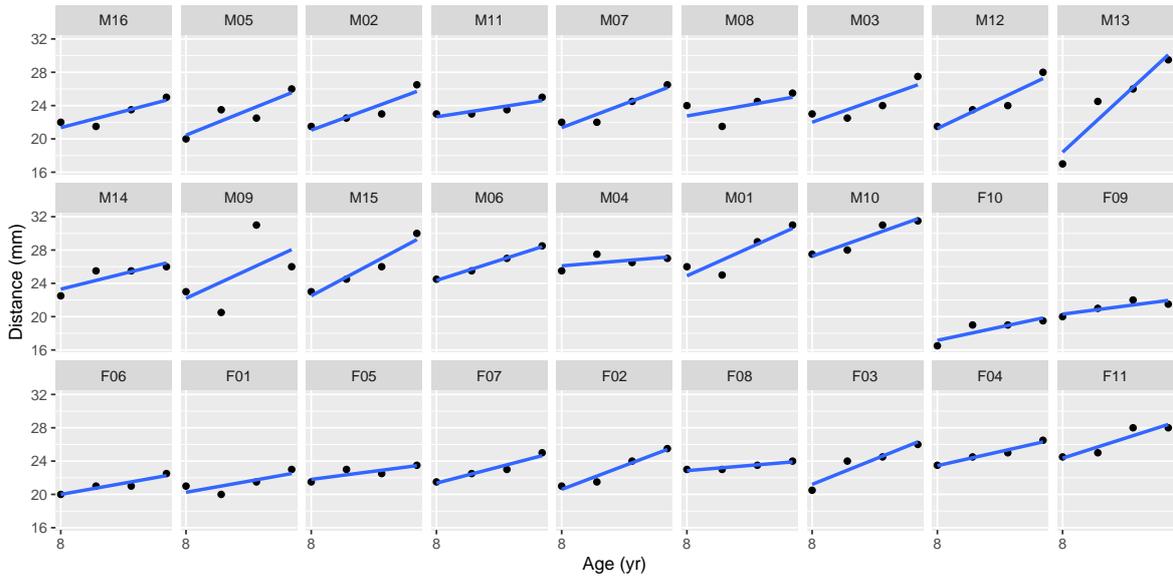
Figure 1: Data from the orthodontal study: distance between the pituitary and the pterygo-maxillary fissure as a function of age, by subject.

3.1. Data

For illustrative purposes, three datasets will be used throughout the paper, each of them covering one of the three types of models that can be treated with the **varTestnlme** package. These datasets are available in packages **nlme**, **lme4** or **datasets**.

Orthodontal data

The first dataset comes from a study on dental growth (Potthoff and Roy 1964), where the distance between the pituitary and pterygomaxillary fissure was recorded every two years from the age of 8 to the age of 14, on 27 children, 16 boys and 11 girls (see Figure 1).

```r
R> data("Orthodont", package = "nlme")
```

These data can be fitted using a linear mixed-effects model, with a random slope and a random intercept. Let us denote by y_{ij}, $1 \leq i \leq 27$, $1 \leq j \leq 4$ the dental measurement of child i of sex x_i at age t_j. Then the model can be written as:

$$
\begin{aligned}
y_{ij} &= (\beta_1 + \beta_2 x_i + b_{i1}) + (\beta_3 + \beta_4 x_i + b_{i2}) t_j + \varepsilon_{ij} \ , \\
\varepsilon_{ij} &\sim \mathcal{N}(0, \sigma^2), \quad (b_{i1}, b_{i2})^\top \sim \mathcal{N}(0, \Gamma) \ .
\end{aligned}
\tag{7}
$$

We denote by $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^\top$ the vector of fixed effects.

Bovine pleuropneumonia

The second dataset comes from a study on contagious bovine pleuropneumonia (`cbpp`) (Lesnoff *et al.* 2004), where the number of new serological cases occurring during a given time period was recorded at 4 occasions, on 15 herds (see Figure 2).
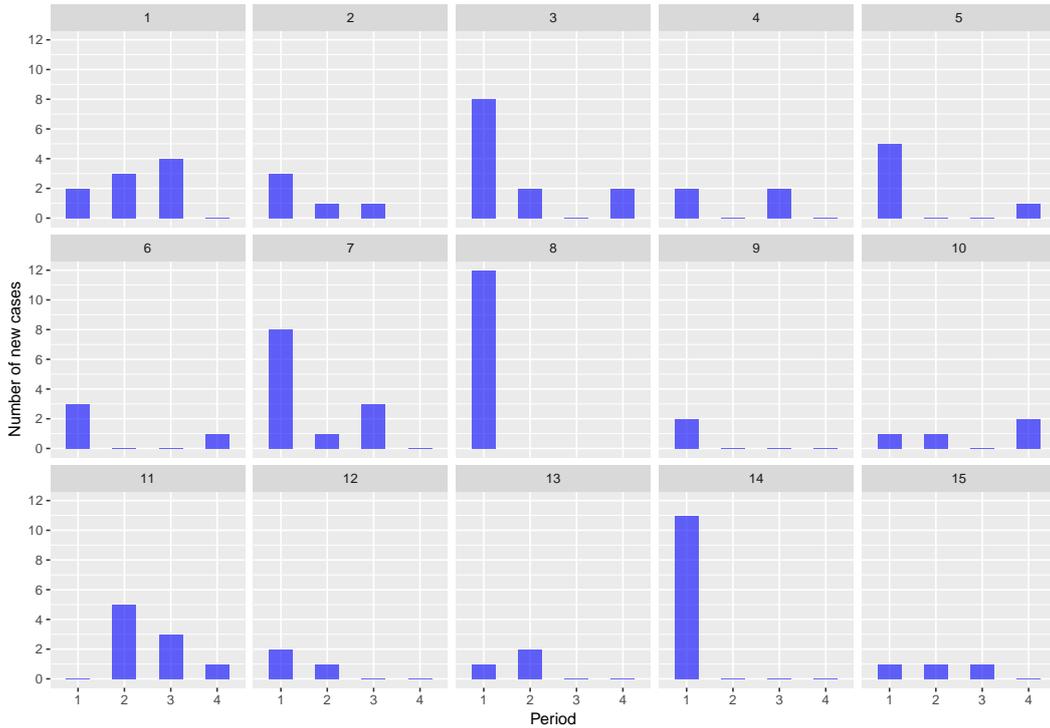
Figure 2: Data from the cbpp study: number of new cases per period, by herd.

```
R> data("cbpp", package = "lme4")
```

This data can be fitted using a generalized linear mixed-effects model and a logistic regression. Let us denote by y_{ij} the number of serological cases in herd i and at time period t_j. If we denote by g the logit function, then the model is given by:

$$\mathsf{E}(y_{ij} \mid b_i) = g^{-1}(b_i + \beta t_j),$$
$$b_i \sim \mathcal{N}(0, \Gamma) \tag{8}$$

Loblolly pine trees

The third dataset comes from a study on Loblolly pine trees (Kung 1986), where the growth of 14 trees was recorded on 6 occasions between the ages of 3 and 25 years (see Figure 3).

```
R> data("Loblolly", package = "datasets")
```

This data can be fitted using a nonlinear mixed-effects model. Let us denote by $y_{ij}, 1 \leq i \leq 14, 1 \leq j \leq 6$ the height of tree i at age x_j. We consider the following model:

$$y_{ij} = Asym_i + (R_{0i} - Asym_i) \exp(-e^{lrc_i} x_j) + \varepsilon_{ij}, \ \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$
$$(Asym_i, R_{0i}, lrc_i)^\top = \beta + b_i, \ b_i \sim \mathcal{N}_3(0, \Gamma), \tag{9}$$

where $Asym_i$, lrc_i and R_{0i} are the individual parameters corresponding respectively to the asymptote, the logarithm of the growth rate and the height at age 0 of tree i.
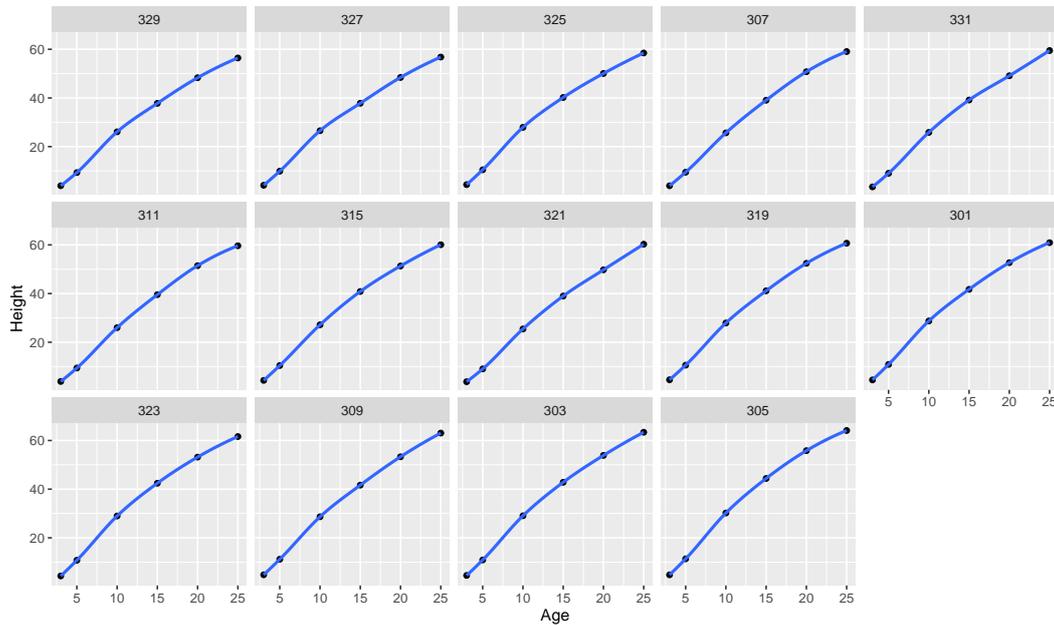
Figure 3: Data from the loblolly study: height as a function of age, by tree.

3.2. Preliminary step: Fitting the models under H_0 and H_1

The first step of the analysis consists in specifying H_0 and H_1, the two hypotheses defining the test. Then, one needs to fit the model under both hypotheses, using **nlme**, **lme4** or **saemix**. The **varTestnlme** package automatically detects the structure of the models under H_0 and H_1.

Note that both models m1 *and* m0 *should be fitted with the same package, except when there is no random effects in* m0 *(see details below).*

Linear model

We will consider three hypothesis testing configurations for the linear model presented in Equation 7. We detail below the null and alternative hypotheses in each case, and the code to fit the associated models.

1. ***Case 1:*** Testing that there is a random slope, in a model where the slope and the intercept are correlated:

$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 \\ 0 & 0 \end{pmatrix}, \sigma^2 > 0\} \text{ and } \Theta = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} \gamma_1^2 & \gamma_{12} \\ \gamma_{12} & \gamma_2^2 \end{pmatrix}, \sigma^2 > 0\}.$$

The syntax using **lme4** and **nlme** packages is:

```
R> lm1.h1.lme4 <- lmer(distance ~ 1 + Sex + age + age * Sex +
+    (1 + scale(age) | Subject), data = Orthodont, REML = FALSE)
```

```
R> lm1.h0.lme4 <- lmer(distance ~ 1 + Sex + age + age * Sex +
+    (1 | Subject),  data = Orthodont, REML = FALSE)
R> lm1.h1.nlme <- lme(distance ~ 1 + Sex + age + age * Sex,
+    random = ~ 1 + age | Subject, data = Orthodont, method = "ML")
R> lm1.h0.nlme <- lme(distance ~ 1 + Sex + age + age * Sex,
+    random = ~ 1 | Subject, data = Orthodont, method = "ML")
```

2. ***Case 2:*** Testing that there is a random slope, in a model where the slope and the intercept are independent:

$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 \\ 0 & 0 \end{pmatrix}, \sigma^2 > 0\} \text{ and } \Theta = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 \\ 0 & \gamma_2^2 \end{pmatrix}, \sigma^2 > 0\}$$

```
R> lm2.h1.lme4 <- lmer(distance ~ 1 + Sex + age + age * Sex +
+    (1 + age || Subject),  data = Orthodont, REML = FALSE)
R> lm2.h0.lme4 <- lm1.h0.lme4
R> lm2.h1.nlme <- lme(distance ~ 1 + Sex + age + age * Sex,
+    random =  list(Subject = pdDiag(~ 1 + age)), data = Orthodont,
+    method = "ML")
R> lm2.h0.nlme <- lm1.h0.nlme
```

3. ***Case 3:*** Testing for the presence of random effects, in a model where the slope and the intercept are independent:

$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \sigma^2 > 0\} \text{ and } \Theta = \{\beta \in \mathbb{R}^4, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 \\ 0 & \gamma_2^2 \end{pmatrix}, \sigma^2 > 0\}$$

```
R> lm3.h1.lme4 <- lm2.h1.lme4
R> lm3.h1.nlme <- lm2.h1.nlme
R> lm3.h0 <- lm(distance ~ 1 + Sex + age + age * Sex, data = Orthodont)
```

Generalized linear model

In the model considered in Equation 8, there is only one random effect. We will then consider the following test:

$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}, \Gamma = 0\} \text{ and } \Theta = \{\beta \in \mathbb{R}, \Gamma \geq 0\}.$$

The corresponding code using **lme4** is:

```
R> glm1 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
+    family = binomial, data = cbpp)
R> glm0 <- glm(cbind(incidence, size - incidence) ~ period,
+    family = binomial, data = cbpp)
```

Nonlinear model

Let us consider the nonlinear model defined in Equation 9. Here we will carry out the following test:
$$H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}^3, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \sigma^2 > 0\} \text{ and } \Theta = \{\beta \in \mathbb{R}^3, \Gamma = \begin{pmatrix} \gamma_1^2 & 0 & 0 \\ 0 & \gamma_2^2 & 0 \\ 0 & 0 & \gamma_3^2 \end{pmatrix}, \sigma^2 > 0\},$$

i.e., that only the asymptote is random.

The corresponding code using **nlme** and **lme4** is:

```
R> start <- c(Asym = 103, R0 = -8.5, lrc = -3.2)
R> nlm1.nlme <- nlme(height ~ SSasymp(age, Asym, R0, lrc),
+    fixed = Asym + R0 + lrc ~ 1, random = pdDiag(Asym + R0 + lrc ~ 1),
+    start = start, data = Loblolly)
R> nlm0.nlme <- nlme(height ~ SSasymp(age, Asym, R0, lrc),
+    fixed = Asym + R0 + lrc ~ 1, random = pdDiag(Asym ~ 1),
+    start = start, data = Loblolly)
R> nlm1.lme4 <- nlmer(height ~ SSasymp(age, Asym, R0, lrc)
+    ~ (0 + Asym + R0 + lrc || Seed), start = start, data = Loblolly)
R> nlm0.lme4 <- nlmer(height ~ SSasymp(age, Asym, R0, lrc)
+    ~ (0 + Asym | Seed), start = start, data = Loblolly)
```

Using **saemix**, the syntax is given by:

```
R> modelSSasymp <- function(psi, id, xidep) {
+    Asym <- psi[id, 1]
+    R0 <- psi[id, 2]
+    lrc <- psi[id, 3]
+    age <- xidep[, 1]
+    ypred <- Asym + (R0 - Asym) * exp( - exp(lrc) * age)
+    return(ypred)
+  }
R> psi0 <- matrix(start, ncol = 3,
+    dimnames = list(NULL, c("Asym", "R0", "lrc")))
R> saemix.modelH1 <- saemixModel(model = modelSSasymp,
+    description = "Asymptotic regression", psi0 = psi0,
+    transform.par = c(0, 0, 0), covariance.model = diag(3),
+    omega.init = 10 * diag(3), error.model = "constant")
```

```
R> saemix.modelH0 <- saemixModel(model = modelSSasymp,
+    description = "Asymptotic regression", psi0 = psi0,
+    transform.par = c(0, 0, 0), covariance.model = diag(c(1, 0, 0)),
+    omega.init = 10 * diag(3), error.model = "constant")
R> saemix.data <- saemixData(name.data = Loblolly,
+    name.group = "Seed", name.predictors = "age", name.response = "height")
R> ctrl <- saemixControl(print = FALSE, save = FALSE, save.graphs = FALSE)
R> nlm1.saemix <- saemix(saemix.modelH1, saemix.data, ctrl)
R> nlm0.saemix <- saemix(saemix.modelH0, saemix.data, ctrl)
```

3.3. Variance component testing

Once the models have been fitted using **lme4**, **nlme** or **saemix**, the test can be run using the
`varCompTest()` function of the **varTestnlme** package. The main function is:

```
R> varCompTest(m1, m0,
+    control = list(M = 5000, parallel = FALSE, nbcores = 1, B = 1000),
+    fim = "extract", pval.comp = "bounds", output = TRUE)
```

where:

- `m1` is the model fitted under H_1, i.e., an object of class 'merMod', 'glmerMod', 'lme',
  'nlme' or 'SaemixObject'.

- `m0` is the model fitted under H_0 using the same package as for `m1`, except when no
  random effects are present under H_0 (when testing for the absence of random effects).
  In this case, `m0` should be fitted using `lm()` for linear, `glm()` for generalized linear, and
  `nls()` for nonlinear models (these functions are available in the **stats** package).

- `control` is a list with: `M`, the Monte Carlo sample size for the computation of the $\bar{\chi}^2$
  weights (5000 by default, see Algorithm 1); `parallel` a boolean to specify whether
  the Monte Carlo sampling should be done in parallel (FALSE by default); `nbcores` the
  number of cores to be used with parallel computing (1 by default); and `B` the size of the
  bootstrap sample used to estimate the Fisher information matrix (1000 by default).

- `fim` could be either `"extract"` (the default) if the FIM should be extracted from the
  fitted model `m1`; `"compute"` if it should be computed using parametric bootstrap im-
  plemented in the package (see Section 4.1 for more details on the method); or `FIM` a
  user-defined matrix to be used as the Fisher information matrix.

- `pval.comp` specifies the way the p value of the test is computed, and could be either
  `"bounds"` (the default), in which case only bounds on the true p value are computed
  (see Section 4.5 for more details), `"approx"`, in which case a Monte Carlo estimation
  of the exact p value is provided (see Section 4.4 for more details), or `"both"` for a
  combination of both approaches. In the case where the weights are known explicitly, no
  approximation is made and the exact weights are return.

- `output` is a boolean indicating if any outputs should be printed out in the console. The
  default value is TRUE, in which case it displays information about the fitted models and
  the results of the test.

Note that the $\bar{\chi}^2$ weights approximation can be time consuming, especially when the Monte Carlo sampling size is high, or when the number of components in the $\bar{\chi}^2$ distribution increases. It is recommended to first run the function using the default setting, i.e., with `pval.comp = "bounds"`, and to possibly re-run it with `pval.comp = "approx"` if more precision is needed.

The function returns an object of classes 'htest' and 'vctest' and as such, can be printed using the provided `print()` function of **varTestnlme** or the `print.htest()` function of package **EnvStats** (Millard 2013). In particular, it contains the following slots:

- `null.value`: The value of the tested parameters under the null hypothesis.

- `alternative`: The value of the parameters under the alternative hypothesis.

- `statistic`: The value of the LRT statistic.

- `method`: The name of the statistical test.

- `parameters`: A list with the $\bar{\chi}^2$ distribution parameters: `df` the degrees of freedom, `weights` the weights associated with each component of the limiting $\bar{\chi}^2$ distribution, `sdWeights` the standard deviation associated with the estimation of each weights (equal 0 if exact weights are available) and `FIM` the estimate of the Fisher information matrix.

- `pvalue`: A named numeric vector with four elements: `pvalue.weights`, the p value obtained using Equation 17 (equals `NA` if weights were not computed, e.g., if option `pval.comp` was set to `"bounds"`), `pvalue.sample`: the p value obtained using Equation 18 (equals `NA` if weights were not computed or if exact weights were available and no sampling was done), `pvalue.lowerbound` and `pvalue.upperbound` the bounds on the p value, obtained from Equation 19.

We describe in the following subsections the practical implementation of the tests defined in Section 3.2 and detail the outputs obtained in each case.

Linear model

Case 1: Testing that the variance of age is null, in a model with two correlated random effects. We first run the function with the default arguments.

```
R> vt <- varCompTest(lm1.h1.lme4, lm1.h0.lme4)
R> print(vt)


Variance components testing in mixed effects models
Testing that the variance of the random effect associated to scale(age)
is equal to 0
Likelihood ratio test statistic:
    LRT = 0.8331072


p-value from exact weights: 0.5103454
```

Using the `summary()` function, we get the following output:

```
R>  summary(vt)
```

```
Variance components testing in mixed effects models
Testing that:
 variance of the random effect associated to scale(age) is equal to 0
against the alternative that:
 variance of the random effect associated to scale(age) > 0

 Likelihood ratio test statistic:
    LRT =  0.8331072

 Limiting distribution:
    mixture of 2 chi-bar-square distributions with degrees of freedom 1 2
    associated (exact) weights: 0.5 0.5

 p-value of the test:
    from exact weights: 0.5103454
```

The LRT statistic is computed, and the asymptotic distribution is identified as a mixture between two χ^2 distributions with degrees of freedom 1 and 2. In this case we can compute the exact weights of the $\bar{\chi}^2$ distribution and hence the exact p value. Using the **EnvStats**, it is possible to print the results of the test using function `print.htest()`.

```
R> library("EnvStats")
R> print.htestEnvStats(vt)
```

```
Results of Hypothesis Test
--------------------------


Null Hypothesis:                 variance of the random effect associated to
                                 scale(age) = 0

Alternative Hypothesis:          variance of the random effect associated to
                                 scale(age) > 0

Test Name:                       Likelihood ratio test for variance components
                                 in mixed effects models
Data:

Test Statistic:                  LRT = 0.8331072

Test Statistic Parameters:       df       = 1, 2
                                 weights  = 0.5, 0.5
                                 sdweights = 0, 0
                                 FIM      = NA

P-values:                        pvalue.weights    = 0.5103454
                                 pvalue.sample     =        NA
                                 pvalue.lowerbound = 0.5103454
                                 pvalue.upperbound = 0.5103454
```

If we re-run the function with the option `pval.comp = "both"`, we get the same results since the weights are explicit in this example. This time, we run the function with fits from **nlme** to show that results are identical.

```
R> vt <- varCompTest(lm1.h1.nlme, lm1.h0.nlme)
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 variance of the random effect associated to age is equal to 0
against the alternative that:
 variance of the random effect associated to age > 0

 Likelihood ratio test statistic:
    LRT =  0.8331072

 Limiting distribution:
    mixture of 2 chi-bar-square distributions with degrees of freedom 1 2
    associated (exact) weights: 0.5 0.5

 p-value of the test:
    from exact weights: 0.5103454
```

Case 2: Testing that the variance of age is null in a model with two independent random effects. The number of components in the mixture is the same as in *Case 1*, but the degrees of freedom have been shifted downward. Note that the weights also have simple analytical expressions in this case, as well as the *p* value. Results are presented using the **lme4** package, but similar results are obtained with the **nlme** package.

```
R> vt <- varCompTest(lm2.h1.lme4, lm2.h0.lme4)
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 variance of the random effect associated to age is equal to 0
against the alternative that:
 variance of the random effect associated to age > 0

 Likelihood ratio test statistic:
    LRT =  0.5304106

 Limiting distribution:
    mixture of 2 chi-bar-square distributions with degrees of freedom 0 1
    associated (exact) weights: 0.5 0.5

 p-value of the test:
    from exact weights: 0.2332171
```

Case 3: Testing the presence of randomness in the model, i.e., testing that the variances of age and of the intercept are null, in a model with two independent random effects.

```
R> vt <- varCompTest(lm3.h1.nlme, lm3.h0)
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of Intercept and age is equal to 0
against the alternative that:
 covariance matrix of Intercept and age > 0

 Likelihood ratio test statistic:
    LRT =  50.13311

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2

 p-value of the test:
    bounds on p-value: lower  7.18311e-13 upper  7.215163e-12

Bounds based on the smallest and biggest degrees of freedom of the chi-bar-
square distribution components. Re-run with option 'pval.comp="both" or
pval.comp="comp" to approximate the weights of each chi-bar-square component
and the p-value.
```

A message is printed in addition to the results, to give some explanation about the bounds on the p value. In this case, using the provided upper bound we see that the exact p value is smaller than 10^{-11}, which is enough in practice to reject the null hypothesis without computing the $\bar{\chi}^2$ weights and the associated p value. It is also possible to compute the weights using the option `pval.comp = "both"`. In that case, the weights are explicit but depend on the FIM. Since we are dealing with a linear mixed-effects model, the FIM has an analytical expression and can be extracted from **nlme** or **lme4** packages using the option `fim = "extract"`. This is the default behaviour of `varCompTest` function. Note that results can differ between packages since the methods used to compute the likelihood and hence the FIM is not the same in **nlme** and **lme4**.

Using **nlme** package:

```
R> vt <- varCompTest(lm3.h1.nlme, lm3.h0, pval.comp = "both")
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of Intercept and age is equal to 0
against the alternative that:
 covariance matrix of Intercept and age > 0
```

```
 Likelihood ratio test statistic:
    LRT =  50.13311

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
    associated (exact) weights: 0.3767598 0.5000000 0.1232402

 p-value of the test:
    from exact weights: 2.319657e-12
    bounds on p-value: lower  7.18311e-13 upper  7.215163e-12
```

Using **lme4** package:

```
R> vt <- varCompTest(lm3.h1.lme4, lm3.h0, pval.comp = "both")
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of (Intercept) and age is equal to 0
against the alternative that:
 covariance matrix of (Intercept) and age > 0

 Likelihood ratio test statistic:
    LRT =  50.13311

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
    associated (exact) weights: 0.3839687 0.5000000 0.1160313

 p-value of the test:
    from exact weights: 2.225988e-12
    bounds on p-value: lower  7.18311e-13 upper  7.215163e-12
```

To compute the FIM using parametric bootstrap (see Section 4.1 for more details on the method), one should use the option `fim = "compute"`. The default bootstrap sample size is $B = 1000$, but it can be changed using the `control` argument.

Using **nlme** package:

```
R> vt <- varCompTest(lm3.h1.nlme, lm3.h0, pval.comp = "both",
+    fim = "compute")
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of Intercept and age is equal to 0
against the alternative that:
 covariance matrix of Intercept and age > 0
```

```
Likelihood ratio test statistic:
   LRT =  50.13311

Limiting distribution:
   mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
   associated (exact) weights: 0.3493855 0.5000000 0.1506145

p-value of the test:
       from exact weights: 2.675351e-12
   bounds on p-value: lower  7.18311e-13 upper  7.215163e-12
```

Using **lme4** package:

```
R> vt <- varCompTest(lm3.h1.lme4, lm3.h0, pval.comp = "both",
+    fim = "compute")
R> summary(vt)

Variance components testing in mixed effects models
Testing that:
 covariance matrix of (Intercept) and age is equal to 0
against the alternative that:
 covariance matrix of (Intercept) and age > 0

Likelihood ratio test statistic:
   LRT =  50.13311

Limiting distribution:
   mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
   associated (exact) weights: 0.3609427 0.5000000 0.1390573

p-value of the test:
       from exact weights: 2.525181e-12
   bounds on p-value: lower  7.18311e-13 upper  7.215163e-12
```

Generalized linear model

Results from the test of the null hypothesis that there is no random effect associated with the herd are given by:

```
R> vt <- varCompTest(glm1, glm0)
R> summary(vt)

Variance components testing in mixed effects models
Testing that:
 variance of the random effect associated to (Intercept) is equal to 0
against the alternative that:
 variance of the random effect associated to (Intercept) > 0
```

```
Likelihood ratio test statistic:
    LRT =  14.00527


Limiting distribution:
    mixture of 2 chi-bar-square distributions with degrees of freedom 0 1
    associated (exact) weights: 0.5 0.5


p-value of the test:
    from exact weights: 9.114967e-05
```

Nonlinear model

Results from the test of the null hypothesis that only the asymptote is random against the alternative hypothesis of a diagonal covariance matrix between the random effects, using **nlme** and **lme4**, are given by:

```
R> vt <- varCompTest(nlm1.nlme, nlm0.nlme)
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of R0 and lrc is equal to 0
against the alternative that:
 covariance matrix of R0 and lrc > 0

 Likelihood ratio test statistic:
    LRT =  2.519869

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2

 p-value of the test:
    bounds on p-value: lower  0.05620995 upper  0.1980462


R> vt <- varCompTest(nlm1.lme4, nlm0.lme4)
R> summary(vt)


Variance components testing in mixed effects models
Testing that:
 covariance matrix of R0 and lrc is equal to 0
against the alternative that:
 covariance matrix of R0 and lrc > 0

 Likelihood ratio test statistic:
    LRT =  2.456656
```

```
Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2

 p-value of the test:
    bounds on p-value: lower  0.05851384 upper  0.2049047
```

Depending on the desired level of the test, more precision may be needed on the *p* value. For models fitted with **nlme**, this can be obtained using the options `pval.comp = "both"` or `pval.comp = "approx"` and `fim = "extract"` to use the FIM computed by `nlme`, or `fim = "compute"` to approximate the FIM using parametric bootstrap, with a default bootstrap sample size of 1000.

```
R> vt <- varCompTest(nlm1.nlme, nlm0.nlme, pval.comp = "both",
+    fim = "compute")
R> summary(vt)

Variance components testing in mixed effects models
Testing that:
 covariance matrix of R0 and lrc is equal to 0
against the alternative that:
 covariance matrix of R0 and lrc > 0

 Likelihood ratio test statistic:
    LRT =  2.519869

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
    associated (exact) weights: 0.2450473 0.5000000 0.2549527

 p-value of the test:
    from exact weights: 0.128533
    bounds on p-value: lower  0.05620995 upper  0.1980462
```

Here again, the weights are exact once the FIM is known. For information purposes, the above code took approximatively 175 seconds on a laptop with a 8 cores Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz processor.

For nonlinear models fitted with **lme4**, to the best of our knowledge no method is available to extract the FIM (although **merDeriv**, Wang and Merkle 2018, provides the FIM for linear and generalized linear models fitted via **lme4**). Only the option `fim = "compute"` can then be used. We get the following results, which are very similar to the ones obtained with the **nlme** package. However the code was a bit longer to run, and took 430 seconds.

```
R> vt <- varCompTest(nlm1.lme4, nlm0.lme4, pval.comp = "both",
+    fim = "compute")
R> summary(vt)

Variance components testing in mixed effects models
Testing that:
```

```
 covariance matrix of R0 and lrc is equal to 0
against the alternative that:
 covariance matrix of R0 and lrc > 0

 Likelihood ratio test statistic:
    LRT =  2.456656

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
    associated (exact) weights: 0.2648237 0.5000000 0.2351763

 p-value of the test:
    from exact weights: 0.1273691
    bounds on p-value: lower  0.05851384 upper  0.2049047
```

Using the **saemix** package and the option `fim = "extract"`, we get the following results, which are consistent with those obtained with **lme4** and **nlme** (note that the likelihood value is different since **saemix** uses an important sampling approach to estimate the likelihood when **lme4** and **nlme** rely on a linearization of the likelihood):

```
R> vt <- varCompTest(nlm1.saemix, nlm0.saemix, pval.comp = "both",
+    fim = "extract")
R> summary(vt)

Variance components testing in mixed effects models
Testing that:
 covariance matrix of omega2.R0 and omega2.lrc is equal to 0
against the alternative that:
 covariance matrix of omega2.R0 and omega2.lrc > 0

 Likelihood ratio test statistic:
    LRT =  1.983014

 Limiting distribution:
    mixture of 3 chi-bar-square distributions with degrees of freedom 0 1 2
    associated (exact) weights: 0.2805023 0.5000000 0.2194977

 p-value of the test:
    from exact weights: 0.160974
    bounds on p-value: lower  0.07953661 upper  0.2650452
```

3.4. Comparison using different FIM estimates

To evaluate the performance of the FIM estimation via parametric bootstrap implemented in **varTestnlme**, we compared the empirical coverages of the nominal 95% confidence intervals based on the asymptotic normality property of the MLE obtained using the FIM extracted from packages **nlme**, **lme4** and **saemix**, and our estimate.

Parameter	Extracted		Boostrap			
	nlme	**lme4**	using **nlme** estimates		using **lme4** estimates	
			$B = 100$	$B = 300$	$B = 100$	$B = 300$
β_1	0.939	0.938	0.932	0.928	0.952	0.934
β_2	0.949	0.949	0.910	0.856	0.952	0.948
γ_1^2	0.944	0.942	0.930	0.916	0.938	0.930
γ_2^2	0.851	0.916	0.918	0.926	0.908	0.912
$\gamma_{1,2}$	0.978	0.931	0.934	0.930	0.932	0.932
σ^2	0.949	1.000	0.938	0.950	0.956	0.948

Table 1: Empirical coverage of the 95% asymptotic confidence intervals according to the type of FIM estimate, in a linear mixed-effect model with two correlated random effects, computed on 500 repetitions.

A linear mixed-effects model with two correlated random effects was fitted to 500 simulated datasets of size $n = 100$.

$$y_{ij} = \beta_1 + \beta_2 t_j + b_{i1} + b_{i2} t_j + \varepsilon_{ij}, \quad i = 1, \ldots, n, j = 1, \ldots, m$$

$$\begin{pmatrix} b_{i1} \\ b_{i2} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \gamma_1^2 & \gamma_{12} \\ \gamma_{12} & \gamma_2^2 \end{pmatrix} \right), \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

We took $\beta_1 = 5, \beta_2 = 7, \gamma_1 = 0.8, \gamma_2 = 1, \gamma_{12} = 0.4$ and $\sigma^2 = 1.2^2$. The FIM was extracted and estimated for each of the simulated datasets, with a bootstrap sample size $B \in \{100, 300\}$. Results are given in Table 1. Note that small discrepancies can be observed for the **nlme** package when running the replication material, due to different versions of the linear algebra library and the way it handles matrix multiplication. Results show that good results are obtained for small values of B. Interestingly, the bootstrap estimate lead to better coverages than the extracted estimate obtained using **lme4**, for variance components. This is crucial since the FIM components involved in the $\bar{\chi}^2$ weights computation are those corresponding to the variance components.

4. Tools for the identification of the limiting distribution

In this section, we detail the technical tools used to identify the key quantities characterizing the limiting $\bar{\chi}^2$ distribution, which is a mixture of χ^2 distributions. These quantities are the degrees of freedom and the associated weights of the χ^2 distributions in the mixture and depend on the FIM and on the tangent cone. Note that an exact expression of the tangent cone can always be derived in the general case of a block-diagonal covariance matrix Γ. However the FIM is not always available in a closed form, especially for nonlinear mixed-effects models. In Section 4.1, we present the parametric bootstrap procedure used to estimate the FIM, in Section 4.2, we give details on how to identify the tangent cone $T(\Theta, \theta^*) \cap T(\Theta_0, \theta^*)^\perp$, and in Section 4.3 we describe the Monte Carlo algorithm used to compute the weights of each χ^2 distribution. We also explain how the p value of the test can be computed, and how bounds can be obtained.

4.1. Estimation of the Fisher information matrix

In linear mixed-effects models, the FIM can be computed exactly. However this is not the case for nonlinear mixed-effects models. The existing R packages provide in some cases an approximation of the FIM based for example on a linearisation of the model, or on a block-diagonal approximation of the FIM where a zero correlation is assumed between fixed effects and variance components. Nevertheless, in general the FIM is not block-diagonal. Moreover these estimates are in general only valid for large samples, leading to possible wrong conclusions when considering small samples. Specific tools have to be developed for such settings.

In the **varTestnlme** package, we provide three options for the FIM. The first option is specifically adapted to small samples, and is more accurate in this setting albeit more time consuming. It consists in building an estimate of the FIM via parametric bootstrap. Let us denote by $\hat{\theta}_1$ the estimate of θ obtained for the alternative model under H_1, with **lme4**, **nlme** or **saemix**. We generate B bootstrap samples $(\varphi_i^{u,*}, y_i^{u,*}), 1 \leq i \leq n, 1 \leq u \leq B$ from the alternative model, using the parameter value $\hat{\theta}_1$ and Equations 1-2. Then, for each of these bootstrap samples, we compute the associated estimate $\hat{\theta}_u^*$. The bootstrap estimate of the FIM is then given by the empirical covariance matrix between the bootstrap estimates:

$$\hat{I}^* = \frac{1}{B} \sum_{u=1}^{B} \hat{\theta}_u^* (\hat{\theta}_u^*)^\top - \bar{\hat{\theta}}^* (\bar{\hat{\theta}}^*)^\top, \tag{10}$$

where $\bar{\hat{\theta}}^* = \frac{1}{B} \sum_{u=1}^{B} \hat{\theta}_u^*$.

The second option consists in using the FIM estimate provided by the package that was used to fit the models, when this estimate is available. Finally, the third option relies on a user-defined positive semi-definite matrix which should be an estimate of the FIM. It can be for example computed using the R package PFIM 4.0 (Dumont *et al.* 2018) based on Monte Carlo tools or by implementing the method presented in Delattre and Kuhn (2019) based on stochastic approximation tools.

4.2. Computation of the tangent cone

The identification of the tangent cone is crucial since it allows to identify the number of components in the $\bar{\chi}^2$ distribution, and the degrees of freedom associated to each of these components. Indeed, results from Shapiro (1985) state that, provided that one can found the largest linear subspace contained in the tangent cone, and the smallest linear subspace containing the tangent cone, then the degrees of freedom of the χ^2 components take values between the dimensions of these two subspaces. In this section, we give the exact expression of the tangent cone, and we identify the dimensions of the two aforementioned linear subspaces in the general case of a block-diagonal covariance matrix Γ.

In Baey *et al.* (2019), it was highlighted that the limiting distribution of the LRT statistic depends on the structure of Γ and its expression was exhibited in two common cases, when Γ was either diagonal or full. In the general case, Γ can always be written in the following block-diagonal form:

$$\Gamma = \begin{pmatrix} \Gamma_1 & 0 & \dots & 0 \\ 0 & \Gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \Gamma_K \end{pmatrix}, \tag{11}$$

where $K \in \mathbb{N}^*$, and for all $1 \leq k \leq K$, Γ_k is a positive semi-definite matrix of size $r_k \times r_k$ with $\sum_1^K r_k = q$. In the sequel, we assume that the blocks $\Gamma_1, \ldots, \Gamma_K$ are full i.e., all the covariances inside these blocks are non-null. This is how the covariance matrix of random effects is implemented in package **lme4** (Bates *et al.* 2015, p. 7, Section 2.2). In the appendix of their paper, the authors mentioned that more general structures for the covariance matrix Γ could be implemented using modularized functions of the **lme4** package. Nevertheless these functionalities are not currently treated in the **varTestnlme** package. In **nlme**, more sophisticated covariance structures can be used, and in **saemix**, no specific restriction is imposed on the covariance matrix structure. However in **varTestnlme** we restrict ourselves to the block-diagonal case with full blocks. This restriction allows for a simple block-diagonal structure of the covariance matrix and eases the computation.

When the covariance matrix Γ is block-diagonal, three cases can arise when testing that a subset of variances are null: i) we are testing that R blocks among the K blocks $\Gamma_1, \ldots, \Gamma_K$ are null, ii) we are testing that R sub-blocks of the K blocks $\Gamma_1, \ldots, \Gamma_K$ are null, or iii) a mixture of i) and ii). Let us denote by ℓ_0 the number of blocks where covariances are tested equal to 0 without testing that the corresponding variances are equal to 0 (only testing non-diagonal elements), by ℓ_1 the number of blocks which are tested entirely equal to 0, and by ℓ_2 the number of blocks where sub-blocks are tested equal to 0. By sub-block, we mean that we are testing a sub-matrix which is strictly smaller than the block matrix from which it was extracted. We assume that $0 \leq \ell_0 \leq K$, $0 \leq \ell_1 \leq K$, $0 \leq \ell_2 \leq K$ and $1 \leq \ell_0 + \ell_1 + \ell_2 \leq K$.

Without loss of generality, and up to a permutation of rows and columns of Γ, we can assume that the blocks are in the following order: first, the blocks which are not tested, then the blocks where only covariances are tested, next the blocks which are tested entirely equal to zero, and finally the blocks where diagonal sub-blocks are tested equal to zero. Then, the null and alternative hypotheses in Equation 3 can be written as:

$$
\begin{aligned}
\Theta_0 = \{\theta \in \mathbb{R}^c \mid \ &\beta_k = 0, k \in \mathcal{I}_f, \beta_k \in \mathbb{R}, k \notin \mathcal{I}_f, \\
&\forall k = 1, \ldots, K - (\ell_0 + \ell_1 + \ell_2), \Gamma_k \in \mathbb{S}_+^{r_k}, \\
&\forall k = K - (\ell_0 + \ell_1 + \ell_2) + 1, \ldots, K - (\ell_1 + \ell_2), \Gamma_k \in \mathbb{S}_+^{r_k}, \text{ with } t_k \text{ covariances null,} \\
&\forall k = K - (\ell_1 + \ell_2) + 1, \ldots, K - \ell_2, \Gamma_k = 0, \\
&\forall k = K - \ell_2 + 1, \ldots, K, \Gamma_k = \left(\begin{array}{c|c} \tilde{\Gamma}_k & 0 \\ \hline 0 & 0 \end{array} \right), \text{ with } \tilde{\Gamma}_k \in \mathbb{S}_+^{r_k - s_k}, \\
&\sigma^2 > 0\}
\end{aligned}
$$

$$
\Theta = \{\theta \in \mathbb{R}^c \mid \ \beta \in \mathbb{R}^p, \Gamma_k \in \mathbb{S}_+^{r_k}, \sigma^2 > 0\}
$$

From the general expressions of Θ_0 and Θ in Equation 4, we can derive the expressions of $T(\Theta, \theta^*)$ and $T(\Theta_0, \theta^*)$ using the results of Hiriart-Urruty and Malick (2012), and consequently the expression of the closed convex cone $\mathcal{C} := T(\Theta, \theta^*) \cap T(\Theta_0, \theta^*)^\perp$, involved in the $\bar{\chi}^2$ distribution in Equation 5.

We obtain:

$$
\mathcal{C} = \{0\}^a \times \underbrace{\mathbb{R}^{r_f}}_{\substack{\text{tested} \\ \text{fixed effects}}} \times \underbrace{\prod_{k=K-(\ell_0+\ell_1+\ell_2)+1}^{K-(\ell_1+\ell_2)} \mathbb{R}^{t_k}}_{\substack{\text{covariances tested without testing} \\ \text{the corresponding variances}}} \times \underbrace{\prod_{k=K-(\ell_1+\ell_2)+1}^{K-\ell_2} \mathbb{S}_+^{r_k}}_{\text{fully tested blocks}}
$$

$$
\times \underbrace{\prod_{k=K-\ell_2+1}^{K} \{\mathbb{R}^{s_k(r_k-s_k)} \times \mathbb{S}_+^{s_k}\}}_{\text{sub-blocks tested}},
\tag{12}
$$

where

$$
a = \underbrace{p - r_f}_{\substack{\text{non tested} \\ \text{fixed effects}}} + \underbrace{\sum_{k=1}^{K-(\ell_0+\ell_1+\ell_2)} \frac{r_k(r_k-1)}{2}}_{\text{non tested blocks}} + \underbrace{\sum_{k=K-\ell_2+1}^{K} \frac{(r_k-s_k)(r_k-s_k+1)}{2}}_{\substack{\text{untested sub-blocks in blocks} \\ \text{where a sub-block is tested}}} + \underbrace{1}_{\substack{\text{residual} \\ \text{variance}}}.
\tag{13}
$$

To identify the components of the $\bar{\chi}^2$ mixture, i.e., the degrees of freedom and the weights of all the χ^2 distributions involved in the mixture, we can use the properties enounced by Shapiro (1985), stating that if $\mathcal{C}$ contains a linear space of dimension d_1, the first d_1 weights of the mixture are null, and if $\mathcal{C}$ is included in a linear space of dimension d_2, the last $(c-d_2)$ weights of the mixture are null. The $\bar{\chi}^2$ distribution $\bar{\chi}^2(I_*^{-1}, \mathcal{C})$ is then a mixture of $(d_2 - d_1 + 1)$ χ^2 distributions with degrees of freedom varying between d_1 and d_2.

According to the general formulation of the cone $\mathcal{C}$ given in Equation 12, we have:

$$
d_1 = r_f + \sum_{k=K-(\ell_0+\ell_1+\ell_2)+1}^{K-(\ell_1+\ell_2)} t_k + \sum_{k=K-\ell_2+1}^{K} s_k(r_k - s_k),
$$

$$
d_2 = c - a.
\tag{14}
$$

In particular, if only blocks of variances are tested (i.e., if $r_f = 0$, $\ell_0 = 0$ and $\ell_2 = 0$), then $d_1 = 0$ and there is a Dirac mass at 0 in the mixture.

From Equations 13 and 14, we can see that the number of elements in the χ^2 mixture only depends on the number of variances being tested, and on the structure of the covariance matrix. When r_f fixed effects are tested simultaneously with a set of variance components, the number of elements in the mixture is the same as in the case where only the set of variance components is tested, but the degrees of freedom of each element in the χ^2 mixture is shifted upward by r_f.

As an example, let us consider a model with 3 random effects, with $\varphi_i = \beta + b_i$, $b_i \sim \mathcal{N}(0, \Gamma)$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_m)$ with Γ a positive definite covariance matrice and $\sigma^2 > 0$. Let us consider the following hypotheses:

$$
H_0 : \theta \in \Theta_0 \quad \text{against} \quad H_1 : \theta \in \Theta,
\tag{15}
$$

with

$$\Theta_0 = \{\beta \in \mathbb{R}^3, \Gamma = \begin{pmatrix} \Gamma_1^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \sigma^2 > 0\}$$

$$\Theta = \{\beta \in \mathbb{R}^3, \Gamma = \begin{pmatrix} \Gamma_1^2 & \Gamma_{12} & \Gamma_{13} \\ \Gamma_{12} & \Gamma_2^2 & \Gamma_{23} \\ \Gamma_{13} & \Gamma_{23} & \Gamma_3^2 \end{pmatrix}, \sigma^2 > 0\}.$$

Since no fixed effect is tested, we have $r_f = 0$. We can consider Γ as a block-diagonal matrix with only $K = 1$ block. In this block of size 3×3, we are testing that a sub-block of size 2×2 is equal to 0. Using the above notations, we have $\ell_0 = 0, \ell_1 = 0, \ell_2 = 1$, $r_1 = 3$, $s_1 = 2$ and $c = 10$. We thus have: $a = 3 - 0 + 0 + \frac{(3-2)(3-2+1)}{2} + 1 = 5$, $d_1 = 0 + 1(3 - 1) = 2$ and $d_2 = 10 - 5 = 5$. The limiting distribution of the LRT associated to the two above hypotheses is then a mixture of $d_2 - d_1 + 1 = 4$ χ^2 distributions with degrees of freedom 2, 3, 4 and 5.

4.3. Computation of the $\bar{\chi}^2$ weights

In some simple cases, the $\bar{\chi}^2$ weights have an analytical expression, which may depends on the FIM. In general however, the weights are non explicit and have to be estimated via Monte Carlo techniques as described in Algorithm 1.

In **varTestnlme**, when the weights of the $\bar{\chi}^2$ distribution are not available in a closed form, they are estimated according to the procedure proposed by Silvapulle and Sen (2011), using an estimate of the FIM. The general idea relies on the simulation of M i.i.d. realizations $X_1, \ldots, X_M$ from the limiting chi-bar-square distribution using its definition as the norm of the projection of a multivariate Gaussian random variable on a closed convex cone. More precisely, let $\mathcal{C}$ be a closed convex cone of $\mathbb{R}^c$, V a positive definite matrix of size $c \times c$ and $Z \sim \mathcal{N}(0, V)$. Then, the random variable X defined below follows a $\bar{\chi}^2$ distribution with parameters V and $\mathcal{C}$ as detailed in Silvapulle and Sen (2011):

$$X = Z^\top V^{-1} Z - \min_{\theta \in \mathcal{C}} (Z - \theta)^\top V^{-1} (Z - \theta). \tag{16}$$

The algorithm first simulate M realizations of a multivariate Gaussian distributions, $Z_1, \ldots, Z_M$, and then computes each term of Equation 16 using quadratic programming when necessary to get the projection on $\mathcal{C}$. This gives M realizations $X_1, \ldots, X_M$ following the $\bar{\chi}^2$ distribution $\bar{\chi}^2(V, \mathcal{C})$. Since the number of components in the $\bar{\chi}^2$ mixture and the associated degrees of freedom are known, the weights of each χ^2 component can be retrieved by defining a system of linear equations in w, the vector of weights. More details are given in Algorithm 1.

4.4. Computation of the p value

Let F_d be the cumulative distribution function (CDF) of the χ^2 distribution with d degrees of freedom. The p value of the test can then be estimated in two different ways, both of them

---

Algorithm 1 Estimation of the weights of the $\bar{\chi}^2$ distribution using Monte Carlo simulations. Note that due to the (pseudo-)randomness of the sequence (Z_i), results will differ from one run of the algorithm to the other. For reproducible results, one might want to fix the seed beforehand.

1: **Define** $\hat{I}^{-1}$ an estimate of the inverse of the FIM under H_1, and let $c_{d_1+2} < \cdots < c_{d_2}$ be a sequence of non-negative increasing numbers (see Figure 4).

2: **for** $i = 1, \ldots, M$ **do**

     1. simulate $Z_i \sim \mathcal{N}(0, \hat{I}^{-1})$

     2. compute

$$X_i := Z_i^\top \hat{I}^{-1} Z_i - \min_{\theta \in \mathcal{C}} (Z_i - \theta)^\top \hat{I}^{-1} (Z_i - \theta),$$

     using quadratic programming when $\mathcal{C}$ is the non-negative orthant $\mathbb{R}^{c'}$, for some $c' \leq c$, and using general nonlinear optimization tools otherwise. In **varTestnlme**, we use respectively the **quadprog** (Turlach, Weingessel, and Moler 2019) and the **alabama** (Varadhan 2022) packages.

3: **end for**

4: Compute matrix A of size $(d_2 - d_1 + 1) \times (d_2 - d_1 + 1)$:

$$A = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 & 1 \\ 1 & 0 & 1 & \ldots & 1 & 0 \\ F_{d_1}(c_{d_1+2}) & F_{d_1+1}(c_{d_1+2}) & F_{d_1+2}(c_{d_1+2}) & \ldots & F_{d_2-1}(c_{d_1+2}) & F_{d_2}(c_{d_1+2}) \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ F_{d_1}(c_{d_2}) & F_{d_1+1}(c_{d_2}) & F_{d_1+2}(c_{d_2}) & \ldots & F_{d_2-1}(c_{d_2}) & F_{d_2}(c_{d_2}) \end{pmatrix},$$

     where F_i is the cumulative distribution function of a χ^2 distribution with i degrees of freedom.

     N.B. The expression of A given above corresponds to the case where $(d_2 - d_1 + 1)$ is even. In the case where $(d_2 - d_1 + 1)$ is odd, the last column has a '1' in the second row, instead of a '0'.

5: Compute vector $\hat{\kappa}$ of size $(d_2 - d_1 + 1)$:

$$\hat{\kappa} = \begin{pmatrix} 1 \\ 1/2 \\ \frac{1}{M} \sum_{i=1}^M \mathbb{1}_{X_i \leq c_{d_1+2}} \\ \vdots \\ \frac{1}{M} \sum_{i=1}^M \mathbb{1}_{X_i \leq c_{d_2}} \end{pmatrix}$$

6: Estimate the weights of the $\bar{\chi}^2$ distribution by solving the system:

$$A\hat{w} = \hat{\kappa} \Leftrightarrow \hat{w} = A^{-1}\hat{\kappa}$$

7: Estimate the covariance matrix of the weights by:

$$\mathsf{VAR}(\hat{w}) = A^{-1} \, \mathsf{VAR}(\hat{\kappa}) \, (A^{-1})^\top$$
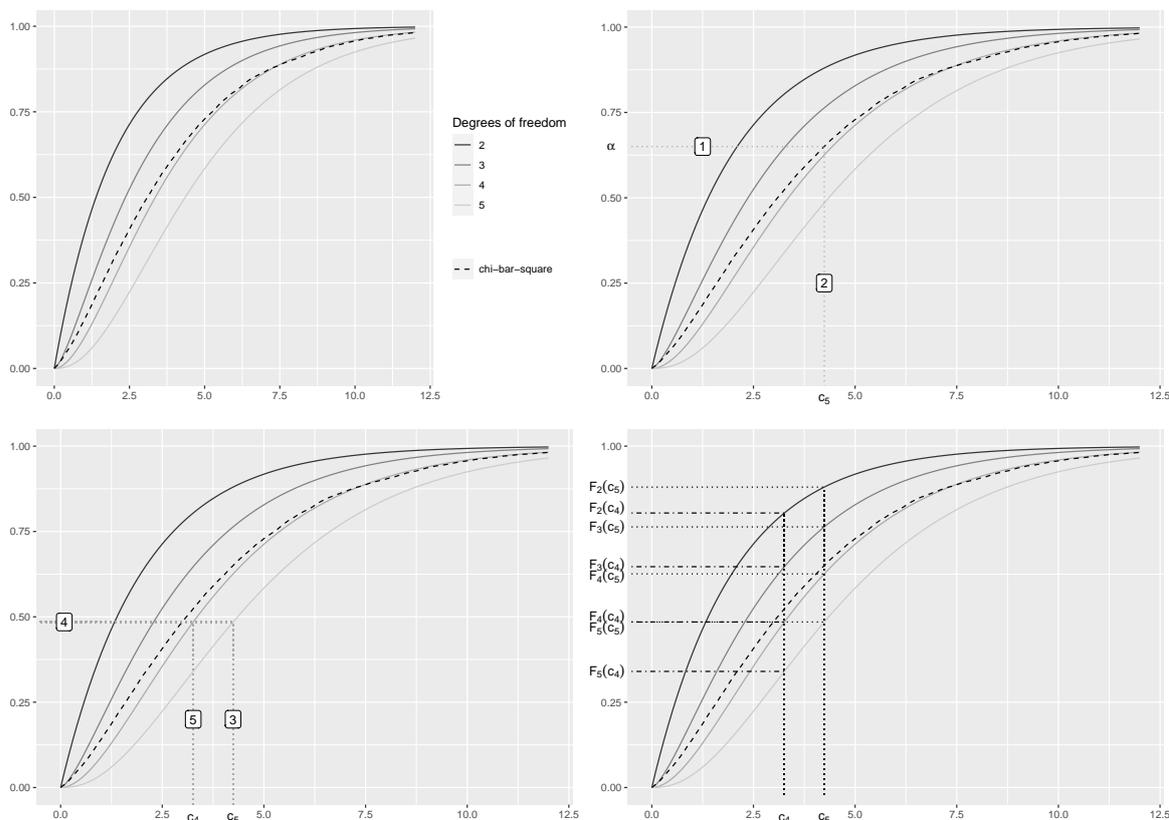

Figure 4: Illustration of the choice of the non-increasing sequence $c_{d_1+2} < \cdots < c_{d_2}$ used in Algorithm 1 to approximate the $\bar{\chi}^2$ weights.

being computed in the **varTestnlme**:

$$\hat{p}_1 = \sum_{j=d_1}^{d_2} \hat{w}_j \left(1 - F_j(LRT_n)\right), \tag{17}$$

$$\hat{p}_2 = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{X_i \geq LRT_n}, \tag{18}$$

where $\hat{w}_j$ is the estimated weight associated to the χ^2 distribution with j degrees of freedom, and where $X_1, \ldots, X_M$ are simulated according to the limiting $\bar{\chi}^2$ distribution (see Algorithm 1 for more details on the notations).

Note that for very small values of the real p value, a very large sample size M would be needed in order to get a non-zero estimate $\hat{p}_2$.

4.5. Bounds on p values

Since the simulation of $X_1, \ldots, X_M$ can be time consuming, and since $F_i(c) \geq F_j(c)$ for $i < j$, it is possible to compute bounds on the p value of the test. Indeed, since the sum of all the weights is equal to 1, and the sum of even (resp. odd) weights is equal to $1/2$, natural bounds

are given for the p value by Silvapulle and Sen (2011):

$$1 - \frac{F_{d_1}(LRT_n) + F_{d_1+1}(LRT_n)}{2} \leq p \leq 1 - \frac{F_{d_2-1}(LRT_n) + F_{d_2}(LRT_n)}{2}. \qquad (19)$$

Even if those bounds can be crude, they are available in closed form and might bring enough information in some cases to support or reject the null hypothesis. By default, the **varTestnlme** package returns these bounds, and if more precision is needed by the user, it is possible to re-run the analysis in order to estimate the weights of the limiting distribution.

5. Summary and discussion

In this paper, we present the R package **varTestnlme**, which provides a unified framework for variance components testing in linear, generalized linear and nonlinear mixed-effects models fitted with **nlme**, **lme4** and **saemix**. The main function `varCompTest()` performs the comparison of two nested models `m1` and `m0`. It is thus possible to test that the variances of a subset of random effects is null, taking into account correlation between them, in a model with one level of random effects. It is also possible to test that one or several correlations are null, and to test simultaneously if some fixed effects are null. The tests are performed using likelihood ratio test and rely on the asymptotic properties of the LRT statistic. Additional tools are provided to approximate the Fisher information matrix when it is needed to compute the limiting distribution of the LRT statistic.

Possible future developments of the package include in particular the consideration of more than one level of random effects, to account for multiple nested random effects. Another perspective would be to allow for more general structures for the covariance matrix Γ of the random effects. This would encompass for example parametric structures such as compound symmetry, auto-regressive or with spatial correlation, as proposed in the **nlme** package. Another interesting development would be the generalisation to models with non-Gaussian random effects, for example using heavy-tailed distributions. Specific tools would be needed to extend the theoretical results to other families of distributions. Finally, it would be useful to derive non-asymptotic results allowing to apply the test to small sample sizes.

References

Baey C, Cournède PH, Kuhn E (2019). "Asymptotic Distribution of Likelihood Ratio Test Statistics for Variance Components in Nonlinear Mixed Effects Models." *Computational Statistics & Data Analysis*, **135**, 107–122. `doi:10.1016/j.csda.2019.01.014`.

Baey C, Kuhn E (2023). ***varTestnlme****: Variance Components Testing for Linear and Nonlinear Mixed Effects Models*. R package version 1.3.5, URL `https://CRAN.R-project.org/package=varTestnlme`.

Baey C, Mathieu A, Jullien A, Trevezas S, Cournède PH (2018). "Mixed-Effects Estimation in Dynamic Models of Plant Growth for the Assessment of Inter-Individual Variability." *Journal of Agricultural, Biological and Environmental Statistics*, **23**, 208–232. `doi:10.1007/s13253-017-0307-4`.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. `doi:10.18637/jss.v067.i01`.

Chen Z, Dunson DB (2003). "Random Effects Selection in Linear Mixed Models." *Biometrics*, **59**(4), 762–769. `doi:10.1111/j.0006-341x.2003.00089.x`.

Comets E, Lavenu A, Lavielle M (2017). "Parameter Estimation in Nonlinear Mixed Effect Models Using **saemix**, an R Implementation of the SAEM Algorithm." *Journal of Statistical Software*, **80**(3), 1–41. `doi:10.18637/jss.v080.i03`.

Crainiceanu CM, Ruppert D (2004). "Likelihood Ratio Tests in Linear Mixed Models with One Variance Component." *Journal of the Royal Statistical Society B*, **66**(1), 165–185. `doi:10.1111/j.1467-9868.2004.00438.x`.

Davidian M, Giltinan DM (1995). *Nonlinear Models for Repeated Measurement Data*. Chapman & Hall/CRC, New York. `doi:10.1201/9780203745502`.

Delattre M, Kuhn E (2019). "Estimating Fisher Information Matrix in Latent Variable Models Based on the Score Function." *arXiv 1909.06094*, arXiv.org E-Print Archive. `doi:10.48550/arxiv.1909.06094`.

Delattre M, Lavielle M, Poursat MA (2014). "A Note on BIC in Mixed-Effects Models." *Electronic Journal of Statistics*, **8**(1), 456–475. `doi:10.1214/14-ejs890`.

Dumont C, Lestini G, Le Nagard H, Mentré F, Comets E, Nguyen TT, **PFIM** Group (2018). "**PFIM** 4.0, An Extended R Program for Design Evaluation and Optimization in Nonlinear Mixed-Effect Models." *Computer Methods and Programs in Biomedicine*, **156**, 217–229. `doi:10.1016/j.cmpb.2018.01.008`.

Groll A (2022). **glmmLasso***: Variable Selection for Generalized Linear Mixed Models by ℓ_1-Penalized Estimation*. R package version 1.6.2, URL `https://CRAN.R-project.org/package=glmmLasso`.

Groll A, Tutz G (2014). "Variable Selection for Generalized Linear Mixed Models by ℓ_1-Penalized Estimation." *Statistics and Computing*, **24**, 137–154. `doi:10.1007/s11222-012-9359-z`.

Gurka MJ (2006). "Selecting the Best Linear Mixed Model under REML." *The American Statistician*, **60**(1), 19–26. `doi:10.1198/000313006x90396`.

Halekoh U, Højsgaard S (2014). "A Kenward-Roger Approximation and Parametric Bootstrap Methods for Tests in Linear Mixed Models – The R Package **pbkrtest**." *Journal of Statistical Software*, **59**(9), 1–30. `doi:10.18637/jss.v059.i09`.

Hiriart-Urruty JB, Malick J (2012). "A Fresh Variational-Analysis Look at the Positive Semidefinite Matrices World." *Journal of Optimization Theory and Applications*, **153**(3), 551–577. `doi:10.1007/s10957-011-9980-6`.

Ibrahim JG, Zhu H, Garcia RI, Guo R (2011). "Fixed and Random Effects Selection in Mixed Effects Models." *Biometrics*, **67**(2), 495–503. `doi:10.1111/j.1541-0420.2010.01463.x`.

ICON plc (2022). **NONMEM** *7.5: Nonlinear Mixed Effects Modelling*. Dublin. URL `https://www.iconplc.com/innovation/nonmem/`.

Kung FH (1986). "Fitting Logistic Growth Curve with Predetermined Carrying Capacity." In *ASA Proceedings of the Statistical Computing Section*, pp. 340–343.

Kuznetsova A, Brockhoff P, Christensen R (2017). "**lmerTest** Package: Tests in Linear Mixed Effects Models." *Journal of Statistical Software*, **82**(13), 1–26. `doi:10.18637/jss.v082.i13`.

Lavielle M (2014). *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. Chapman & Hall/CRC, New York. `doi:10.1201/b17203`.

Lesnoff M, Laval G, Bonnet P, Abdicho S, Workalemahu A, Kifle D, Peyraud A, Lancelot R, Thiaucourt F (2004). "Within-Herd Spread of Contagious Bovine Pleuropneumonia in Ethiopian Highlands." *Preventive Veterinary Medicine*, **64**(1), 27–40. `doi:10.1016/j.prevetmed.2004.03.005`.

Lixoft (2021). **Monolix** *2021R2*. Antony. URL `https://lixoft.com/products/monolix/`.

Millard SP (2013). **EnvStats**: *An* R *Package for Environmental Statistics*. Springer-Verlag, New York. `doi:10.1007/978-1-4614-8456-1`.

Molenberghs G, Verbeke G (2007). "Likelihood Ratio, Score, and Wald Tests in a Constrained Parameter Space." *The American Statistician*, **61**(1), 22–27. `doi:10.1198/000313007x171322`.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2022). **nlme**: *Linear and Nonlinear Mixed Effects Models*. R package version 3.1-161, URL `https://CRAN.R-project.org/package=nlme`.

Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in* S *and* S-PLUS. Springer-Verlag, New York. `doi:10.1007/b98882`.

Potthoff RF, Roy SN (1964). "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems." *Biometrika*, **51**(3-4), 313–326. `doi:10.1093/biomet/51.3-4.313`.

R Core Team (2023). R: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. URL `https://www.R-project.org/`.

Säfken B, Rügamer D, Kneib T, Greven S (2021). "Conditional Model Selection in Mixed-Effects Models with **cAIC4**." *Journal of Statistical Software*, **99**(8), 1–30. `doi:10.18637/jss.v099.i08`.

SAS Institute Inc (2020). *The* SAS *System, Version 15.2*. SAS Institute Inc., Cary. URL `https://www.sas.com/`.

Scheipl F, Greven S, Küchenhoff H (2008). "Size and Power of Tests for a Zero Random Effect Variance or Polynomial Regression in Additive and Linear Mixed Models." *Computational Statistics & Data Analysis*, **52**(7), 3283–3299. `doi:10.1016/j.csda.2007.10.022`.

Shapiro A (1985). "Asymptotic Distribution of Test Statistics in the Analysis of Moment Structures under Inequality Constraints." *Biometrika*, **72**, 133–144. `doi:10.1093/biomet/72.1.133`.

Silvapulle MJ, Sen PK (2011). *Constrained Statistical Inference: Order, Inequality, and Shape Constraints*, volume 912. John Wiley & Sons.

Stram DO, Lee JW (1994). "Variance Components Testing in the Longitudinal Mixed Effects Model." *Biometrics*, **50**(4), 1171–1177. `doi:10.2307/2533455`.

Stram DO, Lee JW (1995). "Corrections to Variance Components Testing in the Longitudinal Mixed Effects Model." *Biometrics*, **51**(3), 1196. `doi:10.2307/2533038`.

Turlach BA, Weingessel A, Moler C (2019). **quadprog**: *Functions to Solve Quadratic Programming Problems*. R package version 1.5-8, URL `https://CRAN.R-project.org/package=quadprog`.

Vaida F, Blanchard S (2005). "Conditional Akaike Information for Mixed-Effects Models." *Biometrika*, **92**(2), 351–370. `doi:10.1093/biomet/92.2.351`.

Varadhan R (2022). **alabama**: *Constrained Nonlinear Optimization*. R package version 2022.4-1, URL `https://CRAN.R-project.org/package=alabama`.

Wang T, Merkle EC (2018). "**merDeriv**: Derivative Computations for Linear Mixed Effects Models with Application to Robust Standard Errors." *Journal of Statistical Software, Code Snippets*, **87**(1), 1–16. `doi:10.18637/jss.v087.c01`.

Wood S (2017). *Generalized Additive Models: An Introduction with R*. 2nd edition. Chapman & Hall/CRC.

Wood S, Scheipl F (2020). **gamm4**: *Generalized Additive Mixed Models Using* **mgcv** *and* **lme4**. R package version 0.2-6, URL `https://CRAN.R-project.org/package=gamm4`.

Affiliation:

Charlotte Baey
Université de Lille, CNRS, UMR 8524
Laboratoire Paul Painlevé
F-59000 Lille, France
E-mail: `charlotte.baey@univ-lille.fr`
URL: `https://baeyc.github.io/`

Estelle Kuhn
Université Paris Saclay
INRAE unité MaIAGE
Domaine de Vilvert
78352 Jouy-en-Josas, France
E-mail: estelle.kuhn@inrae.fr
URL: http://genome.jouy.inra.fr/~ekuhn/