lmerTest Package: Tests in Linear Mixed Effects Models

Kuznetsova, Alexandra; Brockhoff, Per B.; Christensen, Rune Haubo Bojesen

Published in:
Journal of Statistical Software

Link to article, DOI:
10.18637/jss.v082.i13

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Journal of Statistical Software

December 2017, Volume 82, Issue 13. *doi: 10.18637/jss.v082.i13*

lmerTest Package: Tests in Linear Mixed Effects Models

Alexandra Kuznetsova
Technical University
of Denmark

Per B. Brockhoff
Technical University
of Denmark

Rune H. B. Christensen
Technical University
of Denmark
Christensen Statistics

Abstract

One of the frequent questions by users of the mixed model function `lmer` of the **lme4** package has been: How can I get p values for the F and t tests for objects returned by `lmer`? The **lmerTest** package extends the '`lmerMod`' class of the **lme4** package, by overloading the `anova` and `summary` functions by providing p values for tests for fixed effects. We have implemented the Satterthwaite's method for approximating degrees of freedom for the t and F tests. We have also implemented the construction of Type I–III ANOVA tables. Furthermore, one may also obtain the `summary` as well as the `anova` table using the Kenward-Roger approximation for denominator degrees of freedom (based on the `KRmodcomp` function from the **pbkrtest** package). Some other convenient mixed model analysis tools such as a `step` method, that performs backward elimination of non-significant effects – both random and fixed, calculation of population means and multiple comparison tests together with plot facilities are provided by the package as well.

Keywords: denominator degree of freedom, Satterthwaite's approximation, ANOVA, R, linear mixed effects models, **lme4**.

1. Introduction

Linear mixed effects models are tools for modeling continuous correlated hierarchical/multi-level data. During the last decades these models have become more and more prominent in a variety of fields such as the physical, biological and social sciences. Various software packages, commercial as well as open-source, are capable of fitting these types of models. The focus of this paper is on the open-source R package **lme4** (Bates, Mächler, Bolker, and Walker 2015). This package is a well-known and widely used R package designed to fit linear as well as non-linear mixed effects models. Some of the **lme4** package main strengths are the user-friendly

interface, the ability to handle unbalanced data, multiple crossed effects and being very fast even for large data sets.

The `anova` and `summary` functions are two of the main functions providing inference on the parameters of a model. In tests for the fixed effects of a linear mixed effect model, the F-statistics `anova` and the t-statistics `summary` functions are given, though p values for the corresponding F and t tests are not provided by the **lme4** package. The reason is connected with the fact that generally the exact null distributions for the parameter estimates and test statistics are unknown. So the only way to judge about the significance of the effects is by some sort of approximation and/or simulation based approach. A common way is to use the likelihood ratio test (LRT). This test is fast and is available in the **lme4** package. The downside is that it can produce anti-conservative p values in a variety of situations, which we discuss in Section 3. A simulation based alternative is the `bootMer` function from the **pbkrtest** package (Halekoh and Højsgaard 2014, 2017), which is computationally intensive. The authors of the **pbkrtest** package have implemented the Kenward-Roger's approximation method, which provides accurate p values, but for some types of models and large data the method could be computationally intensive. Our aim was to provide a method, that is a nice alternative to the widely used LRT. We have implemented Satterthwaite's method (Giesbrecht and Burns 1985; Fai and Cornelius 1996) as implemented in the SAS software package (SAS Institute Inc. 1978, 2013) and wrapped it into `anova` and `summary` functions for an object returned by `lmer`. We have also integrated the Kenward-Roger's approximation method through the `KRmodcomp` function of the **pbkrtest** package. Hence, there are two available alternatives for the `anova` and `summary` methods.

Another contribution of the package is a generation of the three types of ANOVA hypothesis contrast matrices (SAS Institute Inc. 1978) that result in producing the corresponding types of ANOVA tables. Type II and III may be also obtained through the `Anova` function of the **car** package (Fox and Weisberg 2011). However, some limitations can be found. For instance, sum-to-zero restrictions on parameters should be used in order to get the correct Type III ANOVA table. In our implementation the generation of the three types of the ANOVA tables is invariant with respect to the restrictions used on the parameters of the linear mixed model.

Some other convenience functions such as the `step` function, that performs automated elimination of non-significant effects, the `lsmeansLT` and `difflsmeans` functions, that generate the least squares means and the differences of least squares means tables with confidence intervals are provided by the **lmerTest** package. The functions contained in the **lmerTest** package are listed in Table 1.

The paper is structured in the following way: in Sections 2 and 3 we describe the approach taken by Giesbrecht and Burns (1985); Fai and Cornelius (1996) to address the inference problem and compare the approximation methods to the commonly used LRT. In Section 4 two of the data sets from the **lmerTest** package are introduced. In Section 5 we discuss different types of hypothesis for ANOVA and their implementation in the **lmerTest** package. In Section 6 we introduce least squares means. In Section 7 we introduce our implementation of the step-down model building approach. In Section 8 we describe the methods contained in the package. In Section 9 we discuss the timing issues for approximation methods for a certain class of linear mixed effects models. Section 10 contains discussion and conclusion.

Functionalities	`anova`	`summary`	`rand`	`step`	`lsmeansLT`	`difflsmeans`
Output as from **lme4**	✓	✓				
ANOVA-like table for the random effects (LRT)			✓	✓		
Satterthwaite's approximation to degrees of freedom	✓	✓		✓	✓	✓
Kenward-Roger's approximation to degrees of freedom	✓	✓		✓		
Type I, II, III hypothesis tests (SAS notations)	✓			✓		
Least squares means				✓	✓	
Differences of least squares means				✓		✓
Automated elimination of random and/or fixed effects				✓		

Table 1: Summary of the functions provided by the **lmerTest** package.

2. Inference and test statistic

A linear mixed model can be specified in matrix form as:

$$y = X\beta + Zu + \varepsilon, \quad u \sim N_q(0, G), \quad \varepsilon \sim N_n(0, R), \tag{1}$$

with β representing all fixed-effects parameters, u the random effects, X the $n \times p$ design matrix for the fixed-effects parameters, and Z the $n \times q$ design matrix for the random effects, u and ε are independent and $R = \sigma^2 I$.

To test a hypothesis about the fixed effects β, one may use the LRT. Then a smaller model needs to be constructed with the same error structure as model (1):

$$y_0 = X_0\beta_0 + Zu + \varepsilon. \tag{2}$$

The LRT statistic for the test of the hypothesis

$$H_0 : \beta \in \Theta_{\beta_0},$$
$$H_1 : \beta \in \Theta_{\beta},$$

where Θ_{β_0} is a subspace of the parameter space Θ_β of the fixed effects β:

$$T = 2(ll - ll_0),$$

where ll and ll_0 represent the log-likelihoods of models in Equations 1 and 2 accordingly. Under the null hypothesis, T follows asymptotically a χ^2 distribution. Even though LRT is frequently used, it can produce anti-conservative p values (Pinheiro and Bates 2000).

One may consider an F test of the hypothesis $H_0 : L\beta = 0$, where L is a contrast matrix of $q = \text{rank}(L) > 1$. A test statistic for this hypothesis is:

$$F = \frac{(L\hat{\beta})^\top (L\hat{C}L^\top)^{-1}(L\hat{\beta})}{q}, \tag{3}$$

where $\hat{C}$ is an estimated variance-covariance matrix of $\hat{\beta}$. Even though the statistic is called F, in general it does not exactly follow an F distribution. A method, known as Satterthwaite method, was proposed by Fai and Cornelius (1996) for determining denominator degrees of freedom ν such that: $F \sim F_{q,\nu}$ approximately. We have implemented their work for the F test and also for a one-degree of freedom test, which corresponds to the t test with the method proposed by Giesbrecht and Burns (1985). The details of the algorithm are given in Appendix A. In Kenward-Roger's method the estimated variance-covariance matrix $\hat{C}$ is adjusted in order to improve the small sample distributional properties of F and then the Satterthwaite's method-of-moment of approximation is applied. The algorithm may be found in Halekoh and Højsgaard (2014).

3. Comparisons of F tests and LR tests

As previously mentioned, the LRT can produce anti-conservative p values. This may occur when the data is unbalanced or when the number of parameters is large compared to the number of observations (Pinheiro and Bates 2000, p. 88). In Halekoh and Højsgaard (2014) an example where LRT leads to misleading results and where Kenward-Roger's method is accurate is given.

Pinheiro and Bates (2000) provide a simulation study for the LRT based on the `PBIB` data. The `PBIB` data comes from the **SASmixed** package (Littell *et al.* 2014) and is an example of a partially balanced incomplete block experiment with $i = 1, \ldots, 15$ treatments, $j = 1, \ldots, 15$ blocks and 60 observations. Not every level of treatment appears with every level of blocking factor, but every pair of treatments occur together in a block the same number of times. Pinheiro and Bates (2000) consider the following mixed effects model for this data:

$$y_{ijk} = \alpha_i + b_j + \epsilon_{ijk}, \quad b_j \sim N(0, \sigma_b^2), \text{ and } \epsilon_{ijk} \sim N(0, \sigma^2), \tag{4}$$

where α stands for a treatment effect, b stands for a random block effect.

In order to compare LRT to the F test with Satterthwaite and Kenward-Roger approximation methods we performed a simulation study for a test for a presence of the treatment effect. We performed 1000 simulations from the model with only a random block effect corresponding to the null hypothesis of no treatment effect. The results of the simulations are presented in Figure 1. It can be seen that the LRT gives for all nominal values anti-conservative p values. It is also clear that both Satterthwaite's and Kenward-Roger's methods p values are close to the nominal values.

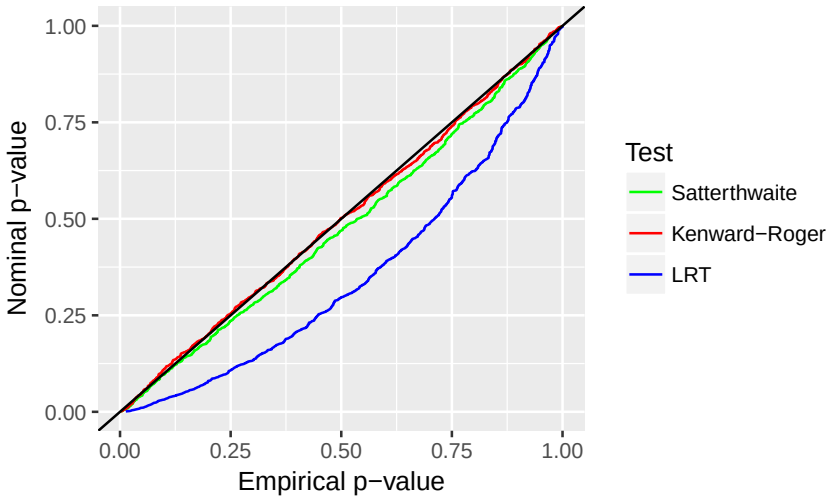

Figure 1: Empirical p values versus nominal p values ranging from 0.001 to 1 for the test of the presence of the treatment fixed effect. The results are based on 1000 simulations from the model with a random block effect applied to the `PBIB` data.

4. Data sets

Package **lmerTest** includes three data sets from sensory and consumer studies. Throughout the paper we will use two of them: the first one with the name `TVbo` comes from a sensory study and consists of tests of TV sets produced by the high-end HIFI company Bang and Olufsen A/S, Struer, Denmark. The second data set is a combination of a sensory and a consumer study and has the name `carrots`.

4.1. The `TVbo` data

The main purpose in this study was to assess 12 products, specified by two features: `Picture`, a factor with 4 levels and `TVset`, a factor with 3 levels. All in all 12 products in 2 replications were assessed by 8 trained panelists (`Assessor`) for 15 different response variables on a scale from 1 to 14. This type of data is very common in sensory science (Lawless and Heymann 2010).

For illustration, let us select the attribute `Sharpnessofmovement` as our response variable. We consider the `Assessor` effect as random since it is generally regarded as the proper approach in the sensory field (Lawless and Heymann 2010). In the fixed part of the model we include `TVset` and `Picture` effects and their interaction. In the random part we also include interaction effects `Assessor:TVset` and `Assessor:Picture`. The choice of including these effects will be later justified in Section 8.4.

A linear mixed effects model for the `Sharpnessofmovement` attribute is then:

$$y_{ijkr} = \alpha_i + \beta_j + \gamma_{ij} + c_k + ac_{ik} + bc_{jk} + \epsilon_{ijkr}, \tag{5}$$
$$c_k \sim N(0, \sigma_c^2), \quad ac_{ik} \sim N(0, \sigma_{ac}^2), \quad bc_{jk} \sim N(0, \sigma_{bc}^2) \text{ and } \epsilon_{ijkr} \sim N(0, \sigma^2),$$

with $i = 1, 2, 3$; $j = 1, 2, 3, 4$; $k = 1, \ldots, 8$; $r = 1, 2$, and where α stands for the `TVset` effect, β for the `Picture` effect, γ stands for the interaction effect `TVset:Picture`, c stands for `Assessor` effect.

4.2. The `carrots` data

The `carrots` data comes from The Royal Veterinary and Agricultural University, Denmark and is an example of a so-called external preference mapping. 103 consumers scored their preference of 12 danish carrot types on a scale from 1 to 7. In addition to the consumer survey, the carrot products were profiled by a trained panel of tasters, the sensory panel, with respect to a number of sensory properties (taste, odor and texture). The goal was to relate the sensory properties of the products to the consumer liking. Since there was a high number of sensory properties (14), a principal component analysis was performed and the first two principle components were extracted that contained most of the information in the sensory properties (`sens1` and `sens2`). `sens1` mainly measured bitterness versus nutty taste, whereas `sens2` measured mainly sweetness. A common method for preference mapping is to fit regression models for the preference as a function of the sensory variables for each individual consumer using the 12 observations across the carrot products. Next, the individual regression coefficients are investigated in an exploratory manner. Another approach, we will use in the paper, is to use a mixed effects model, where consumers and products are treated as random effects. The `product` effect is also considered as random since we wish to consider the entire population of carrot products instead of only the 12 specific products investigated in this experiment. The following linear mixed effects model can then be considered:

$$y_{ijk} = b_{0j} + \beta_0 + (b_{2j} + \beta_2)\texttt{sens2}_{ij} + (b_{1j} + \beta_1)\texttt{sens1}_{ij} + c_k + \epsilon_{ijk} \tag{6}$$

where β_0, β_1 and β_2 stand for fixed intercept and two slopes, b_0, b_1 and b_2 stand for random intercept and random slopes, c stands for `product` effect. We assume the following covariance structure:

$$(b_0, b_1, b_2) \sim N(0, \begin{pmatrix} \sigma_0^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{01} & \sigma_1^2 & \sigma_{12} \\ \sigma_{02} & \sigma_{12} & \sigma_2^2 \end{pmatrix}),\ c \sim N(0, \sigma_c^2),\ \epsilon_{ijk} \sim N(0, \sigma^2).$$

5. Types of hypothesis tests

Type I, II and III ANOVA tables as defined in the SAS software SAS Institute Inc. (1978) are provided by the **lmerTest** package. The Type I ANOVA table performs the sequential decomposition of the contributions of the fixed-effects and is the one produced by the `anova` method of the **lme4** package. The Type I table is order dependent compared to the Type II and III tables, which do not depend on the order in which the effects are entered in the model. In terms of the hypothesis tests, the three types are the same in balanced cases, where number of observations (experimental units) at each factor-level combination are equal.

For illustration, let us consider the `TVbo` data and the model in Equation 5. Since the `TVbo` data is balanced all the types produce the same tests. Following Searle (1987) the hypothesis test for the interaction effect γ is the following one:

$$\gamma_{i'j'} - \gamma_{ij'} - \gamma_{i'j} + \gamma_{ij} = 0, \qquad \forall i, i', j, j'. \tag{7}$$

The hypothesis test for the main α effect is the following one:

$$\alpha_i - \alpha_{i'} + (1/4)\sum_j (\gamma_{ij} - \gamma_{i'j}) = 0, \qquad \forall i, i', \tag{8}$$

which is easy to interpret, namely the test for the effect of the `TVset` factor averaged over all levels of the `Picture` factor is performed. In the unbalanced cases the tests for the higher order terms are still the same, whereas for the lower-order terms the hypotheses differ between the types. For example, if for some reason some observations were missing in the `TVbo` data, the Types I and II hypotheses for the main α effect would no longer produce the test from Equation 8. In unbalanced situations the Types I and II hypotheses become dependent on the number of observations (experimental units) at each factor-level combination, so the hypotheses for these types become hard to interpret (Searle 1987). On the contrary, the Type III hypothesis test is the same whether the data is balanced or not, so the test for the α effect would still be the one from Equation 8. In situations where there are missing cells (some factors, combinations of factors are missing) the Type III hypotheses may loose their simple interpretation. A warning is given in the **lmerTest** package that care must be taken in interpreting the tests.

There have been many debates regarding which type of ANOVA table is the most appropriate and when. We do not touch this topic here and refer to Venables (2000); Speed, Hocking, and Hackney (1978); Senn (2007); Langsrud (2003); Macnaughton (2009) for the discussions. In the **lmerTest** package instead we provide a tool for obtaining the three types of ANOVA tables for the objects returned by `lmer`, which are implemented via calculation of the appropriate hypothesis contrast matrix L in Equation 3. The algorithms for constructing the Types I–III L contrast matrices are given in Appendix B.

6. Least square means and differences of least square means

The least squares means (also called population means) were introduced by Harvey (1960). The least squares means are estimates of the class or subclass means that would be expected if there would have been equal subclass numbers.

For illustration let us again consider the `TVbo` data and the model for response variable `Sharpnessofmovement` in Equation 5. The expectation, for instance, for level i of `TVset` effect is:

$$\mathsf{E}(\overline{y}_{i\cdot}) = \mu + \alpha_i + 1/4 \sum_j (\beta_j + \gamma_{ij}). \tag{9}$$

The `TVbo` data is balanced, so the expectation is estimated by the corresponding mean: $\overline{y}_{i\cdot}$. In an unbalanced case, like, e.g., if some observations were missing from the data, the expectation is no longer estimated by the corresponding mean and Equation 9 is no longer valid. The least square means are then defined in a way that Equation 9 still holds even for unbalanced data.

Generally one is interested in testing the significance about the differences of least square means. In a linear mixed effects model specified in the following form: $\mathsf{E}(Y) = X\beta$ the null hypothesis of equality of difference of least squares means is

$$H_0 : l\beta = 0, \tag{10}$$

where l is a contrast vector. For instance, from Equation 9 the null hypothesis of equality of levels 1 and 2 for `TVset` factor is $H_0 : \alpha_1 - \alpha_2 + (1/4)\sum_j (\gamma_{1j} - \gamma_{2j}) = 0$. The t-statistic for

the hypothesis in Equation 10 is then:

$$t = \frac{l\hat{\beta}}{\sqrt{l\hat{C}l^\top}}, \tag{11}$$

where $\hat{C}$ is an estimated variance-covariance matrix of $\hat{\beta}$. Generally the t-statistic does not follow a t distribution. Giesbrecht and Burns (1985) proposed a method for determining a t-distribution that approximates the distribution of t under the null hypothesis based on Satterthwaite's method-of-moment approximation to the degrees of freedom. We have implemented their work, the algorithm is in Appendix A. The confidence intervals are then computed using the following formula:

$$CI = l\hat{\beta} \pm t_{\frac{\alpha}{2}}(\nu) \cdot \sqrt{l\hat{C}l^\top},$$

where ν is calculated using the Satterthwaite's method of approximation.

The `lsmeansLT` and `difflsmeans` functions from the **lmerTest** package produce least square means and differences of least square means accordingly with 95% confidence intervals for all factors, that are part of an object returned by `lmer`. The construction of l vectors for the least square means uses the `popMatrix` function from the **doBy** package (Højsgaard and Halekoh 2016). The l vectors for differences of least square means are then constructed as pairwise differences of ls vectors from the least square means. The l vectors for differences of least squares means are actually related to Type III contrasts and are equivalent if there are no missing cells (SAS Institute Inc. 1978). There is no multiplicity correction for the multiple comparison tests in the **lmerTest** package, one can use the `p.adjust` function from the **stats** package in order to correct the p values. The l vectors are checked for estimability (Searle 1997) using the package **estimability** (Lenth 2016b).

7. Step-down model-building approach

A practical data-driven approach suggested in Zuur, Ieno, Walker, Saveliev, and Smith (2009) and Diggle (2002) is a step-down strategy. The strategy is based on construction of a maximal possible model followed by deletion of effects with high p values obeying the principle of marginality. In the **lmerTest** package we have implemented a `step` function that automates the step-down approach. An outline of the algorithm is given here:

Step 1: Simplification of the random effects structure.

1. Let M be the linear mixed effects model specified by a user.
2. If there are random effects in M then go to 3, otherwise stop.
3. For each random effect r_i in M do:
   (a) Create a reduced model M_i by eliminating r_i from M.
   (b) Calculate p_i, the p value from the likelihood ratio test of comparing M to M_i.
   (c) Save p_i and M_i.
4. Find $p_{\max}$; the maximum of all p_i and let $M_{\max}$ denote the corresponding model.
5. Set M to $M_{\max}$. If $p_{\max}$ is higher than α level then go back to 3, otherwise stop.

If the initial model is a random-coefficient model, then the principle of simplification of the random effects is similar – the effect that contains slopes and intercept and correlation between them is incrementally reduced by removing first non-significant slopes and then non-significant intercepts. So when the effect is eliminated then the relevant correlations are eliminated as well. In Appendix C an example illustrating the process of the simplification of an error structure in random coefficient models is given.

Step 2: Simplification of the fixed-effects structure.

1. Consider M, the output model from **Step 1**.
2. Construct an ANOVA table for M, calculate F statistics and p values for each fixed-effects term.
3. Consider the highest order interaction effects in M. The effect with the highest p value (p_{eff}) is identified and a model without this effect M_{eff} is constructed.
4. Set M_{eff} to M. If p_{eff} is less than α level or if there are no more fixed-effects then stop, otherwise go to 2.

Model M from **Step 2** is the final model selected by the algorithm.

The `llply` function of the **plyr** package (Wickham 2011) is used for calling the functions for testing random effects in **Step 1** and fixed effects in **Step 2**. The use of the `llply` function has a computational advantage compared to the `lapply` function from the **base** package (R Core Team 2017).

The `step` method of the **lmerTest** package contains arguments that make the step-down approach flexible. For instance, by setting the argument `reduce.random` to `FALSE` **Step 1** can be omitted. Similarly, by setting the argument `reduce.fixed` to `FALSE` **Step 2** can be omitted. One may specify which effects should be part of the model anyways by specifying the names of the terms in the `keep.effs` argument. For example, in the `TVbo` data it may be natural to retain the `Assessor` effect in the model even if the effect is not significant. By default the α level in tests for the fixed effects is 0.05 and the α level in tests for the random effects is 0.1. However, both α levels can be easily changed.

8. Application of the methods

8.1. The 'merModLmerTest' class

In the **lmerTest** package we specify a new class with the name '`merModLmerTest`', which contains the '`lmerMod`' class from the **lme4** package:

```
R> merModLmerTest <- setClass("merModLmerTest",
+    contains = c("merMod", "lmerMod"))
```

So if the **lmerTest** package is loaded, then the models specified with the `lmer` function are coming from the '`merModLmerTest`' class and not '`lmerMod`'. Then we define the `summary` and `anova` methods for the '`merModLmerTest`' class, which are the extensions of the `summary` and `anova` methods of the '`lmerMod`' class. The nice feature about the '`merModLmerTest`' class is

that all the methods provided by the **lme4** package for the objects returned by `lmer` are also available for the '`merModLmerTest`' class. This means that by loading the **lmerTest** package and by specifying the model with the `lmer` function the users of the **lme4** package get all the methods, provided by the **lme4** package plus extensions to the `summary` and `anova` methods and additional ones such as `calcSatterth`, `step`, `lsmeansLT` and `difflsmeans`.

8.2. The `anova` method for objects returned by `lmer`

Let us now consider the `TVbo` data. The `lmer` call to fit the model in Equation 5 is:

```
R> tv <- lmer(Sharpnessofmovement ~ TVset * Picture + (1 | Assessor) +
+    (1 | Assessor:TVset) + (1 | Assessor:Picture), data = TVbo)
```

With the following call we obtain an ANOVA table that comes from the **lme4** package:

```
R> anova(tv)

Analysis of Variance Table
             Df Sum Sq Mean Sq F value
TVset         2  1.765  0.8825  0.2437
Picture       3 51.857 17.2857  4.7735
TVset:Picture 6 90.767 15.1279  4.1777
```

Now let us attach the **lmerTest** package and run again model `tv` and then apply the `anova` method again:

```
R> library("lmerTest")
R> tv <- lmer(Sharpnessofmovement ~ TVset * Picture + (1 | Assessor) +
+    (1 | Assessor:TVset) + (1 | Assessor:Picture), data = TVbo)
R> anova(tv)

Analysis of Variance Table of type III  with  Satterthwaite
approximation for degrees of freedom
             Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
TVset         1.765  0.8825     2    14  0.2437 0.7869818
Picture      51.857 17.2857     3    21  4.7735 0.0108785 *
TVset:Picture 90.767 15.1279     6   138  4.1777 0.0006845 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We may notice that two additional columns are added with the names `DenDF` and `Pr(>F)` referring to denominator degrees of freedom and p values, which are calculated using the Satterthwaite's method of approximation. According to the p values the interaction effect is highly significant, which means that the products differ for the `Sharpnessofmovement` attribute. More than that the products differ mostly due to the `Picture` feature. We may also notice that by default the **lmerTest** package provides the Type III ANOVA table, **lme4** provides the sequential (Type I) ANOVA table. One can require another type of ANOVA by changing the `type` argument, as well as require Kenward-Roger's method for calculating the F test. For instance, the Type II ANOVA table with Kenward-Roger's approximation can be obtained by calling:

```
R> anova(tv, type = 2, ddf = "Kenward-Roger")

Analysis of Variance Table of type II  with  Kenward-Roger
approximation for degrees of freedom
              Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
TVset          1.765  0.8825     2    14  0.2437 0.7869818
Picture       51.857 17.2857     3    21  4.7735 0.0108785 *
TVset:Picture 90.767 15.1279     6   138  4.1777 0.0006845 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case all types of hypotheses are identical since the `TVbo` data is balanced.

8.3. The `summary` method for objects returned by `lmer`

The `summary` method for objects returned by `lmer` in the **lmerTest** package produces an extended output of the `summary` method from the **lme4** package. The extension of the output consists of degrees of freedom using the Satterthwaite's (Kenward-Roger's) approximations for the t test and corresponding p values. To illustrate the `summary` method we consider the `carrots` data. We specify the model in Equation 6 using the **lme4** syntax:

```
R> m.carrots <- lmer(Preference ~ sens1 + sens2 +
+    (1 + sens1 + sens2 | Consumer) + (1 | product), data = carrots)
```

Now let us look at the summary of the model:

```
R> summary(m.carrots)

Linear mixed model fit by REML t-tests use Satterthwaite
  approximations to degrees of freedom [lmerMod]
Formula:
Preference ~ sens1 + sens2 + (1 + sens1 + sens2 | Consumer) +
    (1 | product)
   Data: carrots

REML criterion at convergence: 3739.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6194 -0.5306  0.0190  0.6103  2.9309

Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 Consumer (Intercept) 0.2095136 0.45773
          sens1       0.0002517 0.01586  -0.16
          sens2       0.0030473 0.05520   0.12  0.96
 product  (Intercept) 0.0335564 0.18318
```

```
 Residual             1.0335816 1.01665
Number of obs: 1233, groups:  Consumer, 103; product, 12

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept)  4.79911    0.07529 20.72200  63.740  < 2e-16 ***
sens1        0.01083    0.01503  9.16800   0.721  0.48913
sens2        0.07065    0.01728 10.94400   4.089  0.00181 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr) sens1
sens1 -0.010
sens2  0.023  0.032
```

The output is exactly the same as from the **lme4** package but with additional columns added to the fixed effects: `df` and `Pr(>|t|)`. `df` refers to degrees of freedom based on the Satterthwaite's approximation and `Pr(>|t|)` is the p value for the t test with `df` as degrees of freedom. We may conclude that the intercept and the slope for `sens2` are highly significant, so consumers prefer more sweet carrots. `sens1` has no significant impact on consumer preferences.

By setting argument `ddf` in the `summary` method to `"Kenward-Roger"` one may obtain the Kenward-Roger's approximation.

The calculation using the Satterthwaite approximation took around one second compared to the Kenward-Roger's which took around 16 seconds. The p values were identical up to the fourth digit for both approximations.

8.4. The `step` method for objects returned by `lmer`

Let us consider again the `TVbo` data with the same response variable `Sharpnessofmovement`, but here we choose a different initial model than in Equation 5. Here we also include the `Repeat` effect as a random effect and consider a full model, where both random and fixed structures contain all possible main and interaction effects.

$$y_{ijklm} = \alpha_i + \beta_j + \alpha\beta_{ij} + c_k + ac_{ik} + bc_{jk} + abc_{ijk} + d_l + ad_{il} + bd_{jl} + abd_{ijl} + \epsilon_{ijklm},$$
$$c_k \sim N(0, \sigma_c^2)\ ,\ bc_{jk} \sim N(0, \sigma_{bc}^2), ac_{ik} \sim N(0, \sigma_{ac}^2)\ ,\ abc_{ijk} \sim N(0, \sigma_{abc}^2),$$
$$d_l \sim N(0, \sigma_d^2)\ ,\ bd_{jl} \sim N(0, \sigma_{bd}^2), ad_{il} \sim N(0, \sigma_{ad}^2)\ ,\ abd_{ijl} \sim N(0, \sigma_{abd}^2) \text{ and } \epsilon_{ijklm} \sim N(0, \sigma^2),$$

where α stands for the `TVset` effect, β for the `Picture` effect, c stands for the `Assessor` effect, d stands for the `Repeat` effect. The corresponding model using `lmer` is:

```
R> tv <- lmer(Sharpnessofmovement ~ TVset * Picture +
+    (1 | Assessor:TVset) + (1 | Assessor:Picture) +
+    (1 | Assessor:Picture:TVset) + (1 | Repeat) + (1 | Repeat:Picture) +
+    (1 | Repeat:TVset) + (1 | Repeat:TVset:Picture) + (1 | Assessor),
+    data = TVbo)
```

	χ^2	χ^2 df	Elim. num.	p value
`Assessor:Picture:TVset`	0.00	1	1	1.00000
`Repeat:TVset:Picture`	0.00	1	2	1.00000
`Repeat`	0.00	1	3	1.00000
`Repeat:Picture`	0.00	1	4	1.00000
`Repeat:TVset`	0.00	1	5	1.00000
`Assessor:TVset`	2.79	1	kept	0.09491
`Assessor:Picture`	12.35	1	kept	< 0.001
`Assessor`	7.47	1	kept	0.00627

Table 2: Likelihood ratio tests for the random effects and their order of elimination representing Step 1 of the automated analysis for the `TVbo` data for attribute `Sharpnessofmovement`.

	Sum Sq	Mean Sq	Num. df	Den. df	F value	Elim. num.	`Pr(>F)`
`TVset`	1.76	0.88	2	14.00	0.24	kept	0.7870
`Picture`	51.86	17.29	3	21.00	4.77	kept	0.0109
`TVset:Picture`	90.77	15.13	6	138.00	4.18	kept	< 0.001

Table 3: F tests for the fixed-effects and their order of elimination representing Step 3 of the automated analysis for the `TVbo` data for attribute `Sharpnessofmovement`.

Then we apply the `step` and save the results in a variable `st`:

```
R> st <- step(tv)
```

One may apply the `print` method on the `st` variable to view the results. Here instead we wrap the output into an '`xtable`' object of the **xtable** package (Dahl 2016) in order to nicely represent the results in the paper.

Tables 2 and 3 represent the **Step 1** and **Step 2** of the step-down model building approach in Section 7. The effects that have been kept according to the elimination column denoted by "elim. num." are the ones that form the final reduced model given by the default type I levels ($\alpha = 0.1$ for the random effects and $\alpha = 0.05$ for the fixed effects).

From Table 2 it is seen that five random effects were eliminated. The `Repeat` effect is not part of the final reduced model. From Table 3 it is seen that the interaction effect `TVset:Picture` is significant, so the main effects are kept in the model according to the principle of marginality. We observe that indeed the simplified model is the one from Equation 5.

Least squares means and differences of least squares means tables are also part of the output from the `step` function. Here we visualize the tables in barplots by applying the `plot` function on the `st` object. Since there are too many levels in the `TVset:Picture` effect, the plot is hard to understand and thus we ask to plot the barplots only for the `Picture` and `TVset` effects in the following way (see Figure 2):

```
R> plot(st, effs = c("Picture", "TVset"))
```

The resulting plot is shown in Figure 2. The plot for the `Picture` effect shows that the most different product with respect to the `Picture` feature for the attribute `Sharpnessofmovement` is the one with level `4`. Since the `TVset` effect is non-significant according to Table 3, there

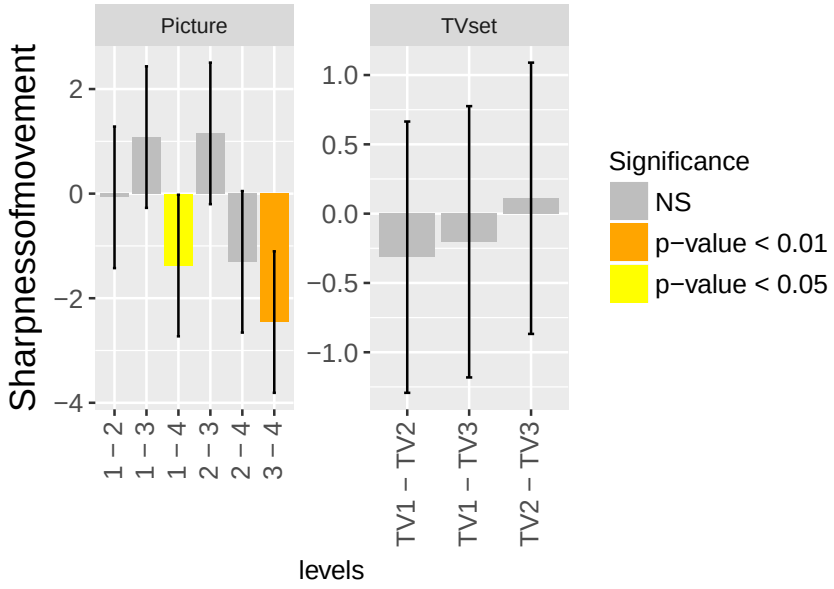

Figure 2: Barplots for differences of least square means for `TVset` and `Picture` effects together with 95% confidence intervals for the `TVbo` data.

are no significant differences between the levels of this effect. The **ggplot2** package (Wickham 2009) is used in the **lmerTest** package for generating the barplots for the least square means and differences of least square means. The **Hmisc** package (Harrell Jr 2017) is used for manipulating the strings in the construction of the least square means and difference of least square means tables.

There are 15 attributes in the `TVbo` data, so 14 more models should be constructed and analyzed similarly to the model for the `Sharpnessofmovement` attribute considered in this example. Constructing models and applying the `step` function in a loop is therefore a useful and fast tool for getting insight into the data. More examples where the usefulness of the `step` function is illustrated are given in Kuznetsova, Christensen, Bavay, and Brockhoff (2015).

8.5. Miscellaneous functions

We have also included a function called `calcSatterth` to perform F tests with the Satterthwaite's approximation to degrees of freedom for a user specified contrast matrix L. For example, the test for the `TVset:Picture` interaction effect for model in Equation 5 could be obtained as follows:

```
R> L <- matrix(0, ncol = 12, nrow = 6)
R> L[1, 7] <- L[2, 8] <- L[3, 9] <- L[4, 10] <- L[5, 11] <- L[6, 12] <- 1
R> calcSatterth(tv, L)

$denom
[1] 138

$Fstat
         [,1]
```

```
[1,] 4.177655

$pvalue
             [,1]
[1,] 0.000684479

$ndf
[1] 6
```

It can be seen that the results agree with the results from the `anova` method in Section 8.2.

9. Computational timing issues

Halekoh and Højsgaard (2014) mention that the calculation of Kenward-Roger's approximation for some models might be computationally intensive. From our practice calculation of the Satterthwaite's approximation as implemented in the **lmerTest** package requires less time than the Kenward-Roger's as implemented in the **pbkrtest** package. The difference in timings depends on the size of the data and the type of the model. We have observed that for random coefficient models, the difference can be quite significant. Here we compare the computational time for the two methods (Kenward-Roger's and Satterthwaite's) using the `carrots` data and the same model set-up as in Equation 6. In order to compare the methods for different sizes of the data, we construct 5 data sets, that are extended versions of the `carrots` data. The extensions consist of replicating randomly selected rows from the `carrots` data. For instance, in the first data set we randomly select 1000 rows from the `carrots` data (with replacement) and then add these rows to the `carrots` data, so the size of the data becomes the size of the `carrots` data (1236 observations) plus 1000. In the following the code for constructing the data sets, fitting the models as in Equation 6 and calculating the time for the `anova` method applied to these models for two approximation methods is given:

```
R> size <- seq(0, 4000, by = 1000)
R> ind.size <- lapply(size, function(x)
+    sample(seq_len(nrow(carrots)), size = x, replace = TRUE))
R> dd <- lapply(ind.size, function(x) carrots[c(1:nrow(carrots), x), ])
R> fit.mcarrots <- function(d) {
+    lmer(Preference ~ sens1 + sens2 +
+      (1 + sens1 + sens2 | Consumer) + (1 | product), data = d)
+  }
R> m.carrots.list <- lapply(dd, fit.mcarrots)
R> time.sat <- lapply(m.carrots.list, function(x) system.time(anova(x))[1])
R> time.kr <- lapply(m.carrots.list, function(x)
+    system.time(anova(x, ddf = "Kenward-Roger"))[1])
```

Figure 3 visualizes `time.kr` and `time.sat`, which stand for the computational time in seconds for the Kenward-Roger's and Satterthwaite's approximation methods accordingly. It can be seen, for instance, that for the data with around 5000 observations Kenward-Roger's method took more than 300 seconds (around 5 minutes) compared to the Satterthwaite's that took around one second. In this example we considered data not exceeding 6000 observations. For

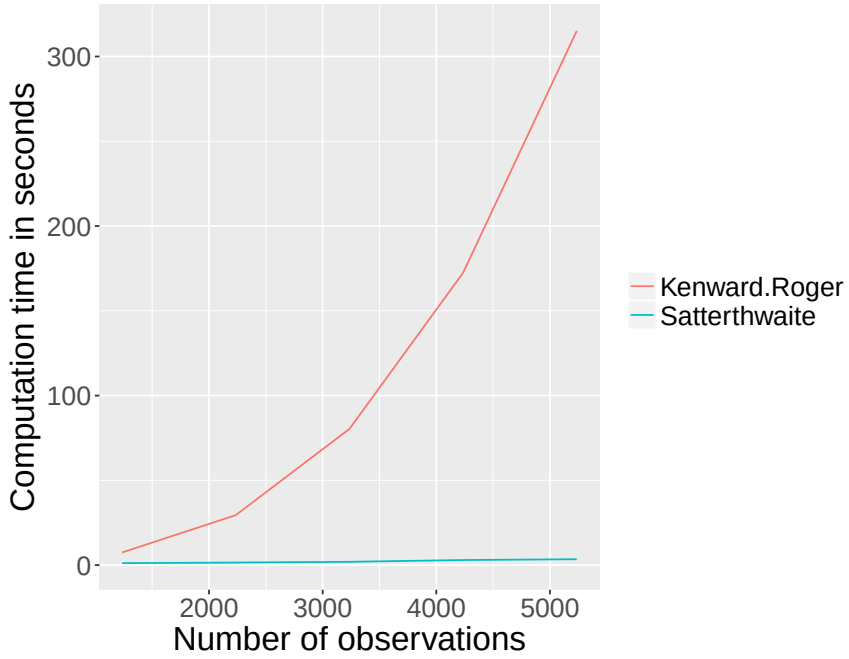

Figure 3: Differences in computational time between Kenward-Roger's and Satterthwaite's approximations for the random coefficient model as specified in Equation 6.

the data with around 10000 observations or more the `KRmodcomp` function threw the following error: `cannot allocate vector of size 957.7 Mb`. The comparisons in computational time were made using Windows 10 version 1703 (OS Build 15063.674) with a 64-bit version of R and the following hardware configuration: processor Intel(R) Core(TM) i3-5005U CPU 2.00GHz with 2 cores (4 threads) and 8 GB of memory. The comparisons were made with version 0.4-7 of the **pbkrtest** package.

10. Discussion and conclusion

In this paper we have presented our implementation of the Satterthwaite's method of approximation to one- and multi-degree of freedom tests. The Kenward-Roger's approximation, which is implemented in the **pbkrtest** package is also available as an option in the **lmerTest** package. Then it is up to the user to decide which approximation to use or whether to use any at all. From our practice, we observed that the p values that the approximation methods provide are generally very close to each other. Schaalje, McBride, and Fellingham (2002) performed a number of simulations in order to investigate the appropriateness of the approximation methods. They discovered that complexity of the covariance structures, sample size and imbalance affect the performance of both approximations. However, these factors affect the Satterthwaite's method more than the Kenward-Roger's. Still we believe that the Satterthwaite's method can be considered as a good alternative as it outperforms LRT in cases with unbalanced and/or small sample designs, generally is faster than Kenward-Roger's method and sometimes quite significantly faster. The reason that the LRT is so widely used is also connected with the fact that it is very easy and fast to use – just apply the `anova` method to two nested models. To maintain the user-friendliness we have wrapped the approximation

methods into the `anova` and `summary` methods. So now the users of the **lme4** package can get an extended version of these methods by simply loading the **lmerTest** package.

Another contribution of the package is a generation of the Type I–III ANOVA tables. By default the Type III ANOVA table is provided by package **lmerTest**. In terms of hypothesis tests this type is the easiest one to interpret both in unbalanced and balanced cases. Nevertheless, in different situations different types of ANOVA tables are advised (Speed *et al.* 1978; Senn 2007; Langsrud 2003; Macnaughton 2009).

We have also introduced the `step` function, which performs backward elimination of non-significant effects. In Kuznetsova *et al.* (2015) we have shown the usefulness of this tool in a number of situations in sensory and consumer studies. When used for the random part of the model, the step-approach can be seen as a goodness-of-fit/model validation approach for the in-experienced user that otherwise might have run the risk of applying a too simple model from a too simplistic view of the data structure. We believe that such data analysis errors, where hierarchies, clusters, dependencies are not fully accounted for, is one of the more commonly occurring ones. If the user friendliness of the `step` function will make some of these more naive users apply and investigate more complex error structures, and include them for their fixed effects conclusions, it will be a step in the right direction, producing less amounts of (artificially) small p values obtained if the too simple error models were used.

Finally, we have implemented the generation of the of least square means and differences of least square means tables which use the Satterthwaite's approximation to degrees of freedom. The R package **lsmeans** package Lenth (2016a) provides a more general approach for analyzing least squares means and also supports not only linear mixed models but a broad range of different types of models.

References

Bates DM, Mäechler 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.

Dahl DB (2016). ***xtable**: Export Tables to LaTeX or HTML*. R package version 1.8-2, URL https://CRAN.R-project.org/package=xtable.

Diggle PJ (2002). *Analysis of Longitudinal Data*. Oxford University Press.

Fai AH, Cornelius PL (1996). "Approximate F-Tests of Multiple Degree of Freedom Hypotheses in Generalised Least Squares Analyses of Unbalanced Split-Plot Experiments." *Journal of Statistical Computation and Simulation*, **54**(4), 363–378. doi:10.1080/00949659608811740.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks. URL http://socserv.socsci.mcmaster.ca/jfox/Books/Companion.

Giesbrecht FG, Burns JC (1985). "Two-Stage Analysis Based on a Mixed Model: Large-Sample Asymptotic Theory and Small-Sample Simulation Results." *Biometrics*, **41**(2), 477–486. doi:10.2307/2530872.

Goodnight JH (1978). "General Linear Model Procedure." *Technical report*, SAS Institute Inc.

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.

Halekoh U, Højsgaard S (2017). ***pbkrtest**: Parametric Bootstrap and Kenward Roger Based Methods for Mixed Model Comparison*. R package version 0.4-7, URL https://CRAN.R-project.org/package=pbkrtest.

Harrell Jr FE (2017). ***Hmisc**: Harrell Miscellaneous*. R package version 4.0-3, URL https://CRAN.R-project.org/package=Hmisc.

Harvey W (1960). "Least-Squares Analysis of Data With Unequal Subclass Numbers." *Technical report*, Agricultural Research Service, US Department of Agriculture.

Højsgaard S, Halekoh U (2016). ***doBy**: Groupwise Statistics, LSmeans, Linear Contrasts, Utilities*. R package version 4.5-15, URL https://CRAN.R-project.org/package=doBy.

Kuznetsova A, Christensen RHB, Bavay C, Brockhoff PB (2015). "Automated Mixed ANOVA Modeling of Sensory and Consumer Data." *Food Quality and Preference*, **40**(A), 31–38. doi:10.1016/j.foodqual.2014.08.004.

Langsrud Ø (2003). "ANOVA for Unbalanced Data: Use Type II Instead of Type III Sums of Squares." *Statistics and Computing*, **13**(2), 163–167. doi:10.1023/a:1023260610025.

Lawless HT, Heymann H (2010). *Sensory Evaluation of Food*. Springer-Verlag. doi:10.1007/978-1-4419-6488-5.

Lenth RV (2016a). "Least-Squares Means: The R Package **lsmeans**." *Journal of Statistical Software*, **69**(1), 1–33. doi:10.18637/jss.v069.i01.

Lenth RV (2016b). ***estimability**: Tools for Assessing Estimability of Linear Predictions*. R package version 1.2, URL https://CRAN.R-project.org/package=estimability.

Littell RC, Milliken GA, Stroup WW, Wolfinger RD, Bates DM, Mächler M, Bolker B, Walker S (2014). ***SASmixed**: Data Sets from "SAS System for Mixed Models"*. R package version 1.0-4, URL https://CRAN.R-project.org/package=SASmixed.

Macnaughton DB (2009). "Which Sums of Squares are Best in Unbalanced Analysis of Variance?" Unpublished manuscript, URL http://www.matstat.com/ss/easleaao.pdf.

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

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

SAS Institute Inc (1978). "Tests of Hypotheses in Fixed-Effects Linear Models." *Technical report*, SAS Institute Inc. SAS Technical Report R-101.

SAS Institute Inc (2013). *The SAS System, Version 9.4*. SAS Institute Inc., Cary. URL http://www.sas.com/.

Satterthwaite FE (1946). "An Approximate Distribution of Estimates of Variance Components." *Biometrics Bulletin*, **2**(6), 110–114. doi:10.2307/3002019.

Schaalje GB, McBride JB, Fellingham GW (2002). "Adequacy of Approximations to Distributions of Test Statistics in Complex Mixed Linear Models." *Journal of Agricultural, Biological, and Environmental Statistics*, **7**(4), 512–524. doi:10.1198/108571102726.

Searle SR (1987). *Linear Models for Unbalanced Data*. John Wiley & Sons.

Searle SR (1997). *Linear Models*. John Wiley & Sons, New York. doi:10.1002/9781118491782.

Senn S (2007). *Statistical Issues in Drug Development Electronic Resource*. John Wiley & Sons.

Speed FM, Hocking RR, Hackney OP (1978). "Methods of Analysis of Linear Models with Unbalanced Data." *Journal of the American Statistical Association*, **73**(361), 105–112. doi:10.2307/2286530.

Venables WN (2000). "Exegeses on Linear Models." Paper presented to the S-PLUS User's Conference. Washington, DC, 1998.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Wickham H (2011). "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software*, **40**(1), 1–29. doi:10.18637/jss.v040.i01.

Zuur AF, Ieno EN, Walker N, Saveliev AA, Smith GM (2009). *Mixed Effects Models and Extensions in Ecology with R*. Springer-Verlag.

A. F- and t-statistics and the Satterthwaite's approximation

Assume we have the mixed model in Equation 1 with X the $n \times p$ design matrix for the fixed-effects and Z the $n \times k$ design matrix for the random effects.

The variance of y is therefore

$$V(\theta) = ZG(\theta)Z^\top + R(\theta),$$

where parameter θ consists of the residual error variance and the variance of random effects.

The variance-covariance matrix of β is

$$C(\theta) = (X^\top V(\theta)^{-1} X)^{-1} = (X^\top (ZG(\theta)Z^\top + R(\theta))^{-1} X)^{-1}.$$

For simplicity we will subsequently suppress θ in the notation.

Giesbrecht and Burns (1985) investigated a one-degree test of the hypothesis $H_0 = l^\top \beta$ where l is a vector. A corresponding t-statistic is then:

$$t = \frac{l^\top \hat{\beta}}{\sqrt{l\hat{C}l^\top}}, \tag{12}$$

where $\hat{C} = C(\hat{\theta})$. They followed Satterthwaite (1946) and assumed that the quantity

$$\frac{df(l^\top \hat{C} l)}{(l^\top C(\theta) l)}$$

approximately follows a χ^2 distribution. Then they used Satterthwaite's method-of moments approximation to the degrees of freedom:

$$df = \frac{2(l^\top \hat{C} l)^2}{[\mathsf{VAR}(l^\top \hat{C} l)]}.$$

Taking $f(\theta) = l^\top C(\theta) l$, $\mathsf{VAR}(f(\theta))$ can be approximated by the applying univariate delta method as:

$$\mathsf{VAR}(f(\theta)) \approx [\nabla_{f(\theta)} \hat{\theta}]^\top A [\nabla_{f(\theta)} \hat{\theta}],$$

where $\nabla_{f(\theta)} \hat{\theta}$ is a vector of partial derivatives of $f(\theta)$ with respect to θ evaluated at $\hat{\theta}$. A is the variance-covariance matrix of the $\hat{\theta}$ vector, which can be determined using the second derivatives of the log-likelihood function. Matrix A is not directly extractable from the **lme4** package. In the **lmerTest** package we specify the deviance function with respect to the θ parameters and determine the second derivatives at the optimum $\hat{\theta}$. Similarly we specify a function that calculates the variance-covariance matrix with respect to the θ parameters. Then we calculate partial derivatives evaluated at the optimum.

In a multi-degree of freedom test a hypothesis of interest is $H_0 : L\beta = 0$, where L is an estimable contrast matrix of $q = \text{rank}(L) > 1$. A commonly used test statistic for this hypothesis is:

$$F = \frac{(L\hat{\beta})^\top (L\hat{C}L^\top)^{-1} (L\hat{\beta})}{q}. \tag{13}$$

Even though the statistic is called F, it usually does not follow an F distribution. Fai and Cornelius (1996) proposed a method for approximating the distributions of F. There they also

used the Satterthwaite's method-of-moment approximation to the degrees of freedom. First they decomposed $(L\hat{C}L^\top)^{-1}$ in order to yield $P^\top(L\hat{C}L^\top)^{-1}P = D$ where P is an orthogonal matrix of eigenvectors and D is a diagonal matrix of eigenvalues. Using this decomposition, $Q = qF$ can be written as a sum of q independent variables with t distributions,

$$Q = \sum_{m=1}^{q} \frac{(PL\hat{\beta})_m^2}{D_m} = \sum_{m=1}^{q} t_{\nu_m}^2,$$

where $(PL\hat{\beta})_m$ denotes the mth element of $PL\hat{\beta}$ and D_m is the mth diagonal element of D. Then Fai and Cornelius (1996) noted that each ν_m can be approximated by the Giesbrecht-Burns single degree-of-freedom method:

$$\nu_m = \frac{2D_m}{g_m^\top A g_m},$$

where g_m is the gradient of $l_m C l_m^\top$ with respect to θ with l_m being the mth row of PL.

Using the relationship $\mathsf{E}(F_{q,\nu}) = \frac{\nu}{\nu-2}$ for $\nu > 2$, they try to find ν such that $q^{-1}Q \sim F_{q,\nu}$ approximately.

Since the t_{ν_m} can be regarded as having independent Student's t-distributions with ν_m degrees of freedom, then $\mathsf{E}(Q)$ can be calculated as:

$$\mathsf{E}(Q) = \sum_{m=1}^{q} \mathsf{E}(t_{\nu_m}^2) = \sum_{m=1}^{q} \mathsf{E}(F_{1,\nu_m}) = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2}. \tag{14}$$

Now from

$$\frac{1}{q}\mathsf{E}(Q) = \frac{\nu}{\nu - 2}$$

ν is found:

$$\nu = \frac{2\mathsf{E}(Q)}{\mathsf{E}(Q) - q}$$

with $\mathsf{E}(Q) = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2}$ (from Equation 14).

B. Hypothesis contrast matrices

The key step in constructing the F test for an effect is in constructing the contrast matrix defining the hypothesis appropriately. Package **lmerTest** implements three types of hypothesis tests introduced in SAS Institute Inc. (1978). This section describes the algorithms.

B.1. Notations and definitions

Complete rank deficient matrix

Let X be a design matrix. It can be partitioned according to the terms in the model:

$$X = [1|X_2|\dots|X_p]. \tag{15}$$

The design matrix is usually assumed to have full (column) rank. If (some of) the model effects are factors, then the matrix will not be of full rank, but it will be reduced to full rank by deletion of a selected columns.

We denote by X (cf. Equation 15) the design matrix before reduction to full column rank. This matrix is generated in **lmerTest** by creating a rank deficient design matrix for each model term separately and then concatenating them column-wise as illustrated in Equation 15.

Estimable functions

A linear function of the parameters $L\beta$ is estimable if and only if L is in the row-space of X (Searle 1997). Therefore rows of X form a *generating set* from which any estimable L can be constructed. Since the row spaces of X, $X^\top X$, $(X^\top X)^-(X^\top X)$ are identical, they all form generating sets for any estimable L. $(X^\top X)^-(X^\top X)$ has the property of containing lots of zeros, so it is used as a *generating set of estimable functions*:

$$L = (X^\top X)^-(X^\top X). \tag{16}$$

Here $^-$ is understood as a generalized inverse, X is the complete (rank-deficient) design matrix from Equation 15.

Contained effects

Consider two effects: e_1 and e_2. Then e_1 is said to be contained in e_2 if

1. all factors that appear in e_1 (if any) also appear in e_2;
2. there are more factors with e_2 than with e_1;
3. both effects involve the same continuous variables (if any).

Note: Consider the intercept (μ) as contained in all factor effects and not contained in any effect involving a continuous variable.

For instance, in the `TVbo` data the effect `TVset` is contained in the effect `TVset:Picture`, intercept μ is contained in `TVset`, `Picture` and `TVset:Picture`. More generally, e_1 is contained in e_2 if columns of the design matrix X associated with e_1 can be represented as linear combinations of the columns associated with e_2.

B.2. Type III hypothesis contrast matrices

Here we refer to the rules of generating Type III hypothesis matrices, as proposed by Goodnight (1978) for the `PROC GLM` procedure in the SAS software. Let L be the *generating set of estimable functions* (16), e be an effect, for which we want to construct hypothesis matrix, say L^e. Then the following rules create hypothesis matrix L^e:

Rule 1 Using row operations, zero out the columns in L associated with the effects that do *not* contain effect e.

1. Find columns in L associated with the effects that do *not* contain e: $j = 1, \ldots, J$.
2. For each j, find indices I of all non-zero elements in $L[, j]$.

- $L[i(1),] \longleftarrow L[i(1),]/L[i(1),j]$.
- For the rest of $i \in I$ set $L[i,] \longleftarrow L[i,] - L[i,j] \cdot L[i(1),]$.
- Set the $i(1)$th row to zero: $L[i(1),] \longleftarrow 0$.

Rule 1 assumes that e are in a so called standard order (all lower order interactions are entered in the model before higher order interactions). This is provided in package **lmerTest**.

Rule 2 Rows associated with the effects that *contain* e are orthogonalized to the rows associated with e. Starting with the first row in L having all zeros associated with e all other rows are made orthogonal to it using row operations, the row is then set to zero. This is done for all other rows having all zeros associated with e.

Example: Calculation of Type III hypothesis contrast matrix

For illustration purposes let us consider a subset of the `TVbo` data with levels `"TVset1"` and `"TVset2"` for `TVset` and levels `"Pic1"` and `"Pic2"` for the `Picture` effect. First we calculate the *generating set of estimable functions* L given in Table 4. We calculate the hypothesis matrix for `TVset` (columns 2 and 3 in L) by applying the two rules.

Apply **Rule 1**: Using row operations, zero out the columns in L associated with the effects that do *not* contain effect e.

1. Find columns in L associated with the effects that do *not* contain `TVset`: $j = 1, 4, 5$ (`(Intercept)`, `Pic1`, `Pic2`).

2. For $j = 1$, find indices I of all non-zero elements in $L[, j]$. Here only L[1,1] is non-zero in column $L[, 1]$, so $I = 1$.

   - $L[1,] \longleftarrow L[1,]/L[1,1]$.
   - Set the 1st row to zero: $L[1,] \longleftarrow 0$.

3. For $j = 4$, find indices I of all non-zero elements in $L[, j]$. Here $L[4, 4]$ is non-zero in column $L[, 4]$, so $I = 4$.

   - $L[4,] \longleftarrow L[4,]/L[4,4]$.
   - Set the 4th row to zero: $L[4,] \longleftarrow 0$.

4. For $j = 5$, find indices I of all non-zero elements in $L[, j]$. There are no non-zero elements in column $L[, 5]$, so $I = \{\}$.

The L matrix with deleted zero rows after applying Rule 1 is given in Table 5.

Rule 2 Rows associated with the effects that contain `TVset` are orthogonalized to the rows associated with `TVset`.

As can be seen the second row has zeros in columns associated with `TVset`, so the first row is orthogonalized to the second one, and then the second row is set to 0. We get the following L contrast vector for `TVset` in Table 6.

	(Intercept)	TVset1	TVset2	Pic1	Pic2	TVset1:Pic1	TVset2:Pic1	TVset1Pic2	TVset2Pic2
1	1.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	−1.00	0.00	0.00	0.00	0.00	1.00	−1.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	1.00	−1.00	0.00	1.00	0.00	−1.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	1.00	−1.00	−1.00	1.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4: L matrix for the `TVbo` example.

	(Intercept)	TVset1	TVset2	Pic1	Pic2	TVset1:Pic1	TVset2:Pic1	TVset1Pic2	TVset2Pic2
1	0.00	1.00	−1.00	0.00	0.00	0.00	0.00	1.00	−1.00
2	0.00	0.00	0.00	0.00	0.00	1.00	−1.00	−1.00	1.00

Table 5: L matrix obtained after applying Rule 1 for the `TVbo` example.

	(Intercept)	TVset1	TVset2	Pic1	Pic2	TVset1:Pic1	TVset2:Pic1	TVset1Pic2	TVset2Pic2
1	0.00	1.00	−1.00	0.00	0.00	0.50	−0.50	0.50	−0.50

Table 6: L matrix obtained after applying Rule 2 for the `TVbo` example.

B.3. Type I hypothesis contrast matrices

The Type I hypothesis contrast matrix L is the Forward-Dolittle transformation of $X^\top X$ with each non-zero row divided by its diagonal. Then the contrast matrix L^e for an effect in question e is corresponding to the effect e rows of the L matrix.

B.4. Type II hypothesis contrast matrices

The Type II hypothesis contrast matrix L^e for an effect in question e is calculated in the following way:

1. The columns of the design matrix X in Equation 15 are rearranged in a way that columns corresponding to effects that do not contain the effect e are put before the columns corresponding to the effect e. Let us denote this rearranged design matrix by X'.

2. The L matrix is calculated as the Forward-Dolittle transformation of ${X'}^\top X'$ with each non-zero row divided by its diagonal.

3. The columns of L are rearranged to reflect the original order of the model.

4. The contrast matrix L^e is corresponding to the effect e rows of the L matrix.

C. Error structure analysis in a random coefficient model

Let us consider the model in Equation 6. The error structure of this model is:

$$(b_0, b_1, b_2) \sim N(0, \begin{pmatrix} \sigma_0^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{01} & \sigma_1^2 & \sigma_{12} \\ \sigma_{02} & \sigma_{12} & \sigma_2^2 \end{pmatrix}), \quad c \sim N(0, \sigma_c^2), \quad \epsilon_{ijk} \sim N(0, \sigma^2) \tag{17}$$

Let us specify it via the `lmer` function:

```
R> m.carrots <- lmer(Preference ~ sens1 + sens2 +
+   (1 + sens1 + sens2 | Consumer) + (1 | product), data = carrots)
```

Then we apply the `step` function from the **lmerTest** package, requiring not to perform tests on the fixed effects since we are not interested in them in this example:

```
R> step(m.carrots, fixed.calc = FALSE)
```

Table 7 represents the output of the `step` function wrapped into an '`xtable`' object of the **xtable** package in order to represent the results in a compact way in the paper. The first row in the random effects table means that the LRT was applied to the model `m.carrots` and the reduced one, which does not contain the random slope `sens1`. We can see that in the following code:

```
R> m.carrots.red.sens1 <- lmer(Preference ~ sens1 + sens2 +
+    (1 + sens2 | Consumer) + (1 | product), data = carrots)
R> anova(m.carrots, m.carrots.red.sens1, refit = FALSE)


Data: carrots
Models:
..1: Preference ~ sens1 + sens2 + (1 + sens2 | Consumer) + (1 | product)
object: Preference ~ sens1 + sens2 + (1 + sens1 + sens2 | Consumer) +
object:     (1 | product)
       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
..1     8 3757.3 3798.2 -1870.7   3741.3
object 11 3761.5 3817.8 -1869.7   3739.5 1.8274      3      0.609
```

The degrees of freedom in this test are equal to 3. The tests were made for three parameters: random slope for `sens1` (σ_1^2) and correlations between the random slope `sens1` and the random slope `sens2` (σ_{12}) and the intercept (σ_{01}). Model `m.carrots.red.sens1` is the final reduced model (the "elim. num." column is equal to 0 for the rest of the rows in the random effects table meaning that the random slope `sens2` and the intercept are kept in the model according to the default Type 1 error equal to 0.1). The error structure of the final reduced model is then:

$$(b_0, b_2) \sim N(0, \begin{pmatrix} \sigma_0^2 & \sigma_{02} \\ \sigma_{02} & \sigma_2^2 \end{pmatrix}), \quad c \sim N(0, \sigma_c^2), \quad \epsilon_{ijk} \sim N(0, \sigma^2).$$

	χ^2	χ^2 df	elim. num.	p value
`sens1:Consumer`	1.83	3	1	0.6090
`sens2:Consumer`	7.81	2	0	0.0202
`product`	16.16	1	0	< 0.001

Table 7: Likelihood ratio tests for the random effects and their order of elimination representing Step 1 of the automated analysis for the `carrots` data.

Affiliation:

Alexandra Kuznetsova
Department of Applied Mathematics and Computer Science
Statistics and Data Analysis Section
DTU Compute
Richard Petersens Plads
Building 324
DK-2800 Kgs. Lyngby, Denmark
E-mail: alku@dtu.dk

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
December 2017, Volume 82, Issue 13 *Submitted:* 2016-06-30
doi:10.18637/jss.v082.i13 *Accepted:* 2016-12-19