

CLME: An R Package for Linear Mixed Effects Models under Inequality Constraints

Casey M. Jelsema

National Institute of Environmental
Health Sciences
(NIEHS)

Shyamal D. Peddada

National Institute of Environmental
Health Sciences
(NIEHS)

Abstract

In many applications researchers are typically interested in testing for inequality constraints in the context of linear fixed effects and mixed effects models. Although there exists a large body of literature for performing statistical inference under inequality constraints, user friendly statistical software for implementing such methods is lacking, especially in the context of linear fixed and mixed effects models. In this article we introduce **CLME**, a package in the R language that can be used for testing a broad collection of inequality constraints. It uses residual bootstrap based methodology which is reasonably robust to non-normality as well as heteroscedasticity. The package is illustrated using two data sets. The package also contains a graphical interface built using the **shiny** package.

Keywords: distribution free, linear inequality constraints, linear fixed effects models, linear mixed effects models, order restricted inference, residual bootstrap, R.

1. Introduction

Inequality constraints arise naturally in many applications. For example, to evaluate if a chemical is a toxin, a toxicologist may conduct a dose-response study to determine if the mean response is monotonic in dose. More precisely, suppose θ_i , $i \geq 2$, are the mean responses of a chemical corresponding to p dose groups. In this case the null and alternative hypotheses of interest are $H_0 : \theta_1 = \theta_2 = \dots = \theta_p$, and $H_a : \theta_1 \leq \theta_2 \leq \dots \leq \theta_p$, with at least one strict inequality (known as the *simple order* constraint), respectively. Sometimes, when the doses exceed the maximum tolerated dose (MTD), it may result in a dose-related toxicity and the monotonicity is violated causing down-turn at some (unknown) dose i (?). In such cases, researchers are interested in testing for an umbrella alternative $H_{ai} : \theta_1 \leq \theta_2 \leq \dots \leq \theta_{i-1} \leq \theta_i \leq \theta_{i+1} \leq \dots \leq \theta_p$, with at least one strict inequality.

In a multi-center rat uterotrophic assay conducted by the OECD (Organization for Economic Cooperation and Development), researchers were interested in studying the effect of exposure to estrogen like compounds in the uterine weights of pre-pubertal rats. They were interested in testing if the mean uterine weights of animals exposed to estrogen like compounds increased in comparison to the uterine weights of control animals (?). Thus the alternative hypothesis of interest is $H_a : \theta_1 \leq \theta_i$, $i \geq 2$, with at least one strict inequality, known as the *simple tree order*. Here θ_1 is the mean of the control group and θ_i , $i \geq 2$, are the means of the treatment

groups.

In cancer trials, it is common for researchers to be interested in evaluating a cocktail of two or more experimental drugs in combination, each tried at low, medium and high doses. In such cases, the typical order restriction of interest is the *loop order* denoted by $\{\theta_{control,control} \leq \theta_{control,low} \leq \theta_{control,medium} \leq \theta_{high,high}\} \cup \{\theta_{control,control} \leq \theta_{low,control} \leq \theta_{medium,control} \leq \theta_{high,high}\}$, where $\theta_{a,b}$ denotes the mean response corresponding to a^{th} dose of the first treatment and b^{th} dose of the second treatment. The above null and alternative hypotheses can in general be expressed as $H_0 : \mathcal{C}\theta = \mathbf{c}$ and $H_a : \mathcal{C}\theta \geq \mathbf{c}$, respectively, where \mathbf{A} is a suitable matrix of zeros, ones and negative ones of appropriate order, $\theta = (\theta_1, \theta_2, \dots, \theta_p)^T$ and \mathbf{c} is a suitable vector of known scalars, for example a vector of zero's. Some examples of \mathcal{C} and \mathbf{c} are provided later, and an illustration of some common orders is given in Figure 1.

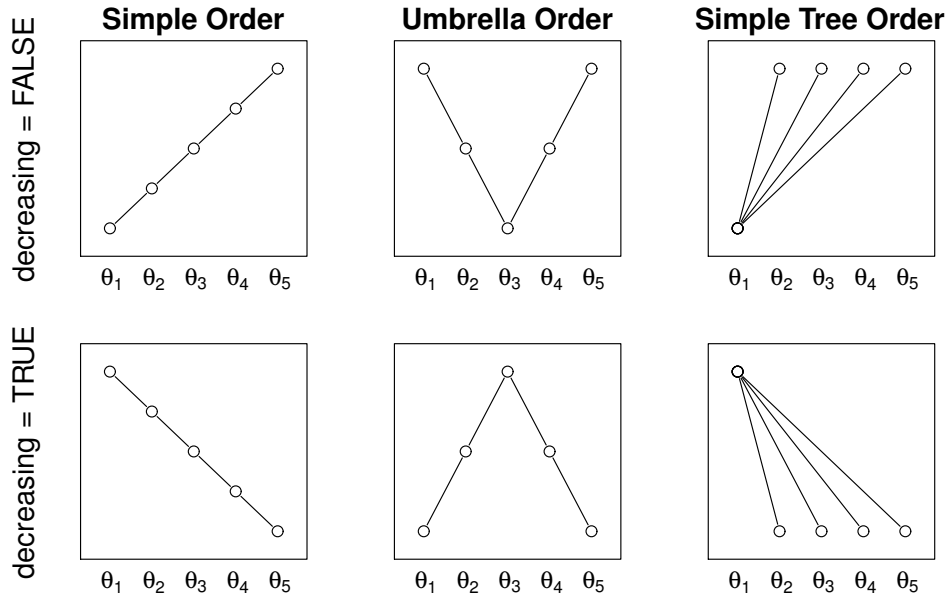


Figure 1: Illustration of order restrictions. Each circle represents a parameter of interest. Inequalities between two parameters (i.e., circles) are provided by the lines. The vertical axis denotes relative magnitude of connected parameters. No relationship (either $<$, $=$, or $>$) is known among parameters that are not connected. A nodal parameter is a parameter whose order relationship with every other parameter is known a priori or given by the hypothesis that is being tested. For example, θ_3 is the nodal parameter in the umbrella orders.

It is of common interest to perform statistical inference under inequality constraints, such as those described above, in a linear mixed effects model setting, especially in the context of repeated measures design where a researcher may be interested in detecting trends. However, despite the existence of a large body of literature on constrained inference spanning over five decades and three books on testing for order restrictions (???), it was only recently that researchers developed methods for performing constrained inference in linear mixed effects models (???). While ? and ? developed likelihood ratio based methods, ? developed a residual bootstrap based method that is designed to be robust to non-normality as well as to heteroscedasticity. Furthermore, Farnan's methodology allows for modeling categorical as well as continuous covariates.

Surprisingly, not even the popular statistical analysis program SAS (?) has the capability to perform tests under general inequality constraints in a linear fixed effects model, let alone in the context of mixed effects models. As demonstrated in ?, statistical methods that are specifically designed for testing inequality constraints are expected to enjoy substantially higher power than the usual omnibus procedures (e.g., ANOVA) which are designed for two-sided alternatives. This observation, together with the fact that there does not exist a general software for performing statistical tests under linear inequality constraints in linear mixed effects models, motivates the current work. In this paper we introduce an R package, called **CLME** ('Constrained Linear Mixed Effects') based on the distribution-free residual bootstrap methodology developed in ?. There are several packages in R which offer constrained fixed effects models, including **glmc** (?) and **ic.infer** (?), but neither of these appear to offer support for mixed models; the present work fills this void. Furthermore, since the methodology is based on residual bootstrap, **CLME** does not depend on normality or homogeneity of variances for the residuals or random effects.

The rest of the paper is organized as follows: Section 2 provides a brief description of the constrained inference for linear mixed effects (LME) models presented by ?. Section 3 describes the contents of the package CLME along with implementation details. Section 4 provides some illustrative examples using the package, and Section 5 concludes the paper with a summary and some comments on planned developments of **CLME**.

2. Linear mixed effect models under inequality constraints

Let

$$Y = \mathbf{X}_1\theta_1 + \mathbf{X}_2\theta_2 + \mathbf{U}\xi + \epsilon \quad (1)$$

denote a linear mixed effects (LME) model where Y is the $N \times 1$ response vector, \mathbf{X}_1 is a design matrix of order $N \times p_1$ and θ_1 is the corresponding $p_1 \times 1$ vector of coefficients (often treatment effects). \mathbf{X}_2 is an $N \times p_2$ a known matrix of covariates, θ_2 is the $p_2 \times 1$ vector of regression coefficients, and \mathbf{U} is a $N \times c$ matrix of known constants (random effects). For simplicity we write $\mathbf{X} = (\mathbf{X}_1 : \mathbf{X}_2)$ and $\mathbf{U} = (\mathbf{U}_1 : \mathbf{U}_2 : \dots : \mathbf{U}_{c_q})$, where $:$ denotes column-binding and \mathbf{U}_i is an $N \times c_i$ matrix, with $\sum_{i=1}^q c_i = c$. We also denote $\theta = (\theta_1^\top, \theta_2^\top)^\top$ and $p = p_1 + p_2$.

The random vector $\xi = (\xi_1^\top, \xi_2^\top, \dots, \xi_q^\top)^\top$ is $c \times 1$, where each ξ_i is a $c_i \times 1$ vector corresponding to \mathbf{U}_i , for $i = 1, \dots, q$. The elements of ξ are independently distributed with mean $\mathbf{0}$ and covariance matrix $\mathbf{T} = \text{diag}(\tau_1^2 \mathbf{I}_{c_1}, \tau_2^2 \mathbf{I}_{c_2}, \dots, \tau_q^2 \mathbf{I}_{c_q})$. The residual term ϵ is similarly defined with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1^2 \mathbf{I}_{n_1}, \sigma_2^2 \mathbf{I}_{n_2}, \dots, \sigma_k^2 \mathbf{I}_{n_k})$, where $i = 1, \dots, k$ and $\sum_{i=1}^k n_i = N$.

Although the above model description and the methodology implemented in **CLME** allows for fairly general settings, in many applications one may not require the full available flexibility. For example, in most applications it may be sufficient to assume that $T = \tau^2 \mathbf{I}$, instead of the general heteroscedastic structure for T described above.

Let \mathcal{C} be an $r \times p$ matrix so that $\mathcal{C}\theta$ represents the linear combinations which are subject to inequality constraints specified by the alternative hypothesis. Thus the hypotheses of interest are given by:

$$H_o : \mathcal{C}\theta = \mathbf{0} \text{ versus } H_a : \mathcal{C}\theta \geq \mathbf{0}, \quad (2)$$

such that at least one of the r inequalities is strict. ? suggested the pool adjacent violators algorithm (‘PAVA’) to implement the order constraints (isotonization). We depart from their methodology in that we use the package **isotone** (?) for isotonization, and in particular the function **activeSet**. In some cases using active set methodology leads to the same results as using PAVA; though using active sets is a more general approach and enables easy specification of complex order restrictions. This also enables access to a number of *solvers* for **activeSet**, including least squares, least absolute deviation, and others.

CLME is designed to implement two general classes of statistical tests. The likelihood ratio type (LRT) statistic (?) is the default setting, but the user may instead choose the Williams’ type test statistic (???). In both cases, to keep the methodology robust to non-normality and potential heteroscedasticity, the p values are evaluated using the residual bootstrap methodology developed in ?. Thus, although the likelihood ratio type statistic is motivated by the likelihood ratio principle under the normality assumption, it does not use the normal theory based asymptotic distribution for the test statistic. Hence we use the phrase ‘likelihood ratio *type* test’ rather than ‘likelihood ratio test’.

Using simulations, ? demonstrated that the Williams’ type test enjoys higher power than the likelihood ratio type statistic for simple alternative hypothesis; hence it may be preferred over the likelihood ratio type statistic in such cases. In general the Williams’ type test statistic is of the form:

$$W = \max \left\{ \left[\mathcal{B}\tilde{\theta}_1 \right] \odot \left[\sqrt{\text{diag} \left\{ \mathcal{B}\text{Var}(\hat{\theta})\mathcal{B}^\top \right\}} \right]^{-1} \right\}, \quad (3)$$

where \odot denote the Schur-product of vectors, i.e., $\mathbf{a} \odot \mathbf{b} = (a_1b_1, a_2b_2, \dots, a_rb_r)^\top$, $\tilde{\theta}_1$ denotes the estimator of θ_1 under the inequality constraint of interest, and $\hat{\theta}_1$ denotes the unconstrained estimator of θ_1 (e.g., the MLE). For a given order restriction specified by \mathcal{C} , the contrast matrix \mathcal{B} is derived from the largest hypothesized difference(s); in the simple order this is the difference between θ_1 and θ_{p_1} .

3. Contents of CLME

In this section we describe the functions included in **CLME** and some notes on their implementation. We start by describing the main function of the package, **clme**. Afterwards, we detail some of the secondary functions which users may find useful.

3.1. Main function

The main function of **CLME** is **clme**. This function implements the order restricted residual bootstrap test described in ?. Among the arguments listed below, only the formula and the dataset are required. A series of flowcharts are provided in the appendix (Figures 5-7) to guide a user through specification of the arguments for **clme**.

formula a formula expression; the constrained effect(s) must come before any unconstrained effects.

data data frame containing the variables in the model.

gfix optional vector of group levels for residual variances. Data should be sorted by this value.

constraints list containing the constraints.

nsim optional number of bootstrap samples to use for significance testing.

tsf function to calculate the test statistic.

tsf.ind function to calculate the test statistic for individual contrasts.

mySolver solver to use in isotonization (passed to **activeSet**).

verbose logical, prints iteration step. Argument can be vector of multiple logicals; successive elements are passed to further functions.

seed set the seed for the random number generator (RNG).

levels list to manually specify labels for constrained coefficients.

ncon the number of constrained terms in the **formula**; the first **ncon** terms are constrained.

... space for additional arguments.

Several of the arguments to **clme** require further explanation.

Constraints The argument **constraints** is a list describing the order restrictions using the following elements:

order text string specifying the type of order. Allowed values are ‘simple’, ‘umbrella’, and ‘simple.tree’.

node numeric indicating which element of θ_1 is the node.

decreasing logical indicating decreasing order. For simple orders, a decreasing order implies a downward trend. For umbrella or simple tree orders, a decreasing order implies a decrease from the node. See Figure 1 for an illustration.

A matrix describing the order restrictions in \mathcal{C} .

B matrix of coefficients defining the Williams type statistic (only necessary if Williams’ type test is desired).

The values of **A** and **B** use the same format as the argument **isomat** from the function **activeSet** in package **isotone**: each is a matrix with two columns where the rows define a specific constraints. For example, **A** and **B** are shown below for the decreasing umbrella order with $p_1 = 5$ and a node at $\theta_{1,3}$ (the third element of θ_1).

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 4 & 3 \\ 5 & 4 \end{bmatrix}$$

and,

$$\mathbf{B} = \begin{bmatrix} 1 & 3 \\ 5 & 3 \end{bmatrix}$$

The alternative hypothesis is $H_a : \theta_1 \leq \theta_2 \leq \theta_3 \geq \theta_4 \geq \theta_5$. The first row of \mathbf{A} defines the constraint $\theta_1 \leq \theta_2$, the second row defines the constraint $\theta_2 \leq \theta_3$, and so on. The entries of the rows are the coefficient indices, and the parameter indexed in the left column is hypothesized to be less than or equal to that indexed by the right column. The \mathbf{B} matrix is similarly structured, but defines only the contrasts for the largest hypothesized difference(s). Under the umbrella order, this will be the node compared to the first and last values; the specific form of the Williams' type statistic from Equation 3 is:

$$W = \max \left\{ \frac{\tilde{\theta}_3 - \tilde{\theta}_1}{\sqrt{\text{Var}(\hat{\theta}_3 - \hat{\theta}_1)}}, \frac{\tilde{\theta}_3 - \tilde{\theta}_5}{\sqrt{\text{Var}(\hat{\theta}_3 - \hat{\theta}_5)}} \right\},$$

hence the \mathbf{B} matrix holds the contrasts $\tilde{\theta}_3 - \tilde{\theta}_1$ and $\tilde{\theta}_3 - \tilde{\theta}_5$.

Not all of the elements of `constraints` are necessary. There are three general formats by which to pass the constraints to `clme`.

Specific Defaults One may specify only the elements `order`, `node`, and `decreasing`. In this case the program will call an internal function to generate the values of \mathbf{A} and \mathbf{B} . Allowed values for `order` are 'simple', 'umbrella', and 'simple.tree'; also, the node may be omitted for simple orders. Each of the three elements may also be vector-valued (e.g., `order=c('simple', 'umbrella')`) to test multiple orders.

Custom constraints Alternatively, the list of constraints may contain \mathbf{A} directly. This is particularly useful for specifying custom order restrictions such as loop orders or block orders. When a custom \mathbf{A} is passed, the program will ignore any values of `order`, `node`, and `decreasing`. If the Williams' type test is selected, a custom \mathbf{B} is also needed.

Unspecified Finally, `constraints` may be left unspecified. In this case the program will search for both simple and umbrella orders with all possible nodes, both increasing and decreasing orders. As with the first case, the program will estimate the order using the maximum test statistic of all the tested orders.

When testing multiple orders the test statistic is taken as the maximum of all the tested orders, and the program will note the order which produced this value as the estimated order. The bootstrap null distribution of the test statistic is constructed from all the order patterns under consideration, not just the estimated order (that is, for each bootstrap sample, the test statistic is computed for all candidate orders, and the maximum is taken). For reproducibility, one may use `seed` argument to set the seed for the pseudo-random number generator.

Test statistic: `tsf` and `tsf.ind` The argument `tsf` is a function which computes the desired global test statistic. This defaults to `lrt.stat`, the LRT statistic. Alternatively one may select the Williams' type statistic from Equation 3 by setting `tsf=w.stat`. The related

argument `tsf.ind` computes the test statistic to test the individual constraints. The Williams type test, `w.stat.ind`, is default. These two arguments are analogous to the global F test and pairwise t tests in the context of analysis of variance. For other test statistics, the user may submit a custom function for `tsf` and/or `tsf.ind`. We refer to the documentation of `lrt.stat` for more details on the format of custom test statistic functions.

The output from any custom `tsf` should be numeric. Output with length greater than 1 corresponds to multiple global hypotheses being tested. This should not be used for testing each individual constraint from the \mathbf{A} matrix, as these are calculated separately using the `tsf.ind` argument. If desired, the test statistic function should also specify the names attribute of the test statistic, for example naming the contrast. An example of testing multiple global hypotheses is shown in Section 4.2, a reanalysis of data from the Fibroid Growth Study (?).

Homogeneity of variances: `gfix` The model described in Section 2 permits a large degree of flexibility. In particular, both ξ (if random effects are included) and ϵ may be modeled under the assumption of homogeneity or heterogeneity of variances. Currently, each random effect term is modeled with a separate variance component. The argument `gfix` defines groups for the residual variance(s). By default, the data are modeled with a single residual variance. If `gfix` is supplied, then each group of `gfix` is modeled with a separate residual variance. For example if the constrained effect is the variable `x1`, defined as treatment groups, then `gfix=x1` will produce a residual variance for each treatment group.

The output of `clme` is a list with elements:

`theta` vector of estimates of fixed-effects coefficients, θ .

`theta.null` vector of estimates of θ under the null hypothesis.

`ssq` estimate of the residual variance(s), σ_i^2 , $i = 1, \dots, k$.

`tsq` estimate of the random effect variance component(s), τ_i^2 , $i = 1, \dots, q$.

`cov.theta` the covariance matrix of the unconstrained estimates of θ .

`ts.glob` test statistic for the global hypothesis.

`ts.ind` vector of test statistics for each of the constraints (each row of \mathbf{A}).

`mySolver` the solver used in `activeSet`.

`p.value` p value for the global hypothesis.

`p.value.ind` Vector of p values for each of the constraints.

`constraints` List containing the constraints (\mathbf{A}) and the contrast for the global test (\mathbf{B}).

`dframe` data frame containing the variables in the model.

`residuals` matrix containing residuals. For mixed models three types of residuals are given.

`random.effects` predicted values of the random effects.

`gfix` group sample sizes for residual variances.

gran group sizes for random effect variance components.

formula the formula used in the model.

call the function call.

order list describing the specified constraints.

P1 the number of constrained parameters.

3.2. Secondary functions

The function **clme** calls several other functions in the course of its evaluation. The three primary ones are: **clme_em**, **clme_resids**, and **resid_boot**. These three functions perform an integral role for **clme** and may be of use as independent functions outside of normal use, such as bootstrapping the residuals but not necessarily running the EM algorithm and obtaining the test statistic. Only **resid_boot** is described here; all output from **clme_em** and most of the output from **clme_resids** can be obtained simply by running **clme** with **nsim=0**.

Residual bootstrap The function **resid_boot** obtains the bootstrap samples \mathbf{Y}^* of the data response vector. The arguments to this function are:

formula a formula expression, the constrained effect should be the first term on the right-hand side.

data data frame containing the variables in the model.

gfix optional vector of group levels for residual variances.

null.resids logical, whether to generate bootstrap samples under the null hypothesis. Defaults to **TRUE**.

eps estimates of residuals.

xi predicted values of the random effects.

theta vector of fixed-effects coefficients.

ssq vector of residual variance estimate(s).

tsq vector of random effect variance component(s).

cov.theta covariance matrix of the unconstrained fixed effects estimates.

seed set the seed for the RNG

nsim number of bootstrap samples to generate (M).

mySolver the solver to pass to **activeSet**.

... space for additional arguments.

The required arguments are `formula` and `data`, these are used to obtain the model matrices. If provided, the values of `theta`, `ssq`, `tsq`, `eps` and `xi` will be used for bootstrapping; any of these that are missing will be estimated. Regardless of whether `theta` is provided or estimated, if `null.resids=TRUE`, then `theta` will be projected onto the space of the null hypothesis. To generate bootstraps with a specific `theta`, set `null.resids=FALSE`. The output of `resid_boot` is a matrix of size $N \times M$, where each column is a bootstrap sample \mathbf{Y}^* of the data vector \mathbf{Y} .

3.3. Other package contents

Shiny application The `shiny` package (?) offers the ability to develop a graphical user interface (GUI) which implements **CLME**. A GUI developed in `shiny` can be run locally or deployed online. This is particularly beneficial to researchers who are not as familiar with R, or programming in general, but wish to use the methods described here. The package **CLME** includes a `shiny` application to run `clme`. After installing the package, a user may run the command `shiny_clme()` to call the GUI and begin using **CLME** without any need for further programming.

The data should be a CSV file with the first row being a header. Variables are identified using their column letter or number (e.g., 1 or A). Multiple variables may be separated by a comma (e.g., 1,2,4 or A,B,D), a range of variables may be defined with a dash (e.g., 1-4 or A-D), or a combination of the two can be used. These values should be set to 'None' to indicate no covariates or random effects. Group levels for the constrained effect may not be read into R with the correct order; an extra column may contain the ordered group levels (it may therefore have different length than the rest of the dataset).

Methods The function `clme` outputs an object of the S3 class `clme`. The methods available for this class are briefly described in Table 1.

4. Sample implementation

In this section we demonstrate the use of **CLME** by applying it to two real-world data sets. Some of the analyses mimic those performed in the original papers but in the context of order-restricted inference. Other analyses are intended to exhibit certain features of the package or compare the available options. We emphasize that these analyses are intended not as a scientific reanalysis, but as an illustration. Consequently some modeling choices, the assumption of homogeneity of variances in particular, are not thoroughly investigated. The data analyzed are included in the package as the datasets `rat.blood` and `fibroid`.

4.1. Hematologic parameters from Sprague-Dawley rats

In a recent study on the effect the amount of time a sample is stored has on various hematological parameters, ? conducted a time course study using blood samples drawn from Sprague-Dawley rats. Blood samples from 11 female and 11 male rats were kept at either room temperature 21 °C (the control group) or refrigerated at 3 °C for 6, 24, 48 or 72 hours (see ? for more details). Although the authors obtained data on a variety of hematological variables in this repeated measure time course study, we shall focus on hematocrit (HCT)

Generic name	Description
AIC	Computes Akaike information criterion.
as	Coerces an object to class <code>clme</code> .
confint	Computes individual confidence intervals for fixed effects parameters. Intervals are centered at the constrained estimates, but use standard errors of the unconstrained estimates.
fixef	Extracts estimates fixed-effects coefficients, θ .
formula	Extracts the formula for the model.
is	Tests if an object is of class <code>clme</code>
logLik	Computes the log-likelihood under the assumption of Normality.
model.frame	Data frame with the variables in the model.
model.matrix	The fixed-effects design matrix.
nobs	Number of observations.
plot	Produces a plot of the constrained coefficients and denotes statistical significance.
print	A basic printout of the model results.
ranef	Extracts predictions of the random effects.
residuals	Extract various types of residuals.
sigma	The residual variance(s).
summary	A more detailed printout of model results.
VarCorr	Estimates of variance components.
vcov	The variance-covariance matrix of the fixed-effects estimates.

Table 1: List of methods currently defined for objects of class `clme`.

and the white blood cell (WBC) count over time. In the case of HCT we shall illustrate some of the options of **CLME** while testing for simple order with an increasing trend in time. In the case of WBC we test for simple tree order the mean WBC count in the freezer group was at least as high as that of the 0 hour.

First, we load the package and the data.

```
R> library("CLME")
R> data("rat.blood")
```

Hematocrit We illustrate **CLME** using three different settings. In the first case (Case A) we test the following hypotheses:

$$H_0 : \theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5$$

Vs.

$$H_{aA} : \theta_1 \leq \theta_2 \leq \theta_3 \leq \theta_4 \leq \theta_5, \quad (A)$$

with at least one strict inequality, here θ_i is the mean corresponding to either 0, 6, 24, 48 or 72 hours. In the second case (Case B), we test for a union of umbrella alternatives. If the null hypothesis is rejected then the algorithm selects the pattern that has largest value of test statistic:

$$H_0 : \theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5$$

Vs.

$$H_{aB} : \left\{ \bigcup_{i=1}^5 \theta_1 \leq \theta_2 \leq \dots \leq \theta_i \geq \dots \geq \theta_5 \cup \bigcup_{i=1}^5 \theta_1 \geq \theta_2 \geq \dots \geq \theta_i \leq \dots \leq \theta_5. \right\} \quad (B)$$

Thus in (B) the order is unspecified but limited to either umbrella or inverted umbrella orders. Note that simple orders (increasing or decreasing) are a special case of umbrella orders, where the peak is the first or last parameter. The peak or the trough of each umbrella is specified using the specification of **node**. Case (C) is a repeat of case (A), but there we will assume heteroscedasticity of variances between the time groups.

We initially use the default arguments as far as possible. We use the gender of the rat and the storage temperature of the sample as covariates. The R code to test case (A) is provided below along with the results.

```
R> const <- list(order = "simple", node = 1, decreasing = FALSE)
R> hct1 <- clme(hct ~ time + temp + sex + (1|id), data = rat.blood, seed = 42,
+   constraints = const, levels = list(2, levels(rat.blood$time)))
R> summary(hct1)
```

```
Linear mixed model subject to order restrictions
Formula: hct ~ time + temp + sex + (1 | id) - 1
```

```
Order specified: increasing simple order
```

```
log-likelihood: -1564
AIC:           3147
BIC:           3160
(log-likelihood, AIC, BIC computed under normality)
```

Global test:

Contrast	Statistic	p-value
Bootstrap LRT	0.17	0.0020

Individual constraints:

Contrast	Estimate	Statistic	p-value
6 Hour - 0 Hour	1.342	4.862	0.0000
24 Hour - 6 Hour	0.086	0.399	0.1480
48 Hour - 24 Hour	0.180	0.829	0.0550
72 Hour - 48 Hour	0.150	0.693	0.1070

Variance components:

Source	Variance
id	2.2006
Residual	1.0314

Fixed effect coefficients (theta):

	Estimate	Std. Err	95% lower	95% upper
0 Hour	40.9121	0.5226	39.888	41.936
6 Hour	42.2542	0.5055	41.263	43.245
24 Hour	42.3405	0.5055	41.350	43.331
48 Hour	42.5201	0.5055	41.529	43.511
72 Hour	42.6701	0.5055	41.679	43.661
tempRT	0.5023	0.1531	0.202	0.802
sexMale	-1.8333	0.6804	-3.167	-0.500

Std. Errors and confidence limits based on unconstrained covariance matrix

Parameters are ordered according to the following factor levels:

0 Hour, 6 Hour, 24 Hour, 48 Hour, 72 Hour

```
R> plot( hct1, ci = TRUE, legendx = "bottomright", inset = 0.08)
```

We find strong evidence ($p = 0.002$) of an increasing pattern in mean HCT. The coefficients are plotted in Figure 2 with indications of significance for the individual contrasts.

To test case (B) we simply need to omit the constraints from the call to `clme`. The code and results are given below.

```
R> hct2 <- clme(hct ~ time + temp + sex + (1/id), data = rat.blood, seed = 42,
+   levels = list(2, levels(rat.blood$time)))
R> summary(hct2)
```

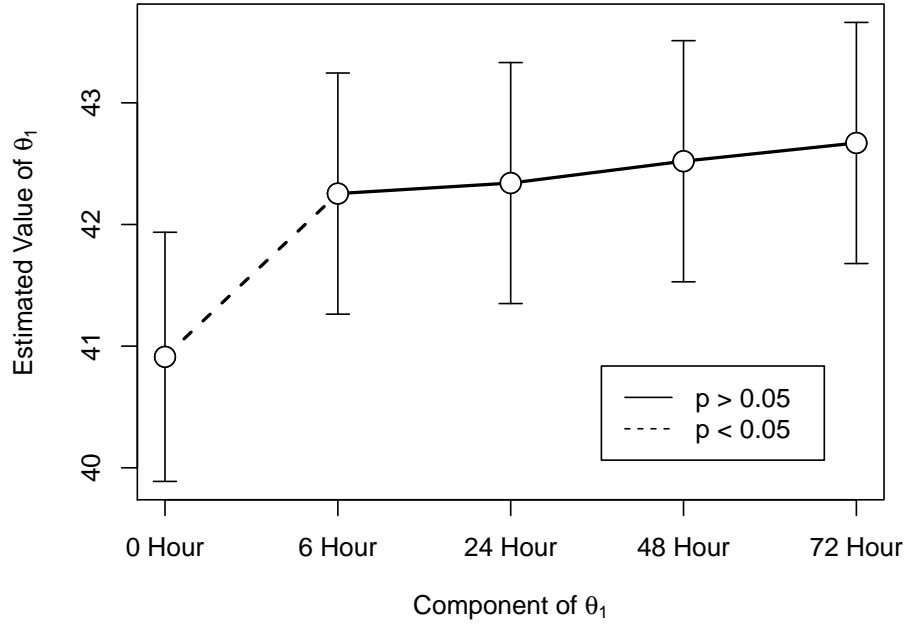


Figure 2: Plot of estimated coefficients of mean hematocrit (HCT) from Case (A). The model assumed an increasing simple order and homogeneity of variances across treatment groups. Solid lines denote no significant difference, while dashed lines denote statistical significance.

Linear mixed model subject to order restrictions

Formula: `hct ~ time + temp + sex + (1 | id) - 1`

Order estimated: increasing simple order

log-likelihood: -1564

AIC: 3147

BIC: 3160

(log-likelihood, AIC, BIC computed under normality)

Global test:

Contrast	Statistic	p-value
Bootstrap LRT	0.17	0.0070

Individual constraints:

Contrast	Estimate	Statistic	p-value
6 Hour - 0 Hour	1.342	4.862	0.0000
24 Hour - 6 Hour	0.086	0.399	0.1480
48 Hour - 24 Hour	0.180	0.829	0.0550

72 Hour - 48 Hour	0.150	0.693	0.1070
-------------------	-------	-------	--------

Variance components:

Source	Variance
id	2.2006
Residual	1.0314

Fixed effect coefficients (theta):

	Estimate	Std. Err	95% lower	95% upper
0 Hour	40.9121	0.5226	39.888	41.936
6 Hour	42.2542	0.5055	41.263	43.245
24 Hour	42.3405	0.5055	41.350	43.331
48 Hour	42.5201	0.5055	41.529	43.511
72 Hour	42.6701	0.5055	41.679	43.661
tempRT	0.5023	0.1531	0.202	0.802
sexMale	-1.8333	0.6804	-3.167	-0.500

Std. Errors and confidence limits based on unconstrained covariance matrix

Parameters are ordered according to the following factor levels:

0 Hour, 6 Hour, 24 Hour, 48 Hour, 72 Hour

Observe that the alternative hypothesis in (B) is much larger than the alternative hypothesis in (A). Thus, while the conclusions of tests for (A) and (B) are the same: that the parameters satisfy an increasing simple order, the p value associated with (B) is larger because the alternative hypothesis in (B) is larger than the alternative in (A).

Accounting for heteroscedasticity is simple in **CLME**. For example, suppose we wish to model each of the time points with a different residual variance. To do this we pass the time groups as the argument `gfix`, as shown below. We will call this case (C).

```
R> hct3 <- clme(hct ~ time + temp + sex + (1|id), data = rat.blood, seed = 42,
+   gfix = rat.blood$time, constraints = const,
+   levels = list(2, levels(rat.blood$time)))
R> summary( hct3 )
```

Linear mixed model subject to order restrictions

Formula: `hct ~ time + temp + sex + (1 | id) - 1`

Order specified: increasing simple order

log-likelihood: -1629

AIC: 3284

BIC: 3303

(log-likelihood, AIC, BIC computed under normality)

Global test:

Contrast	Statistic	p-value
----------	-----------	---------

Bootstrap LRT 0.266 0.0010

Individual constraints:

Contrast	Estimate	Statistic	p-value
6 Hour - 0 Hour	1.326	4.937	0.0000
24 Hour - 6 Hour	0.086	0.465	0.1250
48 Hour - 24 Hour	0.180	0.902	0.0430
72 Hour - 48 Hour	0.150	0.823	0.0890

Variance components:

Source	Variance
id	2.13092
0 Hour	1.22617
6 Hour	0.54469
24 Hour	0.97074
48 Hour	0.77105
72 Hour	0.69144

Fixed effect coefficients (theta):

	Estimate	Std. Err	95% lower	95% upper
0 Hour	40.9258	0.5032	39.940	41.912
6 Hour	42.2521	0.4625	41.346	43.159
24 Hour	42.3385	0.4729	41.412	43.265
48 Hour	42.5180	0.4681	41.601	43.435
72 Hour	42.6680	0.4661	41.754	43.582
tempRT	0.5341	0.1273	0.285	0.784
sexMale	-1.8511	0.6345	-3.095	-0.608

Std. Errors and confidence limits based on unconstrained covariance matrix

Parameters are ordered according to the following factor levels:

0 Hour, 6 Hour, 24 Hour, 48 Hour, 72 Hour

White blood cell count Using the white blood cell count data of ?, we now illustrate our package for testing a simple tree order. Here the nodal parameter is taken to be the population mean corresponding to the 0 hour group. Since boxplots of the residuals (not shown) suggested the variances were potentially equal across the groups, we assume homogeneity of variances. For illustration, in this example we use the Williams' type test statistic. The code and results are below, and the coefficients are plotted in Figure 3.

```
R> const <- list( order = "simple.tree" , node = 1 , decreasing = FALSE)
R> wbc <- clme(wbc ~ time + temp + sex + (1|id), data = rat.blood, seed = 42,
+   constraints = const, levels = list(2, levels(rat.blood$time)),
+   tsf = w.stat )
R> summary(wbc)
```

Linear mixed model subject to order restrictions

Formula: wbc ~ time + temp + sex + (1 | id) - 1

Order specified: increasing tree order with node at 1

log-likelihood: -1624

AIC: 3266

BIC: 3279

(log-likelihood, AIC, BIC computed under normality)

Global test:

Contrast	Statistic	p-value
72 Hour - 0 Hour	5.207	0.0000

Individual constraints:

Contrast	Estimate	Statistic	p-value
6 Hour - 0 Hour	0.000	0.000	1.0000
24 Hour - 0 Hour	0.409	2.240	0.0120
48 Hour - 0 Hour	0.574	3.150	0.0000
72 Hour - 0 Hour	0.949	5.207	0.0000

Variance components:

Source	Variance
id	1.51920
Residual	0.45017

Fixed effect coefficients (theta):

	Estimate	Std. Err	95% lower	95% upper
0 Hour	5.4262	0.4007	4.641	6.212
6 Hour	5.4262	0.3910	4.660	6.193
24 Hour	5.8348	0.3910	5.068	6.601
48 Hour	6.0007	0.3910	5.234	6.767
72 Hour	6.3757	0.3910	5.609	7.142
tempRT	-0.1947	0.1011	-0.393	0.004
sexMale	1.8229	0.5336	0.777	2.869

Std. Errors and confidence limits based on unconstrained covariance matrix

Parameters are ordered according to the following factor levels:

0 Hour, 6 Hour, 24 Hour, 48 Hour, 72 Hour

`R> plot(wbc, legend = "topleft", inset = 0.08)`

Our results are consistent with those of ?, but we have in addition detected the 0 hour - 48 hour and 0 hour - 24 hour contrasts as being significant, which were not identified by ?. There does appear to be an increasing pattern over time, but the differences from control are not statistically significant until sufficient time has passed.

As an alternative to Williams' type test, we repeated the analysis using the LRT (results not provided) and discovered that the LRT did not reject the null hypothesis at the 5% level of significance ($p = 0.252$). This discrepancy between Williams' type and LRT is not surprising

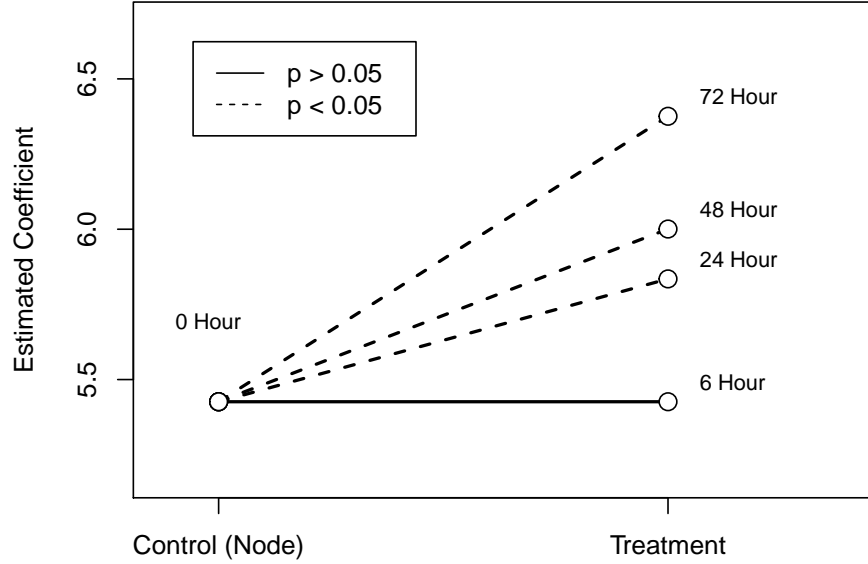


Figure 3: Plot of estimated coefficients of white blood cell (WBC) count. Solid lines denote no significant difference, while dashed lines denote statistical significance.

in view of the simulation study reported in ?, which indicated that Williams' type test can be more powerful than LRT in some cases.

4.2. Fibroid growth rates

? investigated growth rate of of uterine leiomyomata (fibroids) in black and white women. Since fibroids are hormonally mediated and there is a drop in estrogen levels as women age, it may be reasonable to hypothesize a reduction in fibroid growth rates. Interestingly, ? reported that for white women the rate of growth of fibroids decreased with age (i.e., simple order with decreasing pattern), whereas they did not find any reduction in the average growth rate of fibroids with age for black women. They defined the three age groups as follows: Young (< 35), Middle ($35 - 44$), and Old (≥ 45). We shall now re-analyze their data using the methodology available in our package **CLME** where the alternative hypothesis for women of each race group is a decreasing simple order. Note that for confidentiality, we use a subset of the data from the Fibroid Growth Study, excluding cases which may be personally identifiable, particularly those with only one fibroid analyzed in the study. This subset of the data represents 240 fibroids on 54 women. The original analysis in ? represented 262 fibroids on 72 women.

The interest in this case is to test for a simple order for *each* race using a linear mixed effects model. This analysis serves as a useful illustration of customizing the order restrictions, because it cannot be performed with the default settings of **CLME**. First we load the data

and perform some manipulations to get a factor that we can use. We define the variable `race.age` to encode the interaction of the race and age variables; with six levels ordered as: young black, middle-age black, older black, young white, middle-age white, and older white.

```
R> data("fibroid")
R> race.age <- factor(paste0( fibroid$race, ".", fibroid$age ) ,
+   levels = c("Black.Yng", "Black.Mid", "Black.Old",
+   "White.Yng", "White.Mid", "White.Old") )
R> fibroid$race.age <- race.age
```

We performed our analysis adjusting for the initial fibroid volume as a covariate, which was grouped into three categories: $< 14, 14 - 65\text{cm}^3, \geq 65\text{cm}^3$ with the < 14 category taken to be the baseline. To deal with repeated measurements, we took subject ID as the random effect.

```
R> initVol <- rep("small", nrow(fibroid))
R> idx1 <- (14000 <= fibroid$vol & fibroid$vol < 65000)
R> idx2 <- (65000 <= fibroid$vol)
R> initVol[idx1] <- "medium"
R> initVol[idx2] <- "large"
R> fibroid$initVol <- factor(initVol, levels = c("small", "medium", "large"))
```

For the interaction between the Age and Race terms, we require constraints which define a decreasing simple order over age for both blacks and whites, but do not impose any order restriction *between* blacks and whites. We do this as follows:

```
R> const <- list()
R> const$A <- cbind( 2:6 , 1:5 )[-3, ]
R> const$B <- rbind( c(3, 1), c(6, 4) )
R> const
```

```
$A
      [,1] [,2]
[1,]     2     1
[2,]     3     2
[3,]     5     4
[4,]     6     5
```

```
$B
      [,1] [,2]
[1,]     3     1
[2,]     6     4
```

To understand the construction of these matrices, recall the parameter vector θ_1 is ordered as: young black, middle-age black, older black, young white, middle-age white, and older white. The groups of the constrained effect are transformed into column indicators, meaning there will be six parameters: $(\theta_{YB}, \theta_{MB}, \theta_{OB}, \theta_{YW}, \theta_{MW}, \theta_{OW})$, where YB denotes "young black,"

MB denotes "middle-aged black," and so on. Hence, the first three elements of θ correspond to the blacks, and the last three elements correspond to the whites.

The **A** matrix must define the proper order restriction on these elements. The first row defines the constraint $\theta_{MB} \leq \theta_{YB}$, the first row defines the constraint $\theta_{OB} \leq \theta_{MB}$. The second two rows define similar constraints for the white women. None of the rows define a restriction between any of the first three elements (blacks) and any of the last three elements (whites); hence there is no order restriction imposed between the two races.

To test for a decreasing simple order for both blacks and whites, we must also define a function to compute the Williams' type test statistic of ? for both blacks and whites separately. While the matrix of contrasts is provided above, by default the Williams' type test will take the maximum and report a single test statistic. We require a test statistic for each of these contrasts. This is similar to the function `w.stat.ind` which calculates the test statistics for the individual constraints. However, submitting `tsf=w.stat.ind` will test all of the constraints in the matrix `constA` instead of the contrasts in `constB`. To correct this, we make a small modification to `w.stat.ind`.

```
R> w.blk.wht <- function (theta, cov.theta, B, A, ...) {
+   stats <- vector("numeric", length = nrow(B))
+   ctd <- diag(cov.theta)
+   stats <- apply(B, 1, FUN = function(a, theta, cov, ctd) {
+     std <- sqrt(ctd[a[1]] + ctd[a[2]] - 2 * cov.theta[a[1], a[2]])
+     (theta[a[2]] - theta[a[1]])/std
+   }, theta = theta, cov = cov.theta, ctd = ctd)
+   names(stats) <- c("Black.Yng - Black.Old", "White.Yng - White.Old" )
+   return(stats)
+ }
```

All we have done is replace calls to **A** with calls to **B**; this will accomplish our goal of producing a global test for both blacks and whites individually.

We are then ready to run the analysis. For simplicity, we assume homogeneity of variances. Results of the analysis are shown in Figure 4. The code for this figure is given below, since it cannot be produced through **CLME**.

```
R> fib <- clme(lfgr ~ race.age + initVol + (1|id), data = fibroid, seed = 42,
+   constraints = const, tsf = w.blk.wht, levels = list(10, levels(race.age)))
R> summary( fib )
```

```
Linear mixed model subject to order restrictions
Formula: lfgr ~ race.age + initVol + (1 | id) - 1
```

Custom order constraints were provided

```
log-likelihood: -2638
AIC:           5296
BIC:           5313
(log-likelihood, AIC, BIC computed under normality)
```

Global tests:

Contrast	Statistic	p-value
Black.Yng - Black.Old	1.116	0.1880
White.Yng - White.Old	2.270	0.0150

Individual constraints:

Contrast	Estimate	Statistic	p-value
Black.Yng - Black.Mid	8.752	1.454	0.0580
Black.Mid - Black.Old	0.000	0.000	1.0000
White.Yng - White.Mid	10.391	1.217	0.0640
White.Mid - White.Old	7.703	1.059	0.0830

Variance components:

Source	Variance
id	107.58
Residual	419.15

Fixed effect coefficients (theta):

	Estimate	Std. Err	95% lower	95% upper
Black.Yng	21.471	5.087	11.501	31.441
Black.Mid	12.719	3.910	5.055	20.383
Black.Old	12.719	6.277	0.416	25.023
White.Yng	21.795	6.746	8.573	35.016
White.Mid	11.403	5.751	0.132	22.674
White.Old	3.700	4.872	-5.848	13.249
initVolmedium	-4.720	3.201	-10.995	1.554
initVollarge	-3.641	3.854	-11.194	3.912

Std. Errors and confidence limits based on unconstrained covariance matrix

Parameters are ordered according to the following factor levels:

Black.Yng, Black.Mid, Black.Old, White.Yng, White.Mid, White.Old

```
R> plot(x = 1, y = 0, col = 0, ylim = c(-6, 22), xlim = c(0.9, 3.1),
+       xlab = "", ylab = "Estimated Coefficient", xaxt = "n")
R> axis(side=1, at=1:3,
+       labels = c("Young (<35)", "Middle aged (35-44)", "Older (>45)") )
R> for( y1 in seq(-15, 25, 5) ){
+   lines( x = c(0, 7), y = c(y1, y1), col = "grey", lty = 2 )
+ }
R> lty1 <- 1 + (fib$p.value.ind[1]<0.05)
R> lty2 <- 1 + (fib$p.value.ind[2]<0.05)
R> lty3 <- 1 + (fib$p.value.ind[3]<0.05)
R> lty4 <- 1 + (fib$p.value.ind[4]<0.05)
R> points(c(1, 2), fib$theta[1:2], col = 1, type = "l", lwd = 2 , lty = lty1)
R> points(c(2, 3), fib$theta[2:3], col = 1, type = "l", lwd = 2 , lty = lty2)
R> points(c(1, 2), fib$theta[4:5], col = 3, type = "l", lwd = 2 , lty = lty3)
```

```

R> points(c(2, 3), fib$theta[5:6], col = 3, type = "l", lwd = 2, lty = lty4)
R> points(1:3, fib$theta[1:3], col = 1, cex = 1.5, pch = 21, bg = "white")
R> points(1:3, fib$theta[4:6], col = 3, cex = 1.5, pch = 24, bg = "white")
R> legend("bottom", lty = c(1, 1), pch = c(21, 24), col = c(1, 3), pt.bg = 0,
+       pt.cex = 1.1, lwd = 2, inset = 0.03, cex = 0.9,
+       legend = c("Blacks", "Whites"))

```

The global tests found significant evidence of a decreasing simple order for white women ($p = 0.0150$) but not for black women ($p = 0.1880$). In particular, fibroid growth in older white women was found to be less than that of younger women. Neither of the individual contrasts for the white women (Young-Middle and Middle-Old) were significant at the $\alpha = 0.05$ level. The significant decreasing trend confirms the conclusions of ?.

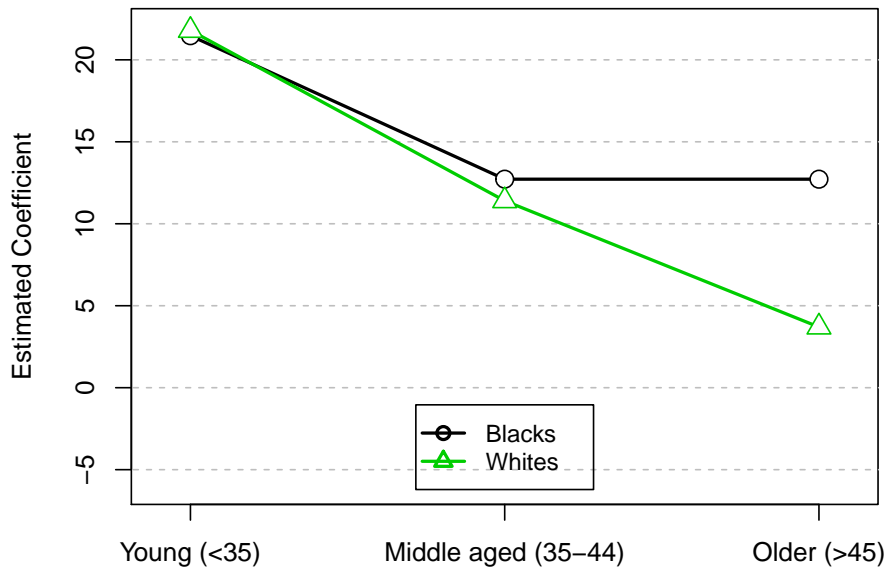


Figure 4: Plot of estimated coefficients of 6-month mean fibroid growth by race and age group. Black lines with circles correspond to Blacks, green lines with triangles correspond to Whites. Growth rates for each fibroid were averaged over the 2-4 time points. None of the individual constraints were significant. The global tests found a significant decreasing trend for white women, but not for black women.

5. Summary

In this paper we have introduced the R package **CLME** for performing statistical tests under linear inequality constraints. It allows the user to choose either the likelihood ratio type

statistic or Williams' type statistic. Since it is based on the residual bootstrap methodology it is not dependant on any Normality assumption. As demonstrated in the paper, the package is simple to implement with default settings (Section 4.1), and more complex hypotheses (Section 4.2) can be accommodated with relatively little effort.

Due to the flexibility and distribution-free nature of the model, as well as the ease of use, we anticipate that many researchers may benefit from using the order-restricted model implemented in **CLME** instead of standard ANOVA models. Other than this package, there does not appear to be any software which offers constrained inference for linear mixed effects models.

While the current release is stable, the authors have an interest in further developing the functionality of **CLME**. There are many potential improvements that we foresee. On the methodological side these include: adding more models, such as logistic models; implementing an automated choice of the number of bootstrap samples (see ?); allowing for correlated random effects; and adding the ability to perform power or sample size calculations. Furthermore, the software does not currently allow for complex covariance structures for the variance components, such as the AR(1) process, although it may be extended to accommodate such structures. Other projected developments include enabling the program to take advantage of parallel processing to speed up the repetitive calculations for each bootstrap sample. Finally, as noted, the **shiny** offers the ability to create apps, making complex models easily available to researchers without the need to write R codes. The included app can be run locally, but **shiny** apps can be hosted on a server and deployed online. A well-designed and web-based application could put the power and flexibility of **CLME** at a researcher's fingertips. Future developments include improving the app and deploying it online.

Acknowledgments

This research is supported (in part) by the Intramural Research Program of the NIH, National Institute of Environmental Health Sciences (Z01 ES101744). The authors thank the following individuals: Drs. Katie O'Brien, Keith Shockley, and Bahjat Qaqish for carefully reading the manuscript and making numerous suggestions which substantially improved the presentation; Dr. Michelle Cora for providing the data on Sprague-Dawley rats analyzed in Section 4.1; and the editors of the Journal of Statistical Software for providing suggestions on streamlining and improving the interface of **CLME**.

A. Flowcharts to determine arguments

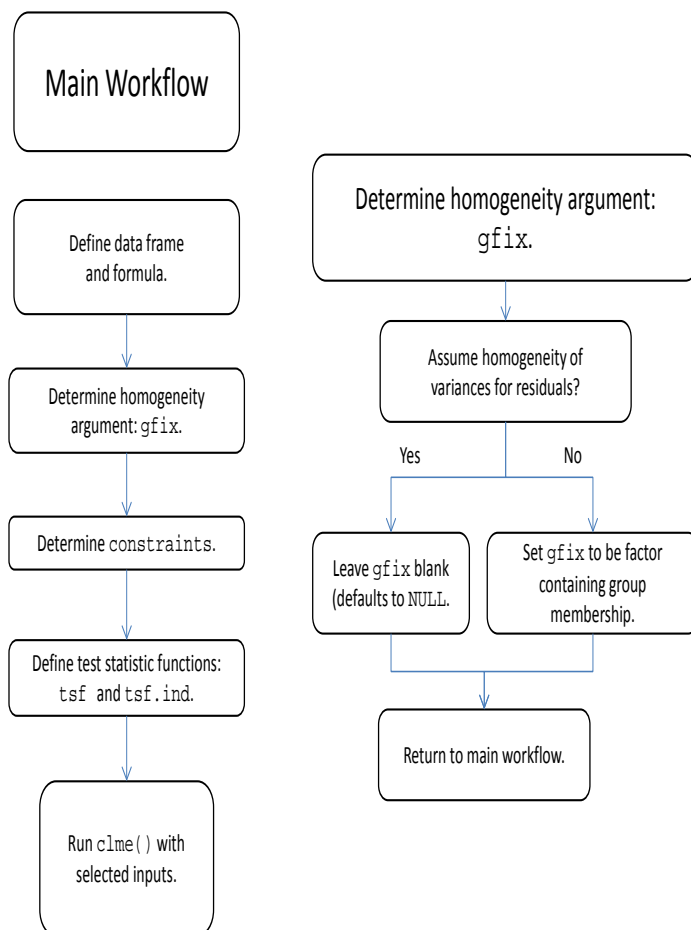


Figure 5: Main flowchart to determine arguments (left) and flowchart to determine groups for residual variance (right).

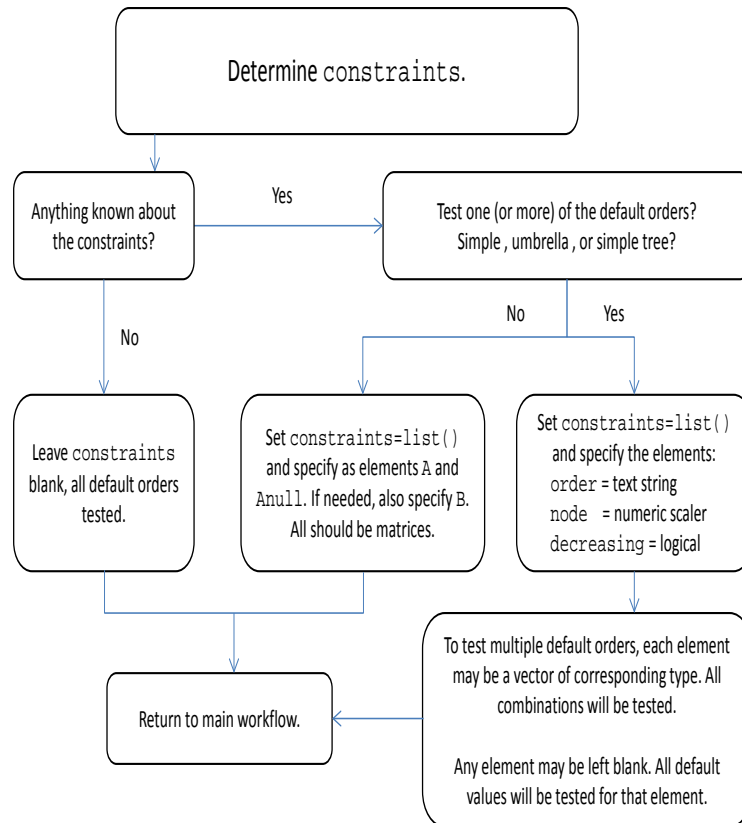


Figure 6: Flowchart to determine constraints.

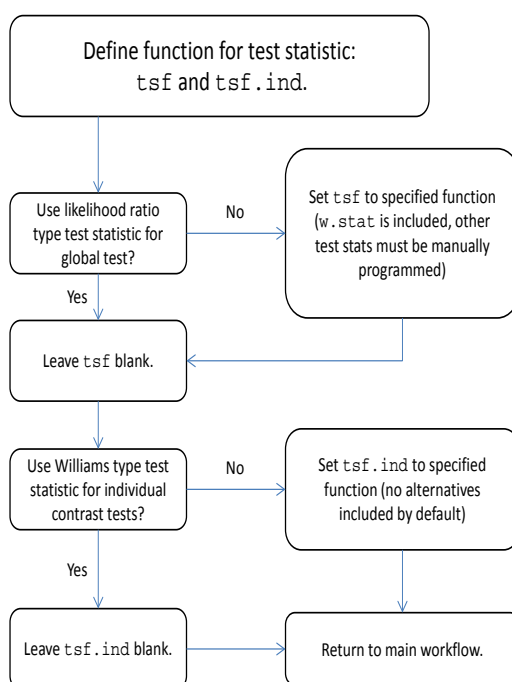


Figure 7: Flowcharts to determine arguments defining the test statistic.

Affiliation:

Casey M. Jelsema

Biostatistics Branch

National Institute of Environmental Health Sciences

111 TW Alexander Dr, RTP, NC, USA

E-mail: casey.jelsema@nih.gov

URL: <http://www.niehs.nih.gov/research/atniehs/labs/bb/staff/jelsema/>

Shyamal D. Peddada

Biostatistics Branch

National Institute of Environmental Health Sciences

111 TW Alexander Dr, RTP, NC, USA

E-mail: peddada@niehs.nih.gov

URL: <http://www.niehs.nih.gov/research/atniehs/labs/bb/staff/peddada/>