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. Thus in this case the null and alternative hypotheses of interest are $H_0 : \theta_1 = \theta_2 = \ldots = \theta_p$, and $H_a : \theta_1 \leq \theta_2 \leq \ldots \theta_p$, 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 (Simpson and Margolin 1986). In such cases, researchers are interested in testing for an umbrella alternative $H_{ai} : \theta_1 \leq \theta_2 \ldots \leq \theta_{i-1} \leq \theta_i \leq \theta_{i+1} \geq \ldots \geq \theta_p$, with at least one strict inequality.

In a multi-center rat uterotophic 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 (Kanno *et al.* 2003). Thus in this case 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 $\{\theta_{control,control} \leq \theta_{control,low} \leq \theta_{control,medium} \leq \theta_{high,high}\} \bigcup \{\theta_{control,control} \leq \theta_{low,control} \leq \theta_{medium,control} \leq \theta_{high,high}\}$, 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 $H_0 : \mathbf{A}\theta = \mathbf{c}$ and $H_a : \mathbf{A}\theta \geq \mathbf{c}$, respectively, where $\mathbf{A}$ is a suitable matrix of zeros, ones and negative ones of appropriate order, $\theta = (\theta_1, \theta_2, \ldots, \theta_p)'$ and $\mathbf{c}$ is a suitable vector of known scalars, for example a vector of zero's. Some examples of $\mathbf{A}$ and $\mathbf{c}$ are provided later, and an illustration of some common orders is given in Figure (1).

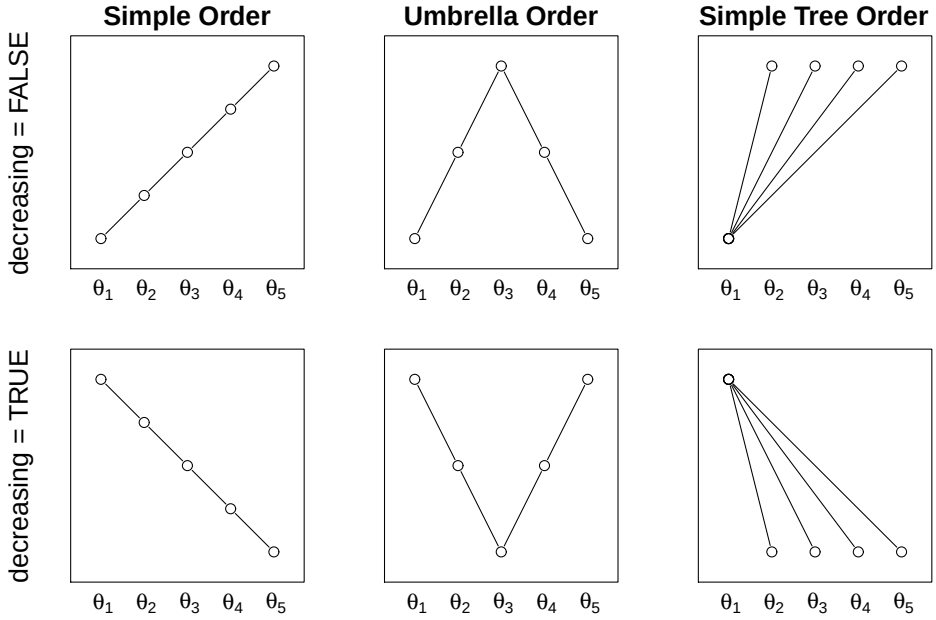

Figure 1: Illustration of order restrictions. Each circle represents a parameter of interest. Inequalities between two parameters (i.e. circles) are provided by the lines. The vertical axis denotes relative magnitude of connected parameters. No relationship (either $<$, $=$, or $>$) is known among parameters 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 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 (Barlow *et al.* 1972; Robertson *et al.* 1988; Silvapulle and Sen 2005), it was only recently that researchers developed methods for performing constrained inference in linear mixed effects models (Davidov and Rosen 2011; Rosen and Davidov 2011; Farnan *et al.* 2014). While Davidov and Rosen (2011) and Rosen and Davidov (2011) developed likelihood ratio based methods, Farnan *et al.* (2014) 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 (SAS Institute Inc. 2011) 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 Farnan *et al.* (2014), statistical methods that are specifically designed for testing inequality constraints are expected to enjoy substantially higher power than the usual omnibus procedures such as the 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** ('**C**onstrainted **L**inear **M**ixed **E**ffects') based on the distribution-free residual bootstrap methodology developed in Farnan *et al.* (2014). There are several packages in R which offer constrained fixed effects models, such as **glmc** (Chaudhuri *et al.* 2006) and **ic.infer** (Grömping 2010), but neither of these offer support for mixed models. Both also assume parametric models: **ic.infer** assumes a normal model, **glmc** offers some flexibility for the error distribution, but still requires a parametric family to be specified. The present work fills this void with a flexible model able to handle fixed or mixed effects models and allows (but does not require) additional, unconstrained, fixed effects. 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 Farnan *et al.* (2014). Section 3 describes the contents of the package CLME along with implementation details. Section 5 provides some illustrative examples using the package, and Section 6 concludes the paper with a summary and some comments on planned developments of **CLME**.

2. Linear Mixed Effect (LME) models under inequality constraints

Let

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

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 $\mathbf{U} = \left(\mathbf{U}_1 : \mathbf{U}_2 : \ldots, : \mathbf{U}_{c_q}\right)$, 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 $\theta = (\theta_1', \theta_2')'$ and $p = p_1 + p_2$.

The random vector $\xi = \left(\xi_1', \xi_2', \ldots, \xi_q'\right)'$ is $c \times 1$, where each ξ_i is a $c_i \times 1$ vector corresponding to $\mathbf{U}_i$, for $i = 1, \ldots, q$. The elements of ξ are independently distributed with mean $\mathbf{0}$ and covariance matrix $\mathbf{T} = \text{diag}\left(\tau_1^2\mathbf{I}_{c_1}, \tau_2^2\mathbf{I}_{c_2}, \ldots, \tau_q^2\mathbf{I}_{c_q}\right)$. The residual term ϵ is similarly defined with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma} = \text{diag}\left(\sigma_1^2\mathbf{I}_{n_1}, \sigma_2^2\mathbf{I}_{n_2}, \ldots, \sigma_k^2\mathbf{I}_{n_k}\right)$, where $i = 1, \ldots, k$ 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 $\mathbf{A}$ be an $r \times p$ matrix so that $\mathbf{A}\theta$ represents the linear combinations which are subject to inequality constraints specified by the alternative hypothesis. Thus the hypotheses of interest are given by:

$$H_o : \mathbf{A}\theta = \mathbf{0} \text{ versus } H_a : \mathbf{A}\theta \geq \mathbf{0}, \tag{2}$$

such that at least one of the r inequalities is strict. $\mathbf{A}$ is represented in block form as $\mathbf{A} = [\mathbf{A}_1 : \mathbf{0}_{r\times p_2}]$, where $\mathbf{A}_1$ is an $r \times p_1$ matrix and $\mathbf{0}_{r\times p_2}$ is a null matrix of size $r \times p_2$ indicating no constraints on the coefficients of any covariate terms. One can specify $\mathbf{A}_1$ to test any desired pattern among the elements of θ_1.

CLME is designed to implement two general classes of statistical tests. The likelihood ratio type (LRT) statistic (Hoferkamp and Peddada 2002; Davidov and Rosen 2011) is the default setting, but the user may instead choose the Williams' type test statistic (Williams 1971, 1977). 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 Farnan *et al.* (2014). 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'. In addition, the constrained estimate of θ_1 can be obtained using either quadratic minimization, or the the algorithm provided in Hwang and Peddada (1994). When the covariance matrix of the unconstrained estimator is diagonal and the means are subject to simple order and the variances are known, then the algorithm provided by Hwang and Peddada (1994) is identical to the classical pool adjacent violators algorithm (PAVA). For convenience of notation, we shall refer to both algorithms as "PAVA".

Using simulations, Farnan *et al.* (2014) demonstrated that the Willaims' 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:

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

where $\odot$ denote the Schur-product of vectors, i.e. $\mathbf{a}\odot\mathbf{b} = (a_1b_1, a_2b_2, \ldots, a_rb_r)'$, $\tilde{\theta}_1$ 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 $\mathbf{A}$, the contrast matrix $\mathbf{B}$ is derived by the largest hypothesized difference; for example in the simple order, the difference between θ_1 and θ_{p_1}. Examples of $\mathbf{A}$ and $\mathbf{B}$ for some specific order restrictions are provided in the next section.

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, `constrained.lme`. Afterwards, we detail some of the secondary functions which users may find useful.

3.1. Main Function

The main function of **CLME** is `constrained.lme`. This function implements the order restricted residual bootstrap test described in Farnan *et al.* (2014). The arguments are listed and described in Table 1. The only required arguments are the method of isotonization, the response vector, and the design matrix (respectively: `method`, `Y`, `X1`). Covariates and random effects, $\mathbf{X}_2$ and $\mathbf{U}$ from equation (1), can be included (respectively: `X2` and `U`), but are not required. A series of flowcharts are provided in the appendix (Figures (6), (7), and (8)) to guide a user through specification of the arguments for `constrained.lme`.

Although under some conditions the PAVA type estimator of Hwang and Peddada (1994) is proved to be efficient (smaller average quadratic loss and higher coverage probability), it is not always straightforward to implement. The quadratic programming estimator (QPE) optimizes the least squares under inequality constraints and is easy to obtain using quadratic programming (function `solve.QP` from package **quadprog**). Furthermore, under the normality assumption, with known covariance matrix, the QPE yields the restricted maximum likelihood estimator. Several of the arguments to `constrained.lme` 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` Scalar indicating which element of θ_1 is the node.

`decreasing` Logical indicating whether the initial order restrictions are increasing or decreasing. The three order restrictions described in section 2 are all increasing. See Figure (1) for an illustration.

`A` The $\mathbf{A}$ matrix containing the order restrictions.

`B` The matrix of coefficients for the Williams type statistic.

As an example, the values of `A` and `B` are shown below. These are for an increasing umbrella order with $p_1 = 5$, no covariates, and a node at $\theta_{1,3}$ (the third element of θ_1).

$$\mathtt{A} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

and,

$$\mathtt{B} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

The alternative hypothesis is $H_a : \theta_1 \leq \theta_2 \leq \theta_3 \geq \theta_4 \geq \theta_5$. The first row of $\mathbf{A}$ thus corresponds to the constraint $\theta_1 \leq \theta_2$, the second row to the constraint $\theta_2 \leq \theta_3$, and so on. Under the umbrella order, the greatest difference in parameters will be between the node and the first or last values, so the Williams' type statistic from Equation (3) takes the form:

Argument	Description
`method`	The method to implement order restriction. Must be text, either 'QPE' or 'PAVA'.
`Y`	The response vector, must be numeric.
`X1`	The design matrix, must be numeric.
`X2`	*(Optional)* Matrix of covariates, must be numeric. Defaults to `NULL`.
`U`	*(Optional)* Matrix of random effects, must be numeric. Defaults to `NULL`.
`Nks`	*(Optional)* Vector of n_i where $i = 1, \ldots, k$. Defaults to the sample size, $N = dim(X1)[1]$.
`Qs`	*(Optional)* Vector of c_i where $i = 1, \ldots, q$. Defaults to $c = dim(U)[2]$.
`constraints`	*(Optional)* List including elements `A`, `B`, `node`, and `decreasing` as described for function `create.constraints`, or the output from that function.
`nsim`	Number of bootstrap samples to generate. Defaults to 1000.
`em.eps`	Convergence criterion for the EM algorithm. Defaults to `sqrt(.Machine$double.eps)`
`em.iter`	Maximum number of iterations the EM algorithm is permitted to run. Defaults to 500.
`mq.eps`	Convergence criterion for the MINQUE algorithm. Defaults to `sqrt(.Machine$double.eps)`
`mq.iter`	Maximum number of iterations the MINQUE algorithm is permitted to run. Defaults to 500.
`verbose`	Vector of 3 logicals. The first causes printing of iteration step, the second two are passed as the `verbose` argument to the functions `minque` and `clme.em`, respectively. Defaults to `rep(FALSE,3)`.
`tsf`	Function to compute the test statistic. Defaults to `w.stat`.
`tsf.ind`	Function to compute the test statistic for individual contrasts. Defaults to `w.stat.ind`.
`pav.alg`	Function to implement the PAVA. Defaults to `NULL`. Ignored when `method='QPE'`.
`hp`	logical to indicate whether weights for PAVA should be the full covariance matrix or just the diagonal elements of the covariance matrix. Ignored when `method='QPE'`.
`seed`	set the seed for the RNG.

Table 1: Arguments for `constrained.lme`

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

hence the **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. The `node` is unnecessary for simple orders or when using the QPE, and `B` is only needed for the Williams type test. If `A` is not specified, **CLME** contains a function `create.constraints` which will be called to generate `A` and `B` using the supplied values of `order`, `node`, and `decreasing`. Each of these elements can be a vector to test for multiple orders or nodes. If any of these are missing, the function will test all possible default values of the missing element(s). Though, because it is not a trend, the simple tree ordering is omitted from this default search pattern. A user may force the program the search to include the simple tree order by setting `constraints$order=c('simple','umbrella','simple.tree')`.

Custom order constraints can be implemented by specifying `A` directly. In this case if the user wishes to use PAVA, a custom function for `pav.alg` is needed. Also, if the Williams test is selected, `B` must be provided. The program will revert to QPE or the LRT, respecitively, if these are not supplied.

The test statistic is taken as the maximum of the test statistics for all these possible orderings, and the program will note this order as the estimated ordering. The bootstrap null distribution of the test statistic is constructed from all the order restrictions under consideration, not just the estimated order (that is, for each bootstrap sample, the test statistic for all candidate orders are computed, and the maximum is taken). For reproducibility, one may use `seed` argument to set the seed for the pseudorandom 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`. For other test statistics, the user may submit a custom function. The related argument `tsf.ind` computes the test statistic to test the individual constraints. The Williams type test, `w.stat.ind`, is default. No other functions are currently available, though a user may submit a custom function.

The full list of arguments available to the test statistic functions is: `theta`, `cov.theta`, `B`, `Y`, `X1`, `X2`, `U`, `tsq`, `ssq`, `Nks`, and `Qs`. Note that `tsq` and `ssq` may be vectors or scalars, depending on the assumption of homogeneity of variances for ξ and ϵ, respectively. The argument `cov.theta` is the $p \times p$ covariance matrix of θ. The output from any custom `tsf` should be a vector or a scalar. Vector output corresponds to multiple global hypotheses being tested, though this should not be used for testing each individual constraint from the **A** matrix, as these are calculated separately using the `tsf.ind` argument. An example of testing multiple global hypotheses is shown in section 5.2, a reanalysis of data from the Fibroid Growth Study (Peddada *et al.* 2008).

PAVA: `pav.alg` There are three functions provided to implement PAVA for simple, umbrella, and simple tree orders. If `A` and `B` were not specified, then `constrained.lme` can

automatically select the proper function to implement PAVA. However, if the user specified a custom `A`, there are two options for estimation under inequality constraints: (1) use QPE, or (2) specify a custom function for the PAVA constraints. The argument `pav.alg` is included for this purpose. If custom constraints are provided and no custom PAVA is supplied, the program will revert to QPE. The PAVA functions, including specification of custom PAVA functions, are described in more detail in section 3.2. Several examples are provided for clarification.

```
> constrained.lme(...,constraints=list(order='simple' ,decreasing=FALSE),...)
```

In this case, the program will automatically generate `A` and `B` and select the appropriate function for PAVA (in this case, `pava.simple.order`).

```
> A.simple = as.matrix(rbind(
>    c(-1 , 1 , 0 , 0),
>    c( 0 ,-1 , 1 , 0),
>    c( 0 , 0 ,-1 , 1)))
> constrained.lme(...,constraints=list(order='simple' , A=A.simple ,
                                       decreasing=FALSE),...)
```

In this case the program will not call `create.constraints`, instead, the provided `A` will be used. When a custom `A` is submitted, the program is 'blind' to the pattern of order restrictions. As a result, even though this `A` is equivalent to a simple order, the program would not be able to generate `B` or select a PAVA algorithm; if either argument is needed, it would need to be submitted manually.

```
> B.simple = matrix( c(-1 , 0 , 0 , 1) , nrow=1 )
> constrained.lme(method='PAVA', tsf=w.stat,
                  constraints=list( A=A.simple , B=B.simple),...)
```

In this case, even though PAVA was selected, the program will note that custom order restrictions are specified, but not a custom `pav.alg`, and so it will revert to using QPE for estimation. However, since `B` was supplied, the Williams type test will be used.

Homogeneity of Variances: `Qs` and `Nks` 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. The arguments `Qs` and `Nks` correspond to this. First, `Qs` is the vector $(c_1, c_s, \ldots, c_q)'$. The default is homogeneity, `Qs = dim(U)[2]`, which models all the elements of ξ with common variance τ^2. Similarly, `Nks` determines the homogeneity of variances for ϵ, and corresponds to the vector $(n_1, n_s, \ldots, n_k)'$. The meaning of this is that the first n_1 elements of $\mathbf{Y}$ share a common variance σ_1^2, the next n_2 elements share a common variance σ_2^2, and so on. Again, the default is homogeneity of variances, setting `Nks = dim(X1)[1]` (that is, equal to N). On the other hand, if there are $N = 50$ observations, and `Nks=c(25,25)`, then the first 25 observations will be modeled with residual variance σ_1^2 and the second 25 will be modeled with residual variance σ_2^2. Typical use of this argument is to model factor levels with separate variance terms.

The output of `constrained.lme` is a list with elements:

`method` the method that was used to apply order constraints (QPE or PAVA).

`theta` vector of estimates of θ.

`ssq` vector of estimates of σ_i^2, $i = 1, \ldots, k$.

`tsq` vector of estimates of τ_i^2, $i = 1, \ldots, q$.

`cov.theta` the covariance matrix of θ.

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

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

`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 (`A`) and the contrast for the global test (`B`).

`est.order` Sentence describing the estimated order (or whether custom constraints were specified).

3.2. Secondary Functions

These are functions that perform an integral role for `constrained.lme` and may be of use as independent functions outside of normal use. For most users these may be ignored as they are automatically called in the background. They are detailed individually to facilitate other functionality (e.g. bootstrapping the residuals, but not necessarily running the EM algorithm and obtaining the test statistic). Most of the arguments are equivalent to arguments passed to `constrained.lme`, so they will not be described in detail.

Residual Bootstrap The function `resid.boot` obtains the bootstrap samples $\mathbf{Y}^*$ of the data response vector. The list of arguments is provided in Table 2; of these, only `mq.phi` is not also an argument to `constrained.lme`. This argument should generally not be used. In typical use of the package, the variance component estimates will have been calculated already and computation time can be saved by passing this as an argument. The initial iterate of the random effect variance component estimates are the Minimum Norm Quadratic Unbiased Estimate (see Rao and Kleffe 1988). If running `resid.boot` separately, the user is suggested to leave this argument blank so that it is automatically calculated. The output of `resid.boot` is a matrix of size $N \times nsim$, where each column is a bootstrap sample $\mathbf{Y}^*$ of the data vector $\mathbf{Y}$.

Constrained Expectation-Maximization Algorithm There are three functions to implement the constrained expectation-maximixzation (EM) algorithm. They are `clme.em.all`, `clme.em.fixed`, and `clme.em.mixed`. The general case is implemented by `clme.em.all`. The other two versions are simplified to handle the more specific situations of fixed effects only or mixed effects. Numerous evaluations of conditional (`if(...)`) are eliminated by having `constrained.lme` check once whether the model is fixed or mixed effects, and selecting the

Argument	Description
`Y`	The response vector, must be numeric.
`X1`	The design matrix, must be numeric.
`constraints`	List containing the order restrictions.
`X2`	*(Optional)* Matrix of covariates, must be numeric. Defaults to `NULL`.
`U`	*(Optional)* Matrix of random effects, must be numeric. Defaults to `NULL`.
`Nks`	*(Optional)* Vector of n_i where $i = 1, \ldots, k$. Defaults to the sample size, $N = dim(X1)[1]$.
`Qs`	*(Optional)* Vector of c_i where $i = 1, \ldots, q$. Defaults to $c = dim(U)[2]$.
`nsim`	*(Optional)* Number of bootstrap samples to generate. Defaults to 1000.
`mq.phi`	*(Optional)* MINQUE estimates of τ_i^2 and σ_j^2, where $i = 1, \ldots, q$ and $j = 1, \ldots, k$. Defaults to `NULL` and is automatically calculated.
`seed`	optional. Set the seed for the RNG.

Table 2: Arguments for `resid.boot`

Argument	Description	
`X1`	The design matrix, must be numeric.	
`X2`	*(Optional)* Matrix of covariates, must be numeric. Defaults to `NULL`	
`constraints`	List of length three with elements:	
	`order`	Text string name of order. One of: simple, umbrella, simple.tree
	`node`	The nodal element for umbrella or simple.tree orders, labeled θ_s. For simple order a node is not required.
	`decreasing`	Logical value that allows order to be reversed. See Figure (1) for illustration.

Table 3: Arguments for `create.constraints`

appropriate function. We mention these internal functions only because they are a primary component of `constrained.lme`. In general they should not be used, because they lack the error-catching ability of `constrained.lme`. If the user simply desires the parameter estimates without running the bootstrap test, it is recommended to run `constrained.lme` with `nsim=0`.

Constraints The function `create.constraints` generates `A` and `B` for simple, umbrella, and simple tree orders. The arguments and their descriptions are given in Table 3.

As an example, constraints for testing the simple order $H_a : \theta_1 \leq \theta_2 \leq \ldots \leq \theta_5$ could be generated using the command:

```
> create.constraints( X1 , constraints=list( order='simple' , decreasing=FALSE))
```

See Figure (1) for an illustration of some common order specifications with the appropriate values of `order` and `decreasing`. The output of `create.constraints` is a list containing the elements of the argument `constraints`, with two additional elements: `A` and `B`, which were described previously.

Argument	Description
`theta`	The estimates of θ_1.
`cov.theta`	The covariance matrix of θ_1.
`node`	The nodal element (for `pava.simple.order`, set `node=NULL`).
`decreasing`	Logical indicating whether the initial order is increasing or decreasing. See Figure (1) for an illustration of this.
`hp`	logical indicating whether the weights should be the full covariance matrix (`TRUE`) or just the diagonal elements (`FALSE`). Default is `FALSE`.

Table 4: Arguments for PAVA functions

PAVA Functions There are three built-in functions to perform the Pool Adjacent Violators Algorithm (PAVA). The functions are `pava.simple.order`, `pava.umbrella`, and `pava.simple.tree` . They implement, respectively, a simple order, umbrella order, and simple tree order. All three functions take the arguments shown in Table 4.

To implement the PAVA for an alternate order restriction, the user must define a custom function, which may be submitted as the argument `pav.alg` to the main function `constrained.lme` or to `clme.em`. The arguments must be equivalent to those shown in Table 4, and the output must be a numeric vector of the same length as the input argument `theta`. Often these may be a problem-specific sequence of calls to the default PAV algorithms. See section 5.2 for an example of a custom `pav.alg`.

3.3. Other package contents

Shiny application The **shiny** package (RStudio and Inc. 2014) offers the ability to develop a graphical user interface (GUI) which implements **CLME**. This GUI can be run locally or deployed online. This would seem to be particularly beneficial to researchers who may not be as familiar with R, but wish to use the methods described here. To this end we have included an application, built in **shiny**, which generates a GUI to implement **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.

Class clme The S3 class `clme` has been defined for objects produced by the package **CLME**, specifically the function `constrained.lme`. The purpose of the class is to facilitate future development of the package, and the display or manipulation of objects produced by the package. The methods defined are as follows.

`clme` Creates an object of class `clme`.

`is` Determine whether an object is of class `clme`. Also available as `is.clme`.

`as` If possible, coerces an object to be of class `clme`. Also available as `as.clme`.

`summary` A function to display the output of `constrained.lme` in a more user-friendly and readable fashion. Results printed by `summary` are the global hypothesis test(s), all tests of individual constraints, and the estimates of θ, σ^2, and τ^2. This method is also

accessible as `summary.clme`. The output will also note the estimated order if specific constraints were not supplied. The individual tests are based on the estimated order.

`plot` Plots the estimated values of θ_1 and denotes the significance of individual constraints. The function is capable of plotting confidence intervals for the parameters which, although they are centred at the constrained estimates, use standard errors of the unconstrained estimates.

4. Examples of 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 in the context of order-restricted inference. Other analysis are intended to exhibit certain features of the package or compare the available options. We emphasize that these analyses are intended as illustration, not scientific reanalyses of the data. Consequently some modeling choices, the assumption of homegeneity of variances in particular, are not be thoroughly investigated.

The data from section 5.1 is included in the package. However, the data in section 5.2 concerns human subjects and cannot be released publically.

4.1. Hematologic Parameters from Sprague-Dawley rats

In a recent study of the effect of amount of time a sample is stored on various hematological parameters, Cora *et al.* (2012) 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 Cora *et al.* (2012) 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) 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 with the following commands. The package **nnet** (Venables and Ripley 2002) is called for converting factor objects to indicator matrices. While **CLME** is not dependent upon it, the function `class.ind` is useful.

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

Hematocrit We illustrate **CLME** using two 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.

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

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`.

We initially use the default arguments as far as possible. This entails assuming homogeneity of variances between the time groups, using the QPE for isotonization, the LRT statistic, and `nsim=1000` bootstrap samples. We use the gender of the rat, and the storage temperature of the sample as covariates in these models. The R code to test case (A) are provided below along with the results. While not shown, we also ran three other models with differing options for isotonization (`method`) and the test statistic (`tsf`). These are for later comparisons of computational time.

```
> Y  <- as.matrix(rat.blood$hct)
> X1 <- class.ind(rat.blood$time)
> U  <- class.ind(rat.blood$id)
> X2 <- cbind( class.ind(rat.blood$temp) , class.ind(rat.blood$sex) )
> X2 <- X2[ , -c(2,4) ]
> # Case (A)
> const <- list( order="simple" , decreasing=FALSE)
> set.seed(42)
> timea <- system.time(
+   hct.a <- constrained.lme(Y=Y, X1=X1, X2=X2, U=U, constraints=const)
+   )[3]
> summary(hct.a)

Global Test:
 W =  44.582     p =  0.0000
 Order was increasing simple order.

Individual Tests:
 Contrast 1: 6 Hour - 0 Hour
    W = 4.862    p = 0.0000
 Contrast 2: 24 Hour - 6 Hour
    W = 0.399    p = 0.1510
 Contrast 3: 48 Hour - 24 Hour
    W = 0.829    p = 0.0550
```

```
 Contrast 4: 72 Hour - 48 Hour
    W = 0.693    p = 0.1080

Theta Coefficients:
 0 Hour  =  39.58
 6 Hour  =  40.92
 24 Hour =  41.01
 48 Hour =  41.19
 72 Hour =  41.34
 Ref     = -0.50
 Female  =  1.83


Variances (ssq = sigma^2 , tsq = tau^2):
 ssq_1 = 1.0314
 tsq_1 = 2.2006
```

```
> plot( hct.a )
```

The program found strong evidence ($p < 0.0001$) 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 erase the constraints from the call to `constrained.lme`. The code and results are given below.

```
> # Case (B)
> timeb <- vector(length=4)
> set.seed(42)
> timeb[1] <- system.time(
+   hct.b1 <- constrained.lme(Y=Y, X1=X1, X2=X2, U=U)
+   )[3]
> summary(hct.b1)

Global Test:
 W =  55.326     p =  0.0000
 Estimated order was increasing umbrella order with node=4.

Individual Tests:
 Contrast 1: 6 Hour - 0 Hour
    W = 5.446    p = 0.0000
 Contrast 2: 24 Hour - 6 Hour
    W = 0.447    p = 0.1820
 Contrast 3: 48 Hour - 24 Hour
    W = 1.317    p = 0.0480
 Contrast 4: 48 Hour - 72 Hour
    W = 0.000    p = 0.9950
```

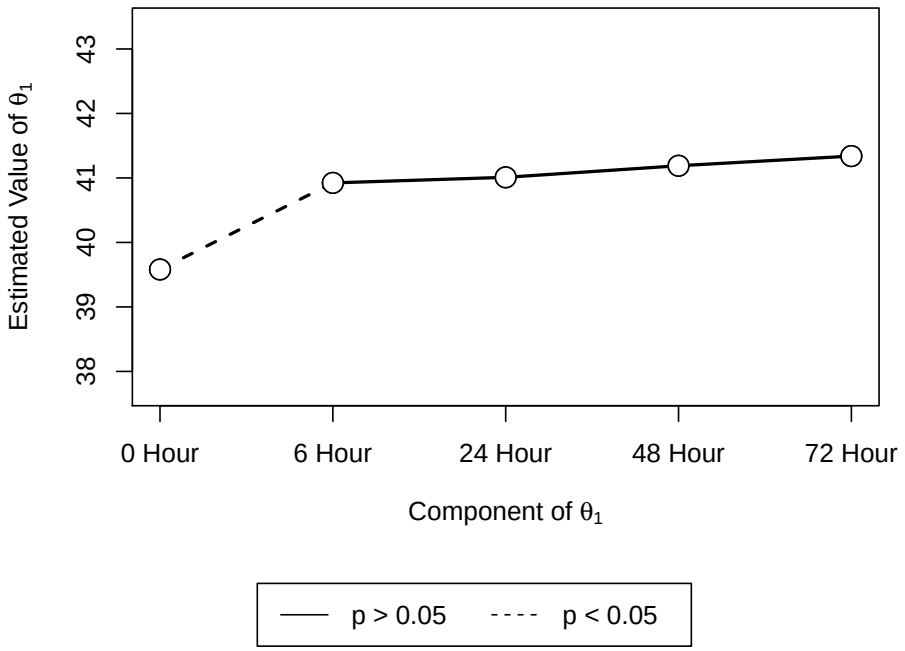

Figure 2: Plot of estimated coefficients of mean hematocrit (HCT) from Case (A). The model assumed an increasing simple order and homogeneity of variances across treatment groups. Solid lines denote no significant difference, while dashed lines denote statistical significance.

```
Theta Coefficients:
 0 Hour  =  39.58
 6 Hour  =  40.92
 24 Hour =  41.01
 48 Hour =  41.26
 72 Hour =  41.26
 Ref     = -0.50
 Female  =  1.83


Variances (ssq = sigma^2 , tsq = tau^2):
 ssq_1 = 0.8221
 tsq_1 = 2.2015
```

Note that in searching for the order, the program determined the order was an increasing *umbrella* order with node at θ_4 (the 48 hour group). Inspecting the coefficient estimates, we see there is no decrease from the node (the estimate for θ_5, the 72 hour group, is equal to that of the 48 hour group). Hence, in reality the result is an increasing simple order, but the

program described it as an umbrella order.

This may seem odd, but is not problematic. For example, suppose the true order is $\theta_{1,1} < \theta_{1,2} < \theta_{1,3} < \theta_{1,4} = \theta_{1,5}$. In this case, the alternative hypotheses of an increasing simple order or an increasing umbrella order with node at θ_4 are indistinguishable. **CLME** selects the order that produces the maximum of the test statistics for all the tested orders. In these data, that occurred for the umbrella order with node of 4.

This also presents an interesting scenario to explore the data and the package. First, we note that this is not particularly dependent upon the method: all four combinations of the method of isotonization and test statistic provide the same estimated order, as seen below.

```
> # Case (B)
> set.seed(42)
> timeb[2] <- system.time(
+   hct.b2 <- constrained.lme( method="QPE", tsf=w.stat, Y=Y, X1=X1, X2=X2, U=U)
+   )[3]
> hct.b2$est.order

[1] "Estimated order was increasing umbrella order with node=4."

> set.seed(42)
> timeb[3] <- system.time(
+   hct.b3 <- constrained.lme( method="PAVA", tsf=lrt.stat, Y=Y, X1=X1, X2=X2, U=U)
+   )[3]
> hct.b3$est.order

[1] "Estimated order was increasing umbrella order with node=4."

> set.seed(42)
> timeb[4] <- system.time(
+   hct.b4 <- constrained.lme( method="PAVA", tsf=w.stat, Y=Y, X1=X1, X2=X2, U=U)
+   )[3]
> hct.b4$est.order

[1] "Estimated order was increasing umbrella order with node=4."
```

To take this a step further, see the boxplot of residuals (from an unconstrained glm) in Figure (3). This provides some indication that the assumption of homogenous variances is not optimal in this case.

We may be able to improve the model by modeling the time groups with individual variances instead of with a pooled variance. Since `X1` is already an indicator matrix, we can accomplish this simply be setting `Nks=colSums(X1)`. We will call this case (C), and again run all four models.

```
> # Case (C)
> timec <- vector( length=4 )
```

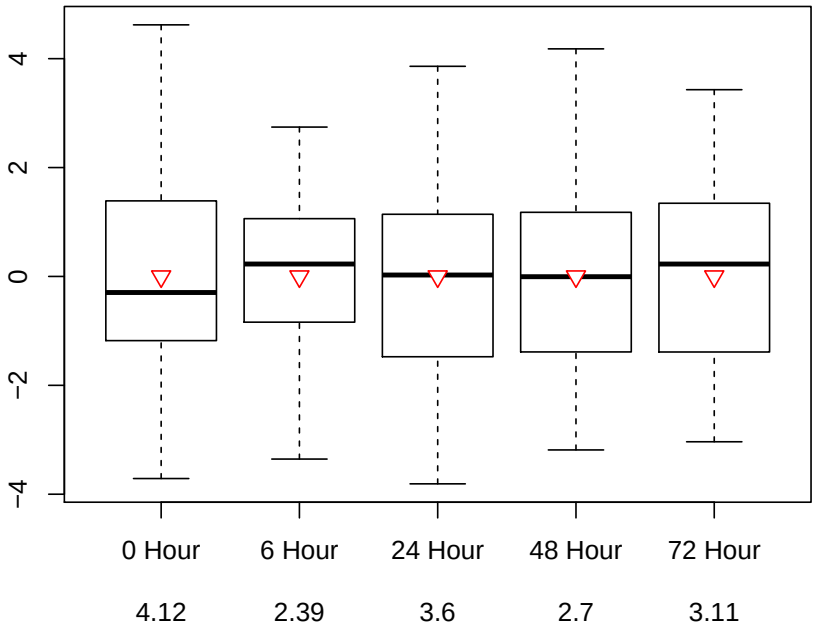

Figure 3: Boxplots of residuals from unconstrained GLM on rat hematocrit. Group means are denoted by red triangles, and variances are given below the labels.

```
> set.seed(42)
> timec[1] <- system.time(
+   hct.c1 <- constrained.lme( method="QPE"  , tsf=lrt.stat ,
+                              Y=Y, X1=X1, X2=X2, U=U , Nks=colSums(X1))
+   )[3]
> hct.c1$est.order

[1] "Estimated order was increasing simple order."

> set.seed(42)
> timec[2] <- system.time(
+   hct.c2 <- constrained.lme( method="QPE"  , tsf=w.stat   ,
+                              Y=Y, X1=X1, X2=X2, U=U , Nks=colSums(X1) )
+   )[3]
> hct.c2$est.order

[1] "Estimated order was increasing simple order."

> set.seed(42)
> timec[3] <- system.time(
+   hct.c3 <- constrained.lme( method="PAVA" , tsf=lrt.stat ,
+                              Y=Y, X1=X1, X2=X2, U=U , Nks=colSums(X1) )
+   )[3]
> hct.c3$est.order

[1] "Estimated order was decreasing umbrella order with node=2."

> set.seed(42)
> timec[4] <- system.time(
+   hct.c4 <- constrained.lme( method="PAVA" , tsf=w.stat   ,
+                              Y=Y, X1=X1, X2=X2, U=U , Nks=colSums(X1) )
+   )[3]
> hct.c4$est.order

[1] "Estimated order was increasing simple order."
```

When modeling the groups with unequal variances, the estimated order is now the increasing simple order for three cases, and a decreasing umbrella order with node of 2 for the PAVA/LRT combination. This highlights the fact that, as with many statistical procedures, heteroskedasticity can affect the results of an analysis. A researcher may test for the equality of variances to verify equality of variances, or may simply assume the more general case of heteroskedasticity.

When using an alternative implementation of PAVA (notably, with weights based on the full covariance matrix of $\hat{\theta}$ instead of just the diagonal), the estimated order for all four combinations was the increasing simple order. This is simply a result of using different estimation techniques.

A factor to consider when selecting which model to run is the computation time necessary. We ran four models for each of cases (A), (B), and (C). These models considered each combination of the method of isotonization and type of test statistic. The runtimes for every model is presented in Table 6. As should be expected, the more complex the model becomes, the more computation time is generally needed. We can also see that the main computational burdens come from estimating the order, assuming heteroskedasticity, and using PAVA instead of the QPE.

Isotonization	Test Statistic	Case (A)	Case (B)	Case (C)
QPE	LRT	15	112	198
QPE	Williams	12	98	184
PAVA	LRT	74	1856	2625
PAVA	Williams	72	1841	2612

Table 5: CPU runtimes (in seconds) for each model of the three cases. Case (A) used fully specified constraints and assumed homogeneity of variances. Case (B) used unspecified constraints and assumed homogeneity of variances. Case (C) used unspecified constraints and assumed heterogeneity of variances. All models used 1000 bootstrap samples.

White Blood Cell Count To illustrate testing for more specific patterns, we mimic one of the tests conducted in Cora *et al.* (2012) by testing for a simple tree order with the 0 hour group as the control group, i.e. the nodal parameter. We assume homogenous variances based on exploratory boxplots. Since we are interested in specifying a control group, we select a simple tree order with the 0 hour group as the node. The R code and results are below, and the coefficients are plotted in Figure (4).

```
> Y  <- as.matrix(rat.blood$wbc)
> X1 <- class.ind(rat.blood$time)
> U  <- class.ind(rat.blood$id)
> X2 <- cbind( class.ind(rat.blood$temp) , class.ind(rat.blood$sex) )
> X2 <- X2[ , -c(2,4) ]
> idx <- rat.blood$temp=="Ref" | rat.blood$time=="0 Hour"
> Yb  <- Y[ idx==TRUE ]
> X1b <- X1[idx==TRUE,]
> X2b <- X2[idx==TRUE, -1,drop=FALSE]
> Ub  <- U[ idx==TRUE,]
> const <- list( order="simple.tree" , node=1 , decreasing=FALSE)
> set.seed(41218)
> clme.wbc <- constrained.lme( Y=Yb , X1=X1b , X2=X2b , U=Ub ,
+                          constraints=const , mq.eps=0.001 , em.eps=0.001 )
> summary(clme.wbc)

Global Test:
 W =  119.117     p =  0.0000
 Order was increasing simple.tree order with node=1.
```

```
Individual Tests:
 Contrast 1: 6 Hour - 0 Hour
    W = 0.000    p = 0.9890
 Contrast 2: 24 Hour - 0 Hour
    W = 1.638    p = 0.0610
 Contrast 3: 48 Hour - 0 Hour
    W = 3.744    p = 0.0000
 Contrast 4: 72 Hour - 0 Hour
    W = 9.259    p = 0.0000

Theta Coefficients:
 0 Hour  =  7.25
 6 Hour  =  7.25
 24 Hour =  7.47
 48 Hour =  7.76
 72 Hour =  8.51
 Female  = -1.84


Variances (ssq = sigma^2 , tsq = tau^2):
 ssq_1 = 0.2034
 tsq_1 = 2.1772


> # plot(clme.wbc)
```

Our results are consistent with those of Cora *et al.* (2012), but we have in addition detected the 0 hour - 48 hour contrast as being significant, which was not identified by Cora *et al.* (2012). There does appear to be an increasing pattern over time, but the differences from control are not statistically significant until sufficient time has passed.

4.2. Fibroid Growth Rates

Peddada *et al.* (2008) 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, Peddada *et al.* (2008) 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.

The interest in this case is to test for a simple order for *each* race using a linear mixed effects model. Thus in this case we define $\mathbf{X}_1$ to be a matrix with $p_1 = 6$ columns, these being indicators for: Young Black, Middle-age Black, Older Black, Young White, Middle-age White, and Older White. This analysis serves as a useful illustration of customizing the order restrictions, because it cannot be performed with the default settings of **CLME**.

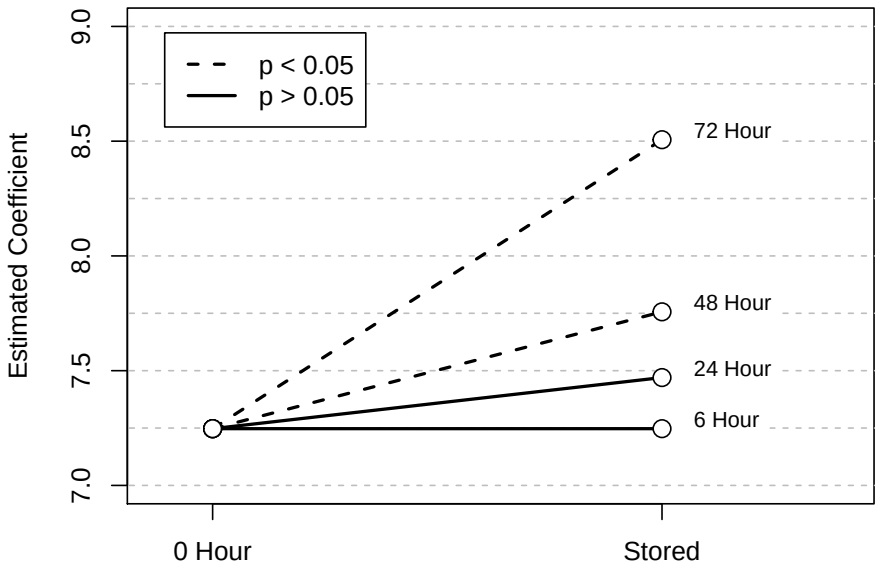

Figure 4: Plot of estimated coefficients of white blood cell (WBC) count. Solid lines denote no significant difference, while dashed lines denote statistical significance. This is plot is not produced by default, but future updates to the package may include it.

We performed our analysis adjusting for all covariates that were considered in Peddada *et al.* (2008). Specifically, these included: Fibroid type (namely, submucosal, subserosal and intramural), location (namely, fundus, corpus and lower segment), parity (binary, parenthood or not), initial fibroid volume ($14 - 65\text{cm}^3$, $\geq 65\text{cm}^3$), BMI ($25 - 30$, ≥ 30), number of fibroids ($1, 2, 3 - 8$), and an intercept term (a column of 1's). Therefore, $\mathbf{X}_2$ has $p_2 = 12$ columns.

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

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 12} \\ \mathbf{0}_{2\times 3} & \mathbf{A}_1 & \mathbf{0}_{2\times 12} \end{bmatrix},$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix},$$

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 $\mathbf{A}_1$ matrix above defines a decreasing simple order for three parameters, which

we must apply individually to the black and white women. Hence, the first row of $\mathbf{A}$ contains $\mathbf{A}_1$ to define the simple order on the black women, then a null matrix to define no constraints for the white women, and a null matrix to define no constraints on the covariates. The second row is constructed similarly, but $\mathbf{A}_1$ is shifted so that it defines constraints for the white women instead of the black women.

Next, we define a function which implements the PAVA separately for the first three elements of θ_1 (blacks) and the last three elements of θ_1 (whites). This is easily done by:

```
> pava.blk.wht <- function( theta , wt , node , decreasing ){
+   coef.blk <- pava.simple.order( theta[1:3] , wt=wt[1:3] , node , decreasing )
+   coef.wht <- pava.simple.order( theta[4:6] , wt=wt[4:6] , node , decreasing )
+   theta <- c( coef.blk , coef.wht )
+   return(theta)
+ }
```

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 Farnan *et al.* (2014) for both blacks and whites separately. The matrix of contrasts is:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & \mathbf{0}_{1\times 12} \\ 0 & 0 & 0 & 1 & 0 & -1 & \mathbf{0}_{1\times 12} \end{bmatrix}.$$

The construction of this matrix is similar to constructing the $\mathbf{A}$ matrix. The first row defines the constrast *Young Black - Older Black* and the second row defines the constrast *Young White - Older White*. A slight modification to the default Williams' type test statistic function (`w.stat`), such that it outputs the test statistic for *both* contrasts instead of the maximum, will allow the program to provide a $p-$value testing for a simple order for both blacks and whites separately.

```
> w.stat2 <- function( theta , cov.theta , B , ... ){
+   cov.contrast <- c( diag( B %*% cov.theta %*% t(B) ) )
+   diffs        <- c( B%*%theta )
+   W.to.return  <- diffs/sqrt(cov.contrast)
+   W.to.return
+ }
```

For simplicity, homogeniety of variances was assumed. Results of the analysis are shown in Figure (5). The global tests found evidence of a decreasing simple order for whites ($p = 0.004$), but not for blacks ($p = 0.300$). This confirms the observation of Peddada *et al.* (2008). Tests on the individual contrasts show that for whites, the decrease from Young to Middle-aged was nearly significant at 5% level of significance ($p = 0.070$), and the decrease from Middle-aged to Older was significant ($p = 0.011$). For blacks, while there was a small decrease, neither Young vs. Middle-aged ($p = 0.107$) nor Middle-aged vs. Older ($p = 1.000$) were statistically significant. The full output from `constrained.lme` is given in Table (7).

5. Summary

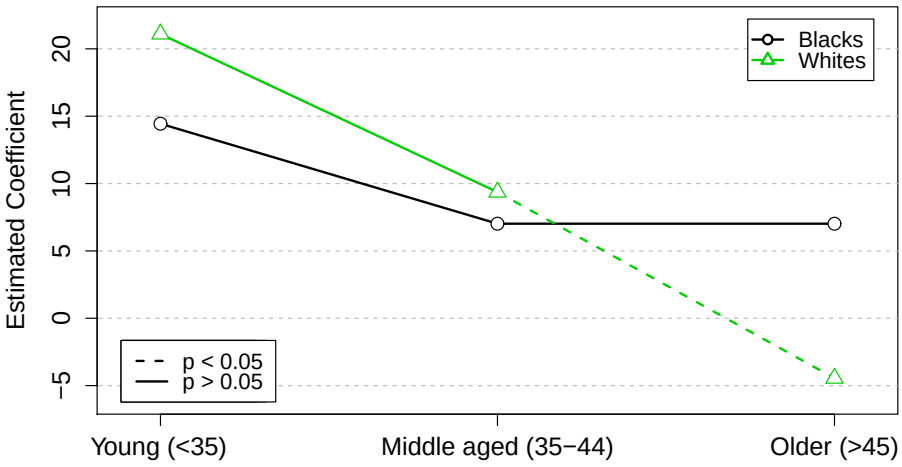

Figure 5: Plot of estimated coefficients of 6-month mean fibroid growth by race and age group. Black lines with circles correspond to Blacks, green lines with triangles correspond to Whites. Solid lines denote no significant difference, dashed lines denote statistical significance. Growth rates for each fibroid were averaged over the 2-4 time points.

In this paper we have introduced the R package **CLME** for performing statistical tests under linear inequality constraints using either the likelihood ratio type statistic or Williams' type statistic. It allows the user to choose between either constrained quadratic programing or PAVA type algorithm to derive the constrained estimates for parameters. Since it is based on the residual bootstrap methodology it alleviates the need for any normality assumption on the data. As demonstrated in the paper, the package is simple to implement with default settings (section 5.1), and that more complex hypotheses (section 5.2) 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 implementing an automated choice of the number of bootstrap samples (see Jiang and Salzman 2012), allowing for correlated random effects, and adding functionality for power calculations. Further, the software does not currently allow for complex covariance structures for the variance components, such as the AR(1) process, although it can 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 development include improving the app and deploying it online.

6. 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 Drs. Katie O'Brien, Keith Shockley, and Bahjat Qaqish for carefully reading the manuscript and making numerous suggestions which substantially improved the presentation.

```
Global Tests:
 W1 = 0.9    p = 0.3000
 W2 = 3.1    p = 0.0040
 Custom order restrictions were specified.

Individual Tests:
 Contrast 1: Yng.Blk - Mid.Blk
    W = 1.1    p = 0.1070
 Contrast 2: Mid.Blk - Old.Blk
    W = 0.0    p = 1.0000
 Contrast 3: Yng.Wht - Mid.Wht
    W = 1.4    p = 0.0700
 Contrast 4: Mid.Wht - Old.Wht
    W = 1.8    p = 0.0110

Theta Coefficients:
 Yng.Blk    =  14.44
 Mid.Blk    =  7.02
 Old.Blk    =  7.02
 Yng.Wht    =  21.10
 Mid.Wht    =  9.36
 Old.Wht    = -4.45
 Subser     =  1.93
 Fundus     = -3.88
 LSegmnt    = -0.90
 Parity     =  5.97
 InitVol.2  = -4.50
 InitVol.3  = -3.30
 BMI.2      = -3.51
 BMI.3      =  2.07
 nFibroid.1 =  30.94
 nFibroid.2 =  2.00
 nFibroid.3 =  7.21

Variances (ssq = sigma^2 , tsq = tau^2):
 ssq_1 = 417.65
 tsq_1 = 110.26
```

Table 6: Output of constrained.lme for fibroid data.

A. Flowcharts to determine arguments

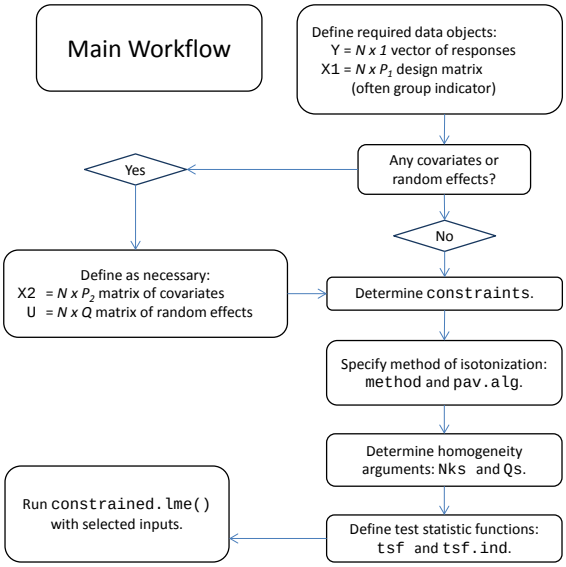

Figure 6: Main flowchart to determine arguments for `constrained.lme`.

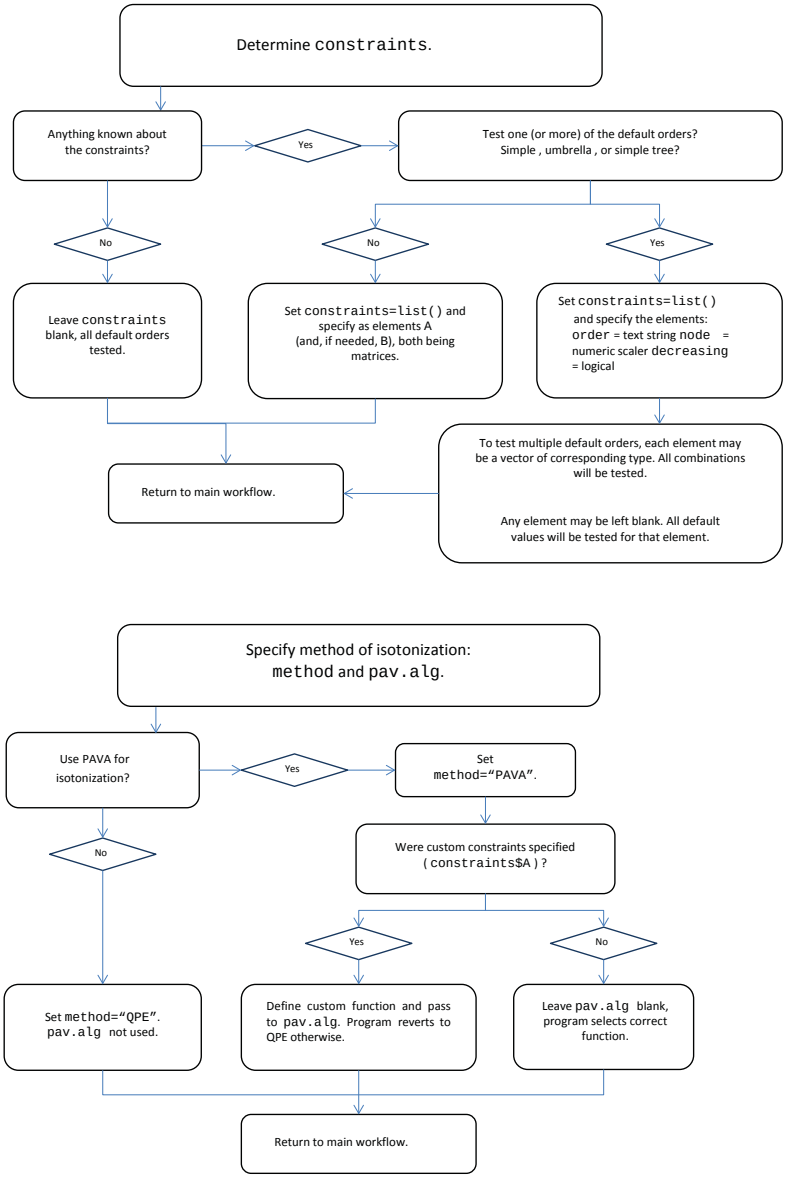

Figure 7: Flowcharts to determine constraints (top) and method of isotonization (bottom).

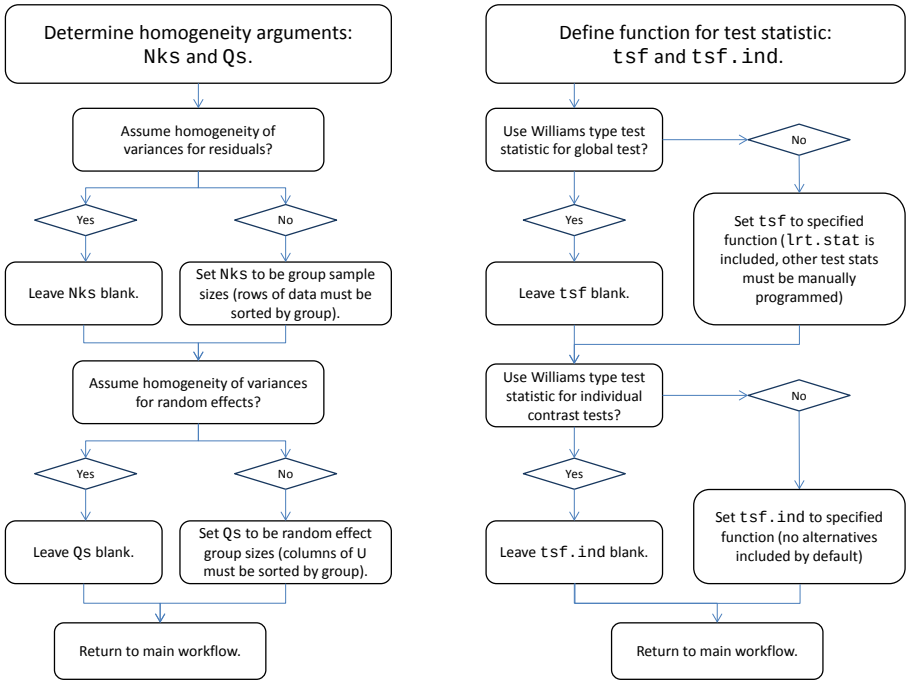

Figure 8: Flowcharts to determine arguments controlling homogeneity and test statistic.

References

Barlow R, Bartholomew J, Brenner J, Brunk H (1972). *Statistical Inference under Order Restrictions*. Wiley, London.

Chaudhuri S, Handcock M, Rendall M (2006). ***glmc**: An R package for generalized linear models subject to constraints*. URL `http://www.stat.washington.edu/handcock/combining`.

Cora M, King D, Betz L, Wilson R, Travlos G (2012). "Artifactual changes in Sprauge-Dawley rat hematologic parameters after storage of samples at 3 °C and 21 °C." *Journal of the American Association for Laboratory Animal Science*, **51**(5), 616–621. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3447451/`.

Davidov O, Rosen S (2011). "Order-restricted inference for multivariate longitudinal data with applications to the natural history of hearing loss." *Statistics in Medicine*, **31**, 1761–1773. `doi:10.1093/biostatistics/kxq051`.

Farnan L, Ivanova A, Peddada S (2014). "Linear mixed effects models under inequality constraints with applications." *PLOS One*, **9**(1). `doi:10.1371/journal.pone.0084778`.

Grömping U (2010). "Inference with Linear Equality and Inequality Constraints Using R: The Package **ic.infer**." *Journal of Statistical Software*, **33**(10), 1–31. URL `http://www.jstatsoft.org/v33/i10/`.

Hoferkamp C, Peddada S (2002). "Parameter estimation in linear models with heteroscedastic variance subject to order restrictions." *Journal of Multivariate Analysis*, **82**, 65–87. `doi:10.1006/jmva.2001.2010`.

Hwang JG, Peddada S (1994). "Confidence interval estimation subject to order restrictions." *The Annals of Statistics*, **22**(1), 67–93.

Jiang H, Salzman J (2012). "Statistical properties of an early stopping rule for resampling-based multiple testing." *Biometrika*, **99**(4), 973–980. `doi:10.1093/biomet/ass051`.

Kanno J, Onyon L, Peddada S, Ashby J, Jacob E, Owens W (2003). "The OECD program to validate the rat uterotrophic bioassay phase 2: dose-response studies." *Environmentral Health Perspectives*, **111**(12).

Peddada S, Laughlin S, Miner K, Guyon JP, Haneke K, Vahdat H, Semelka R, Kowalik A, Armao D, Davis B, Baird D (2008). "Growth of uterine leiomyomata among premenopausal black and white women." *Proceedings of the National Academy of Sciences of the United States of America*, **105**(50), 19887–19892. `doi:10.1073/pnas.0808188105`. URL `http://www.pnas.org/content/105/50/19887.full.pdf`.

Rao C, Kleffe J (1988). *Estimation of variance components and applications*. North-Holland series in statistics and probability. North-Holland.

Robertson T, Wright F, Dykstra R (1988). *Order Restricted Statistical Inference*. Wiley, New York.

Rosen S, Davidov O (2011). "Constrained inference in mixed-effects models for longitudinal data with application to hearing loss." *Biostatistics*, **12**(2), 327–340. doi:10.1002/sim.5335.

RStudio, Inc (2014). *shiny: Web Application Framework for R*. R package version 0.10.1, URL http://CRAN.R-project.org/package=shiny.

SAS Institute Inc (2011). *The SAS/STAT System, Version 9.3*. Cary, NC. URL http://www.sas.com/.

Silvapulle M, Sen P (2005). *Constrained statistical inference: order, inequality, and shape constraints*. Wiley, Hoboken, NJ.

Simpson D, Margolin B (1986). "Recursive nonparametric testing for dose-response relationships subject to downturns at high doses." *Biometrika*, **73**(3), 589–596.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Williams D (1971). "A Test for Differences between Treatment Means When Several Dose Levels are Compare with a Zero Dose Control." *Biometrics*, **27**(1), 103–117.

Williams D (1977). "Some inference procedures for monotonically ordered normal means." *Biometrika*, **64**(1), 9–14.

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/