

# Inverse Estimation with Linear Mixed-Effects Models

with Application in R

Brandon M. Greenwell

January 15, 2017

## Abstract

Inverse estimation, more commonly known as calibration, is a classical and well-known problem in regression. In simple terms, it involves the use of an observed value of the response (or specified value of the mean response) to make inference on the corresponding unknown value of the explanatory variable. In this paper, we describe how to calculate approximate calibration confidence intervals for the unknown value of the explanatory variable in linear mixed-effects models using a parametric bootstrap. Coverage probability for asymptotic intervals is estimated as a function of sample size using Monte-Carlo simulations. An example is given using the R programming language with a real data set.

## Introduction

This paper concerns inverse estimation with linear mixed-effects models (LMMs); however, the methods presented here can be applied to nonlinear mixed-effects models (NLMMs) and generalized linear mixed-effects models as well. Consider an ordinary regression model  $\mathcal{Y}_i = f(x_i; \beta) + \epsilon_i$  ( $i = 1, \dots, n$ ), where  $f$  is a known expectation function (called a *calibration curve*) that is monotonic over the range of interest and  $\epsilon_i \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$ . A common problem in regression is to predict a future response  $\mathcal{Y}_0$  (or estimate the mean response  $f_0 = E[\mathcal{Y}_0]$ ) for a known value of the explanatory variable  $x_0$ . Often, however, there is a need to do the reverse; that is, given an observed value of the response  $\mathcal{Y} = y_0$  (or a specified value of the mean response), infer the unknown value of the explanatory variable  $x_0$ . This is known as the *calibration problem*, though we refer to it more generally as inverse estimation. A thorough overview of the calibration problem is given in Osborne (1991). Oman (1998) considers the case of a random intercept and slope model.

## Linear mixed-effects models

In practice, multiple observation are often taken from the same subject. This type of data is called *repeated measures* data. This includes, for example, *longitudinal* data or *panel* data (experimental units are observed over time). The one feature to remember about repeated measures is that the individual observations are no longer independent. This feature must be taken into account in order to obtain valid standard error, confidence intervals, etc. One of the most common and flexible ways for handling repeated measures data is to use LMMs.

LMMs (i.e., linear regression models with random coefficients) can be represented in many different (but equivalent) forms. One of the most common forms, attributed to Laird and Ware (1982), is

$$\mathcal{Y}_i = \mathbf{X}_i \beta + \mathbf{Z}_i \mathbf{b}_i + \epsilon_i, \quad i = 1, \dots, m, \quad (1)$$

where

- $\mathcal{Y}_i$  is an  $n_i \times 1$  response vector for the  $i$ -th subject/cluster/group;
- $\mathbf{X}_i$  is an  $n_i \times p$  design matrix for the fixed-effects;
- $\mathbf{Z}_i$  is an  $n_i \times q$  design matrix for the random-effects;
- $\beta$  is a  $p \times 1$  vector of fixed-effects coefficients;
- $\mathbf{b}_i$  is a  $q \times 1$  vector of random-effects coefficients with mean zero and variance-covariance matrix  $\mathbf{D}$ ;
- $\mathbf{D}$  is a  $q \times q$  variance-covariance matrix for the random-effects;
- $\epsilon_i$  is an  $n_i \times 1$  vector of random errors with mean zero and variance-covariance matrix  $\sigma^2 \mathbf{I}$ .

Equation (1) (along with the above assumptions) is known as a linear mixed-effects model (LMM). The random-effects and errors are often assumed to follow a normal distribution. By stacking the data, the (normal) LMM can be written concisely as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}, \quad \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \sigma^2 \mathbf{I} \end{bmatrix} \right),$$

where  $\mathbf{y} = \text{col} \{ \mathbf{y}_i \}$ ,  $\mathbf{X} = \text{col} \{ \mathbf{X}_i \}$ ,  $\mathbf{Z} = \text{diag} \{ \mathbf{Z}_i \}$ ,  $\mathbf{b} = \text{col} \{ \mathbf{b}_i \}$ , and  $\boldsymbol{\epsilon} = \text{col} \{ \boldsymbol{\epsilon}_i \}$  for  $i = 1, \dots, m$ . Since  $\text{COV} [\mathbf{b}, \boldsymbol{\epsilon}] = \mathbf{0}$ , it is assumed that the random vectors  $\{ \mathbf{b}_i, \boldsymbol{\epsilon}_i \}_{i=1}^m$  are mutually independent.

The additional term  $\mathbf{Z}\mathbf{b}$  in the model imposes a specific variance-covariance structure on the response vector  $\mathbf{y}$ :

$$\mathbf{y} \sim \mathcal{N} (\mathbf{X}\boldsymbol{\beta}, \mathbf{V}), \quad \mathbf{V} = \mathbf{Z}\mathbf{D}\mathbf{Z}^\top + \sigma^2 \mathbf{I}.$$

Thus, the fixed-effects determine the mean of  $\mathbf{y}$ , while the random-effects govern the variance-covariance structure of  $\mathbf{y}$ . Different random-effects structures impose different variance-covariance structures on the response resulting in a highly flexible framework for modelling *grouped data*.

The random-effects variance-covariance matrix  $\mathbf{D}$  has at most  $q(q+1)/2$  unique elements which we represent by the vector  $\boldsymbol{\theta}$ . There are a number of methods available for estimating  $(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})$ ; see, for example, McCulloch, Searle, and Neuhaus (2008), chap. 6, and Demidenko (2013), chap. 2. Most commonly, the fixed-effects  $\boldsymbol{\beta}$  are estimated via the method of maximum likelihood (ML), while the variance components  $(\sigma^2, \boldsymbol{\theta})$  are estimated via restricted maximum likelihood (REML). The ML estimator of  $\boldsymbol{\beta}$ , given by

$$\hat{\boldsymbol{\beta}} = \left( \mathbf{X}^\top \hat{\mathbf{V}}^{-1} \mathbf{X} \right)^{-1} \mathbf{X}^\top \mathbf{y},$$

depends on the estimated variance components through  $\hat{\mathbf{V}}$  which makes it difficult to capture the variability of  $\hat{\boldsymbol{\beta}}$  in small sample sizes (see McCulloch, Searle, and Neuhaus 2008, 165–67). The usual practice is to ignore the variability of the estimated variance components when making inference about the fixed-effects; that is, treat  $\hat{\mathbf{V}}$  as the true (fixed) value of  $\mathbf{V}$ . Modern computational procedures such as the parametric bootstrap and Markov chain Monte Carlo (MCMC) methods are two ways of accounting for the variability of the estimated variance components.

## Point estimation

The standard methods of calibration, (i.e., the Wald-based and inversion confidence intervals) are easily extended to the case of random coefficients. For convenience, let us write the linear random coefficient model as

$$\mathcal{Y}_{ij} = f(x_{ij}; \boldsymbol{\beta}) + R(x_{ij}; \mathbf{b}_i) + \epsilon_{ij},$$

where  $f(\cdot)$  and  $R(\cdot)$  are linear in  $\boldsymbol{\beta}$  and  $\mathbf{b}_i$ , respectively. For instance, the model for the transformed bladder data has  $f(V_{ij}; \boldsymbol{\beta}) = \beta_0 + \beta_1 V_{ij}$  and  $R(V_{ij}; \mathbf{b}_i) = b_{0i} + b_{1i} V_{ij}$  with  $E[R(V_{ij}; \mathbf{b}_i)] = 0$  and  $\text{VAR}[R(V_{ij}; \mathbf{b}_i)] = \theta_0^2 + V_{ij}^2 \theta_1^2$ .

Assume that, after the data are collected and a model is fitted, we obtain a new observation, denoted  $\mathcal{Y}_0$ , from the same population under study for which the value of the explanatory variable  $x_0$  is unknown. We assume that the new observation belongs to a group not included in our analysis. Estimating  $x_0$  is rather straightforward. By assumption, the new observation  $\mathcal{Y}_0$  is distributed as a  $\mathcal{N} \{ f(x_0; \boldsymbol{\beta}), \sigma_0^2 \}$  random variable with  $\sigma_0^2 = \text{VAR}[R(x_0; \mathbf{b}_0)] + \sigma^2$ . A natural estimator for  $x_0$  is then

$$\hat{x}_0 = f^{-1}(\mathcal{Y}_0; \hat{\boldsymbol{\beta}}), \quad (2)$$

where  $\hat{\boldsymbol{\beta}}$  is the ML estimator of  $\boldsymbol{\beta}$ . We shall refer to Equation (2) as the classical estimator. Note that the point estimate  $\hat{x}_0$  does not involve any of the random-effects; the random-effects only contribute to the variance-covariance structure of the response. Further arguments for the use of (2) as an estimate for  $x_0$  are given in Greenwell (2014), Section 5.2.

## Wald interval

An approximate  $100(1 - \alpha)\%$  Wald-type confidence interval for  $x_0$  has the simple form

$$CI_{wald}(x_0) = (\hat{x}_0 - SE[\hat{x}_0] \Phi(\alpha/2), \hat{x}_0 + SE[\hat{x}_0] \Phi(1 - \alpha/2)). \quad (\#)$$

There is no “textbook” formula for the standard error of  $\hat{x}_0$ —not even in the case of the simple linear regression model. Instead, an estimate of the standard error can be obtained using a first-order Taylor series approximation, or better yet, a parametric bootstrap approximation.

The Taylor series approximation relies on the variance-covariance matrix of  $(\mathcal{Y}_0, \hat{\beta})$ , namely,

$$\Sigma = \begin{bmatrix} VAR[\mathcal{Y}_0] & \mathbf{0} \\ \mathbf{0} & VAR[\hat{\beta}] \end{bmatrix} = \begin{bmatrix} \sigma_0^2 & \mathbf{0} \\ \mathbf{0} & (\mathbf{X}^\top \mathbf{V}^{-1} \mathbf{X})^{-1} \end{bmatrix}.$$

Since  $\mathcal{Y}_0$  is independent of  $\mathcal{Y}$ , it is also independent of  $\hat{\beta}$ , hence the diagonal structure of  $\Sigma$ . Recall that our point estimate has the form  $x = f^{-1}(y; \beta)$ . Let  $f_1^{-1}(y; \beta)$  and  $f_2^{-1}(y; \beta)$  denote the partial derivatives of  $f^{-1}$  with respect to the parameters  $y$  and  $\beta$ , respectively. Our point estimator is given by  $f^{-1}(\mathcal{Y}_0; \hat{\beta})$ , where  $\mathcal{Y}_0$  is a new observation and  $\hat{\beta}$  is the ML estimator of  $\beta$ . A first-order Taylor-series approximation for the variance of  $\hat{x}_0$  is given by

$$VAR[\hat{x}_0] = \left[ f_1^{-1}(\mathcal{Y}_0; \hat{\beta}) \right]^2 \sigma_0^2 + \left[ f_2^{-1}(\mathcal{Y}_0; \hat{\beta}) \right]^\top (\mathbf{X}^\top \mathbf{V}^{-1} \mathbf{X})^{-1} \left[ f_2^{-1}(\mathcal{Y}_0; \hat{\beta}) \right] \quad (\#).$$

To obtain  $SE[\hat{x}_0] = \left\{ \widehat{VAR}[\hat{x}_0] \right\}^{1/2}$ , we simply replace  $\sigma_0^2$  and  $\mathbf{V}$  with their respective estimates  $\hat{\sigma}_0^2$  and  $\hat{\mathbf{V}}$ .

The Wald-based interval (#) is simple to compute as long as we have an estimate for the standard error. As we will discuss in Section **ADD SECTION NUMBER**, the R package `investr` (Greenwell 2013) can be used to obtain the Wald-based interval (#) using a Taylor series approximation of the standard error based in Equation (#). If a closed-form formula is available for  $\hat{x}_0$ , then the `deltaMethod` function from the `car` package (Fox and Weisberg 2011) can also be used to obtain the Taylor series approximation of the standard error. Alternatively, one can use the parametric bootstrap instead of relying on a Taylor series approximation; see Section **ADD SECTION NUMBER**. The bootstrap estimate may be more accurate in smaller sample sizes because, unlike the Taylor series approximation, it takes into account the variability of the estimated variance components. The `bootmer` function in the `lme4` package (D. Bates et al. 2014) can be used for model-based parametric bootstrapping in mixed-effects models; see Section **ADD SECTION NUMBER**.

## Inversion method

In the case of the simple linear regression model with constant variance, an exact  $100(1 - \alpha)\%$  confidence interval for  $x_0$  can be derived (Graybill 1976). This can be generalized to an approximate method in the case of polynomial or nonlinear regression models with independent observations and constant variance (see Seber and Wild (2003) Huet (2004)). In a similar fashion, we can generalize the same results to an approximate method for linear mixed-effects models.

Let  $\hat{f}_0 = f(\mathcal{Y}_0; \hat{\beta})$  be the predicted mean at  $x = x_0$ . A prediction interval for  $\mathcal{Y}_0$  at  $x_0$  with asymptotic coverage probability  $100(1 - \alpha)\%$  is

$$\mathcal{I}_\infty(x_0) = \hat{f}_0 \pm z_{1-\alpha/2} \left\{ \widehat{VAR}[\mathcal{Y}_0 - \hat{f}_0] \right\}^{1/2}. \quad (\#)$$

If instead,  $\mathcal{Y}_0$  is observed to be  $y_0$  and  $x_0$  is unknown, then an asymptotic  $100(1 - \alpha)\%$  confidence interval for the unknown  $x_0$  can be obtained by inverting (#):

$$CI_{inv}(x_0) = \left\{ x : z_{\alpha/2} \leq \frac{\mathcal{Y}_0 - f(x; \hat{\beta})}{\left\{ \widehat{VAR}[\mathcal{Y}_0 - f(x; \hat{\beta})] \right\}^{1/2}} \leq z_{1-\alpha/2} \right\}. \quad (\#)$$

This is known as the *inversion interval* and typically cannot be written in closed-form; therefore, numerical techniques are required to find the lower and upper bounds. Further, note that  $CI_{inv}(x_0)$  is not symmetric about  $\hat{x}_0$  and will not necessarily result in a single finite interval (see, e.g., Greenwell and Kabban 2014, 91–92).

Fortunately, the inversion interval (#) can be computed automatically using the `investr` package. However, like the Wald-based interval (?), the inversion interval ignores the variability of the estimated variance components and will likely perform poorly in small sample sizes. An alternative approach involving the parametric bootstrap will be discussed in Section ?.

Finally, the inversion interval uses a normal approximation. While it is likely that a  $t$ -distribution may be more accurate, it is difficult to find the appropriate degrees of freedom. Oman (1998), suggests a  $t$ -distribution with  $N - 1$  degrees of freedom ( $N$  being the total sample size).

## Parametric bootstrap

The bootstrap (Efron 1979) is a general-purpose computer-based method for assessing accuracy of estimators and forming confidence intervals for parameters. Jones and Rocke (1999) proposed a nonparametric bootstrap algorithm for controlled calibration with independent observations. However, since our application involves random coefficients (i.e., dependent observations), the nonparametric bootstrap does not easily apply, and instead, we adopt a “fully parametric” approach. In a parametric bootstrap, bootstrap samples are generated from a fitted parametric model rather than sampling with replacement directly from the data. Fortunately, parametric bootstrap confidence intervals are usually more accurate than nonparametric ones, however, by sampling from a fitted parametric family, we are implicitly assuming that we have the “correct model”.

Let  $\hat{\sigma}_0^2$  be an estimate of the variance of the new observation  $\mathcal{Y}_0$ . An algorithm for bootstrapping  $\hat{x}_0$  in an LMM is given in Figure ?. Note that step 5. is crucial for calibration problems because we need to treat  $y_0$  as a random quantity in the bootstrap simulation, otherwise the variability of  $\hat{x}_0$  will be underestimated; see, for example, Jones and Rocke (1999) and Greenwell and Kabban (2014).

- Fit an LMM (1) to the data and obtain estimates  $\hat{\beta}$ ,  $\hat{D}$ , and  $\hat{\sigma}^2$ .
  - (1) Define  $\mathbf{y}^* = \mathbf{X}\hat{\beta} + \mathbf{Z}\mathbf{b}^* + \boldsymbol{\epsilon}^*$ , where  $\mathbf{b}^* \sim \mathcal{N}_q(\mathbf{0}, \hat{D})$  and  $\boldsymbol{\epsilon}^* \sim \mathcal{N}_N(\mathbf{0}, \hat{\sigma}_\epsilon^2 \mathbf{I})$ ;
  - (2) Update the original model using  $\mathbf{y}^*$  as the response vector to obtain  $\hat{\beta}^*$  and  $\hat{\sigma}_0^{2*}$ ;
  - (3) Generate  $y_0^* \sim \mathcal{N}(y_0, \hat{\sigma}_0^{2*})$ ;
  - (4) Define  $\hat{x}_0^* = f^{-1}(y_0^*; \hat{\beta}^*)$ ;
- Repeat steps (1)-(4)  $R$  times.

There are three main bootstrap confidence interval procedures: The percentile methods introduced in `efron-bootstrap-1979`, the studentized bootstrap `$t$` method introduced in `efron-jackknife-1982`, and the double bootstrap method (Hall 1986). For a good overview of all these confidence interval procedures, see Davison and Hinkley (1997), chap. 5, and Boos and Stefanski (2013), chap. 11. In the next section, we discuss how to use this algorithm to adjust the previously discussed inversion interval.

Rather than using the parametric bootstrap to estimate the sampling distribution of  $\hat{x}_0$  it is possible to adopt a Bayesian approach to obtain the posterior distribution of  $x_0$ . However, as discussed in Hoadley (1970) (and the commenting articles), finding a prior for  $x_0$  is not always straightforward, even in the case with independent observations. For the bladder volume example, it seems that any prior with positive support would be reasonable.

## A bootstrap adjusted inversion interval for $x_0$

Huet (2004) suggests a bootstrap modification of the usual inversion interval in nonlinear regression models with dependent data. In a similar fashion, we could use the parametric bootstrap to adjust the approximate

inversion interval given in Equation (?). The inversion interval assumes that the *predictive pivot*

$$Q_I = \frac{\mathcal{Y}_0 - f(x; \hat{\beta})}{\left\{ \widehat{VAR} \left[ \mathcal{Y}_0 - f(x; \hat{\beta}) \right] \right\}^{1/2}} \sim \mathcal{N}(0, 1).$$

A bootstrap modified inversion interval would then use the bootstrap distribution of

$$Q_I^* = \frac{\mathcal{Y}_0^* - f(\hat{x}_0; \hat{\beta}^*)}{\left\{ \widehat{VAR} \left[ \mathcal{Y}_0 - f(\hat{x}_0; \hat{\beta}^*) \right] \right\}^{1/2}},$$

to estimate the true distribution of  $Q_I$ . If  $\hat{F}_{Q_I}$  is the empirical distribution function for a sample of  $R$  bootstrap replicates of  $Q_I$ , then the modified inversion interval for  $x_0$  is given by

$$CI_{inv}^*(x_0) = \left\{ x : \hat{F}_{Q_I}(\alpha/2) \leq Q_I \leq \hat{F}_{Q_I}(1 - \alpha/2) \right\}.$$

## Bladder volume example

For illustration, let us consider the bladder volume data which can be found in Brown (1993), pg. 7, and Oman (1998). In Brown’s words:

“A series of 23 women patients attending a urodynamic clinic were recruited for the study. After successful voiding of the bladder, sterile water was introduced in additions of 1, 1.5, and then 2.5 cl increments up to a final cumulative total of 17.5 cl. At each volume a measure of height ( $H$ ) in mm and depth ( $D$ ) in mm of largest ultrasound bladder images were taken. The product  $H \times D$  was taken as a measure of liquid volume.”

We took Brown’s suggestion and transformed the data so that the relationship is approximately linear. A spaghetti plot of the transformed data is displayed in the left side of Figure 1. The right side of Figure 1 displays the estimated coefficients from individual linear regression models fit to each subject (the predictor  $V$  has been centered to remove correlation between the intercepts and slopes); the bars represent one-at-a-time 95% confidence intervals.

As indicated by Figure #, the intercept and slopes vary between subjects; hence, the following random intercept and slope model seems appropriate:

$$HD_{ij}^{3/2} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) V_{ij} + \epsilon_{ij}$$

$$b_{ki} \sim \mathcal{N}(0, \theta_k^2), k = 0, 1,$$

$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

Here we also assume that the random effects are uncorrelated (i.e.,  $COV[b_{0i}, b_{1i}] = 0$ ). Table 1 displays the fixed-effects results from applying this model to the transformed bladder volume data. To fit such a model in R, we can use the recommended `nlme` package (Jose Pinheiro et al. 2013):

```
bladder.lme <- lme(HD ~ (3 / 2) ~ volume, data = bladder,
  random = list(subject = pdDiag( ~ volume)))
```

Table 1: Fixed-effects t-table for the random intercept and slope model fit to the bladder volume data.

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-53.8	11.60	142	-4.64	0
volume	69.1	3.11	142	22.19	0

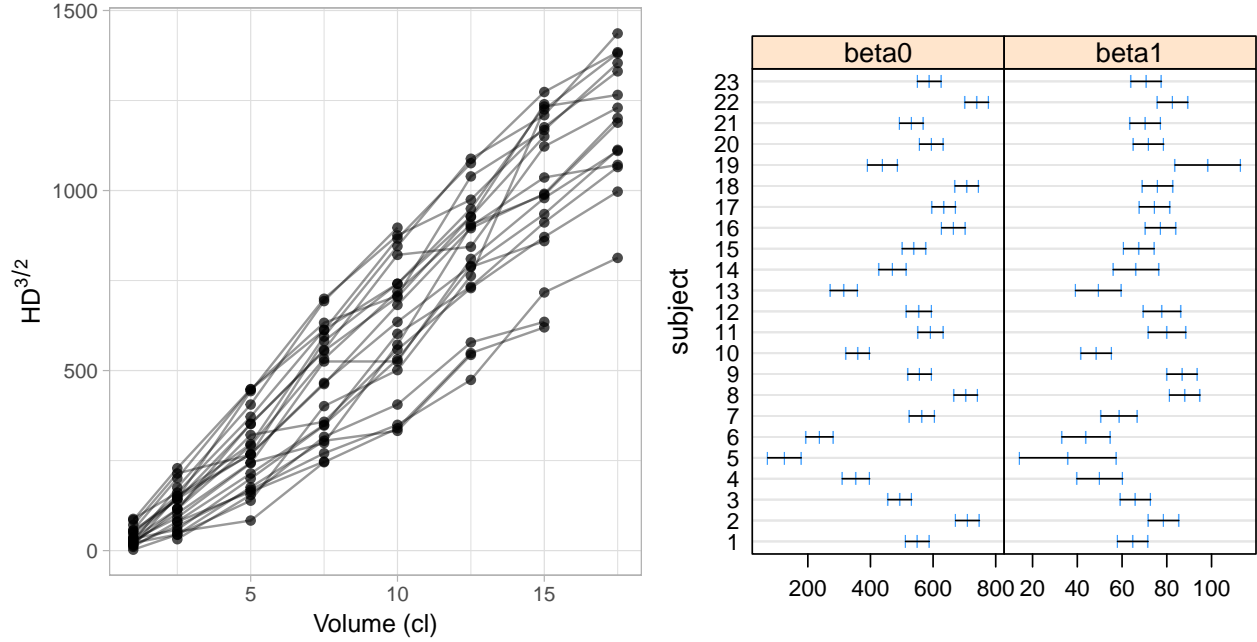


Figure 1: Bladder volume data. Left: spaghettiplot of transformed data (lines connect measurements belonging to the same subject). Right: Estimated subject specific intercepts and slopes (with one-at-a-time 95% confidence limits).

The `pdDiag` function forces a diagonal variance-covariance structure on the random-effects; hence, a covariance of zero. For an in-depth treatment on fitting LMMs using the `nlme` software, see J.C. Pinheiro and Bates (2000).

### The `investr` package

The R package `investr` facilitates calibration/inverse estimation with linear and nonlinear regression models. The main function, `invest`, can be used for inverse estimation of  $x_0$  given an observed response  $y_0$ . More recently, the package has been updated to also handle objects of class `"lme"` from the `nlme` package; `nlme` is a recommended R package for fitting linear and nonlinear mixed-effects models and is installed with R. Current functionality includes both the Wald-based and inversion methods discussed previously. The main arguments for this function (as it applies to `"lme"` objects) are noted in Table 2 below. The code for the package is hosted on GitHub at <https://github.com/bgreenwell/investr>, but the latest stable release can be found on CRAN at <https://CRAN.R-project.org/package=investr>. An in-depth introduction to the `investr` package is given in Greenwell and Kabban (2014). Though the paper only covers cases with independent observations, the discussion is still relevant to the use of the package with LMMs.

Argument	Description
<code>object</code>	An object that inherits from class <code>"lm"</code> , <code>"glm"</code> , <code>"nls"</code> , or <code>"lme"</code> .
<code>y0</code>	The value of the observed response(s) or specified value of the mean response. For <code>"glm"</code> objects, <code>y0</code> should be on the scale of the response variable
<code>interval</code>	The type of interval required.
<code>level</code>	A numeric scalar between 0 and 1 giving the confidence level for the interval to be calculated.
<code>mean.response</code>	Logical indicating whether confidence intervals should correspond to an individual response ( <code>FALSE</code> ) or a mean response ( <code>TRUE</code> ). For <code>glm</code> objects, this is always <code>TRUE</code> .
<code>lower</code>	The lower endpoint of the interval to be searched.

Argument	Description
<code>upper</code>	The upper endpoint of the interval to be searched.
<code>tol</code>	The desired accuracy passed on to <code>uniroot</code> . Recommend a minimum of <code>1e-10</code> .
<code>maxiter</code>	The maximum number of iterations passed on to <code>uniroot</code> .
<code>q1</code>	Optional lower cutoff to be used in forming confidence intervals. Only used when object inherits from class "lme". Defaults to <code>stats::qnorm((1+level)/2)</code> .
<code>q2</code>	Optional upper cutoff to be used in forming confidence intervals. Only used when object inherits from class "lme". Defaults to <code>stats::qnorm((1-level)/2)</code> .

Returning to the bladder volume example, suppose we obtained an ultrasound measurement from a new patient for which  $HD^{3/2} = 500$  (that's roughly 63 on the original scale). What is the true volume of fluid ( $x_0$ ) in the patients bladder? We can estimate the true volume and form an approximate 95% confidence interval using the methods discussed previously. The point estimate is simply given by

$$\hat{x}_0 = \frac{500 + 53.83164}{69.09491} = 8.0155 \text{ (cl).}$$

This estimate can be obtained in R as follows:

```
library(investr) # install.packages("investr")
(x0.est <- invest(bladder.lme, y0 = 500, interval = "none"))

## volume
## 8.02
```

The code used by `invest` to obtain this point estimate is basically

```
fun <- function(x) {
  predict(bladder.lme, newdata = list("volume" = x), level = 0) - 500
}
uniroot(fun, lower = 1, upper = 17.5, tol = 1e-10, maxiter = 1000)$root

## [1] 8.02
```

In other words, `invest` relies on R function `uniroot` from the `stats` package to solve the equation  $f(x; \hat{\beta}) - y_0 = 0$  numerically for  $x$ . If the solution does not lie in the range of predictor values, then an error message will be displayed, as in

```
invest(bladder.lme, y0 = 1500)
```

```
## Error: Point estimate not found in the search interval (1, 17.5). Try tweaking the values of lower and upper.
```

The values for `lower`, `upper`, `tol`, and `maxiter` are controlled via the arguments of the same name listed in Table 2.

When `interval = "Wald"`, an asymptotic  $100(1 - \alpha)\%$  confidence interval (where  $\alpha$  is equal to `1 - level`) for  $x_0$  is calculated according to Equation (#):

```
invest(bladder.lme, y0 = 500, interval = "Wald")
```

```
## estimate    lower    upper    se
## 8.02       4.18    11.85    1.95
```

The standard error is computed using a First-order Taylor series approximation. Similar to the code snippet shown below, `invest` calls the `stats` function `numericDeriv` to numerically evaluate the gradient of  $\hat{x}_0$  as a function of  $y_0$  and  $\hat{\beta}$ .

```
x0Fun <- function(params) { # x0 as function of y0 and fixed effects
  fun <- function(x) {
```

```

X <- model.matrix(eval(bladder.lme$call$fixed)[-2],
                  data = data.frame("volume" = x))
X %*% params[-length(params)] - params[length(params)]
}
uniroot(fun, lower = 1, upper = 17.5, tol = 1e-10,
        maxiter = 1000)$root
}
params <- c(fixef(bladder.lme), 500)
covmat <- diag(3) # set up variance-covariance matrix
covmat[1:2, 1:2] <- vcov(bladder.lme) # fixed effects var/cov matrix
covmat[3, 3] <- 17572.35 # VAR[Y_0]
gv <- attr(numericDeriv(quote(x0Fun(params)), "params"), "gradient")
(se <- as.numeric(sqrt(gv %*% covmat %*% t(gv))))

```

```
## [1] 1.95
```

Alternatively, one can use the very useful `deltaMethod` function from the `car` package (Fox and Weisberg 2011) to obtain `se`:

```

library(car) # install.packages("car")
params <- c(fixef(bladder.lme), 500)
covmat <- diag(3) # set up var/cov matrix
covmat[1:2, 1:2] <- vcov(bladder.lme) # fixed effects var/cov matrix
covmat[3, 3] <- 17572.35 # VAR[Y_0]
names(params) <- c("b0", "b1", "y0")
(se <- deltaMethod(params, g = "(y0 - b0)/b1", vcov. = covmat)$SE)

```

```
## [1] 1.95
```

The only drawback here is that `deltaMethod` relies on the `stats` package symbolic differentiation function `D`; hence,  $\hat{x}_0 = f^{-1}(y_0; \hat{\beta})$  has to be obtainable in closed-form.

To obtain the approximate inversion interval (??), we specify `interval = "inversion"` (the default) as in the following:

```
invest(bladder.lme, y0 = 500, interval = "inversion")
```

```
## estimate    lower    upper
##      8.02     4.23     11.92
```

Notice there is no standard error estimate when computing the inversion interval. Essentially, `invest` finds the lower and upper inversion confidence limits (#) by solving the equations

$$Q_I - z_{\alpha/2} = 0 \quad \text{and} \quad Q_I - z_{1-\alpha/2} = 0$$

numerically for  $x$  using the R function `uniroot`. To use the quantiles from a  $t$ -distribution instead (as suggested in Oman (1998)), we can supply them via the arguments `q1` and `q2`:

```

N <- nrow(bladder) # total sample size
tvals <- qt(c(0.025, 0.975), df = N-1) # quantiles from t dist with N-1 d.f.
invest(bladder.lme, y0 = 500, q1 = tvals[1], q2 = tvals[2])

```

```
## estimate    lower    upper
##      8.02     4.20     11.95
```

As expected, this leads to a slightly larger inversion interval for  $x_0$ . Being able to specify specific quantiles via the arguments `q1` and `q2` will also be useful when implementing the bootstrap adjusted inversion interval described in Section (#).

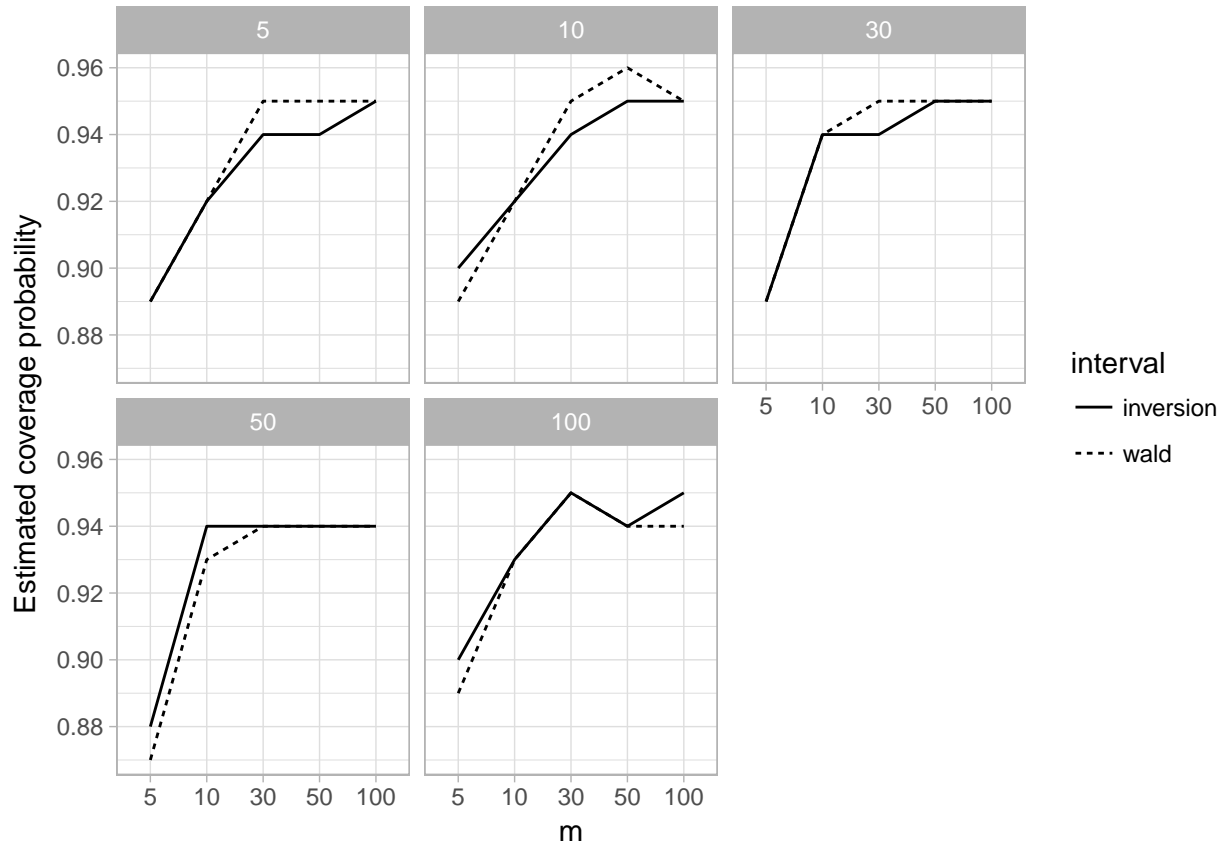


## Monte carlo study

To assess the empirical performance of these confidence intervals, we carried out a small Monte Carlo study (The simulation described in this section was conducted in R using packages `plyr` (Wickham 2011), `nlme`, and `lme4`; the source code can be made available upon request to the authors). The results are reported in Table # and indicate that the Wald-based confidence interval (?) and inversion confidence interval (?) have asymptotic coverage probability close to  $100(1 - \alpha)\%$ . The main point of this experiment is to highlight the fact that it is the number of subjects  $m$ , not the sample size per subject  $n$ , that has the biggest impact on the asymptotic coverage probability for the Wald-based and inversion intervals discussed previously.

We consider the values 5, 10, 30, 50, and 100 for both the number of subjects  $m$  and the number of observations per subject  $n$ . For each combination of sample sizes, we generated 1,000 data sets from the random intercept and slope model fit to the bladdr volume data; that is, a model with fixed effects given by those listed in Table 1. The standard deviations for the (uncorrelated) random intercept and slope were 39.62499, and 14.28841, respectively. The residual standard deviation was  $\sigma = 53.71511$ . We chose  $f(x_0; \beta) = 500$  so that the true unknown is  $x_0 = 8.0155$ . The standard deviation of the coverage estimates is approximately  $\sqrt{0.95(1 - 0.95)/1000} = 0.001$ . A trellis plot of the results is given in Figure #. The coverage estimates are plotted against the number of subjects  $m$  and paneled by number of observations per subject  $n$ . The results indicate that  $m \geq 30$  with  $n \geq 5$  is sufficient for achieving close to the stated  $1 - \alpha$  coverage probability in this particular example.

$m$	Method	$n = 5$	$n = 10$	$n = 30$	$n = 50$	$n = 100$
5	Wald	0.89	0.89	0.89	0.87	0.89
	Inversion	0.89	0.90	0.89	0.88	0.90
10	Wald	0.92	0.92	0.94	0.93	0.93
	Inversion	0.92	0.92	0.94	0.94	0.93
30	Wald	0.95	0.95	0.95	0.94	0.95
	Inversion	0.94	0.94	0.94	0.94	0.95
50	Wald	0.95	0.96	0.95	0.94	0.94
	Inversion	0.94	0.95	0.95	0.94	0.94
100	Wald	0.95	0.95	0.95	0.94	0.94
	Inversion	0.95	0.95	0.95	0.94	0.95



### The parametric bootstrap

Implementation of the parametric bootstrap algorithm in Figure (#) is relatively straight forward using the new `bootMer` function from the well-known R package `lme4` (D. Bates et al. 2014) in conjunction with the `boot` package.

Since we will be using the `lme4` package, we need to refit the model using the `lmer` function (notice the different syntax required for specifying the same random effects structure):

```
library(lme4) # install.packages("lme4")
bladder.lmer <- lmer(HD^(3/2) ~ volume + (0+1|subject) + (0+volume|subject),
  data = bladder)
```

Theoretically, the parameter estimates from this model should be the same as those from `bladder.lme`; however, there are likely to be small numerical differences between the two. For this reason, let us re-estimate  $\hat{x}_0$  using `bladder.lmer`. Since `invest` does not work on objects fit using `lmer` from the `lme4` package (i.e., object of class "lmerMod"), we have to do things manually:

```
fe <- unname(fixef(bladder.lmer)) # fixed effects without dimnames attribute
(x0.est <- (500 - fe[1]) / fe[2])
```

```
## [1] 8.02
```

Also, for convenience, we define the following function which estimates  $VAR[\mathcal{Y}|x] = \sigma_0^2 + x^2\sigma_1^2 + \sigma^2$  for a given value of  $x$ :

```
var.y <- function(object, x) {
  vc <- as.data.frame(lme4::VarCorr(object))$vcov
```

```
vc[1] + vc[2]*x^2 + vc[3]
}
```

For example, to estimate  $\sigma_0^2 = \widehat{VAR}[\mathcal{Y}_0]$ , we have `var.y(bladder.lmer, x = x0.est)`, which gives  $1.757 \times 10^4$ , the same value used in the previous section.

Although we could easily compute all the bootstrap intervals previously discussed in one call to `bootMer` and `boot.ci`, we will discuss and compute each interval separately.

The following snippet of code generates  $R = 9999$  bootstrap replicates of  $\hat{x}_0$ ,  $Q_W$ , and  $Q_I$  according to the algorithm in Figure (#):

```
boot.fun <- function(.) { # bootstrap function

  # Point estimate
  var.y0.boot <- var.y(., x = x0.est) # VAR[Y0]
  fe.boot <- unname(fixef(.)) # fixed effects
  if (all(getME(., "y") == bladder$HD^(3/2))) {
    y0.boot <- 500
  } else {
    y0.boot <- rnorm(1, 500, sqrt(var.y0.boot))
  }
  x0.boot <- (y0.boot - fe.boot[1])/fe.boot[2]

  # Approximate variance
  covmat <- diag(3)
  covmat[1:2, 1:2] <- as.matrix(vcov(.))
  covmat[3, 3] <- var.y0.boot
  params <- c("b0" = fe.boot[1], "b1" = fe.boot[2], "y0" = y0.boot)
  dm <- deltaMethod(params, g = "(y0 - b0)/b1", vcov. = covmat)
  var.x0.boot <- dm$SE^2

  # Approximate predictive pivot
  mu0.boot <- as.numeric(crossprod(fe.boot, c(1, x0.est)))
  var.mu0.boot <- t(c(1, x0.est)) %*% as.matrix(vcov(.)) %*% c(1, x0.est)
  QI.boot <- (y0.boot - mu0.boot)/sqrt(var.y0.boot + var.mu0.boot)

  # Return vector of results
  c(x0.boot, var.x0.boot, QI.boot)
}

pb <- bootMer(bladder.lmer, boot.fun, nsim = 9999, seed = 105) # run simulation
```

The `bootMer` function returns an object of class `boot` which can then be processed via the `boot` package to obtain the various bootstrap confidence intervals discussed earlier. A basic summary of `pb` is given by

```
library(boot) # load boot package
summary(pb)
```

```
##      R original bootBias bootSE bootMed
## 1 9999      8.02 -0.00219   1.97  8.0186
## 2 9999      3.82  0.05243   1.02  3.7702
## 3 9999      0.00 -0.00810   1.00  0.0016
```

The estimated standard error and bias of  $\hat{x}_0$ , based on  $R = 9999$  bootstrap replicates, are 1.974 and 0.052, respectively. The original estimate  $\hat{x}_0$  and the median of the bootstrap replicates are also given in the first row of the summary. A graphical summary of the bootstrap simulation is given in Figure (#). These graphs

indicate that the sampling distributions of  $\hat{x}_0$ ,  $Q_W$ , and  $Q_I$  are all approximately normal; hence, we would expect the bootstrap confidence intervals to be similar to the asymptotic methods based on the normal distribution discussed in Sections (#)-(#).

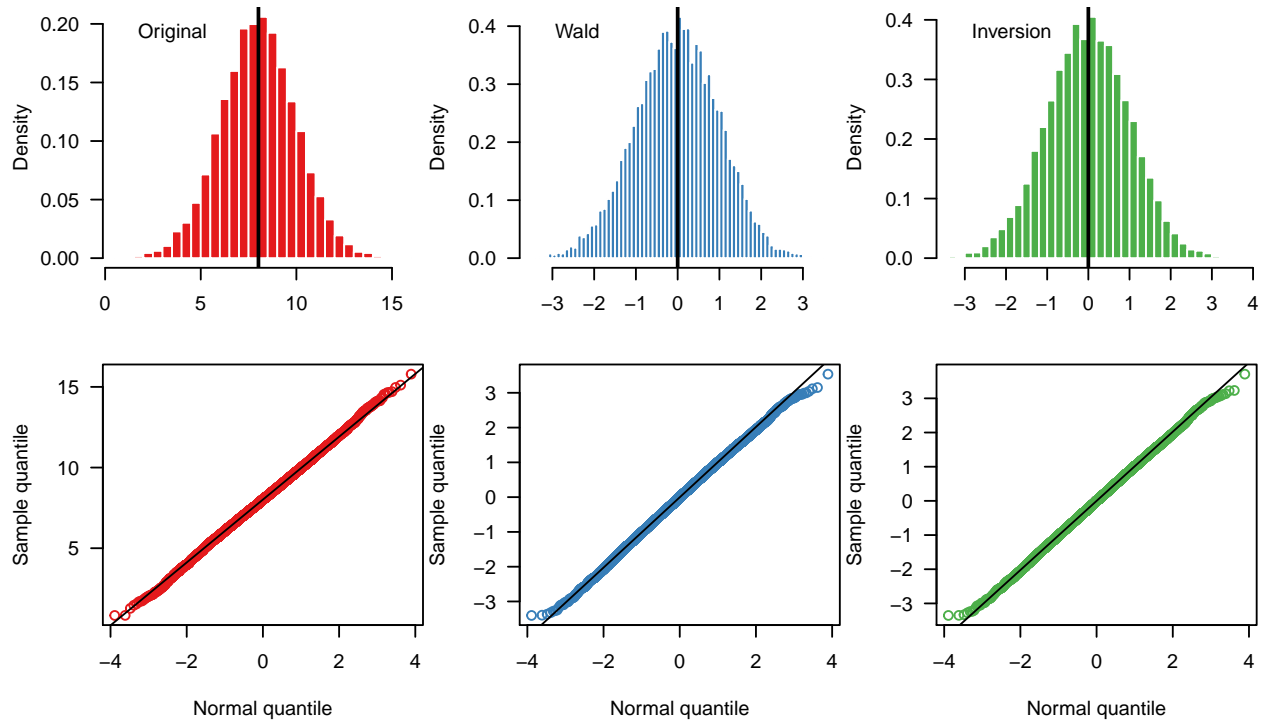


Figure 2: Graphical summary of bootstrap replicates.  $R = 9,999$  bootstrap replicates of  $\hat{x}_0$  (left),  $Q_W$  (middle), and  $Q_I$  (right).

To obtain the percentile and studentized  $t$  intervals (see Sections (#)-(#)), we can use the `boot` package function `boot.ci`:

```
boot.ci(pb, type = c("norm", "perc", "stud"), index = 1:2)

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = pb, type = c("norm", "perc", "stud"), index = 1:2)
##
## Intervals :
## Level      Normal      Studentized      Percentile
## 95%    ( 4.15, 11.89 )  ( 4.30, 11.99 )  ( 4.05, 11.91 )
## Calculations and Intervals on Original Scale
```

For comparison, we also included the option to compute a bootstrap normal-approximation confidence interval. This interval has the form

$$(\hat{x}_0 - \text{bootBias}) \pm z_{\alpha/2} \text{bootSE}$$

where `bootBias` and `bootSE` can be found in the first row of `summary(pb)`. In other words, it is just a Wald-type interval that uses a bias-corrected estimate of  $x_0$ , along with a bootstrap estimate of the standard error of  $\hat{x}_0$ . While this may be more accurate than the ordinary Wald-based interval (#), it may still not perform well in small sample sizes because of the strict normality assumption. In this example, however, normality does not appear to be an issue.

The bootstrap adjusted inversion interval can be computed as easily as the ordinary inversion interval, except we need to supply `invest` with the estimated quantiles  $\hat{F}_{Q_I}(0.025)$  and  $\hat{F}_{Q_I}(0.975)$ :

```
QI.boot <- pb$t[, 3] # bootstrap replicates of Q_I
qvals <- quantile(QI.boot, c(0.025, 0.975)) # sample quantiles
invest(bladder.lme, y0 = 500, q1 = qvals[1], q2 = qvals[2])
```

```
## estimate    lower    upper
##      8.02     4.28     12.02
```

All of the approximate 95% confidence intervals we computed for the true volume of fluid are summarized in Table (3) below. Notice that all of the bootstrap-based confidence intervals for  $x_0$  (indicated by a  $\star$ ) are slightly wider than those based on large sample normal theory results. This is likely due to the fact that the large sample intervals do not take into account the variability of the estimated variance components.

Method	Estimate	SE	95% Bounds	Length
$CI_{wald}(x_0)$	8.016	1.954	(4.185, 11.846)	7.66
$CI_{inv}(x_0)$	8.016	—	(4.228, 11.919)	7.691
$CI_{percentile}^{\star}(x_0)$	8.016	1.974	(4.05, 11.913)	7.863
$CI_{wald}^{\star}(x_0)$	8.016	1.974	(4.252, 11.947)	7.695
$CI_{inv}^{\star}(x_0)$	8.016	1.974	(4.278, 12.019)	7.741

Table 3. Summary of results for the bladder volume example. A  $\star$  symbol indicates a parametric bootstrap-based confidence interval

For the bladder volume data, we were able to transform the response so that a straight line provided a reasonable fit. This simple form led to closed-form solution for  $\hat{x}_0$ . This is not always the case in practice. For example, suppose that we observed a new ultra sound measurement  $HD = 90$  and we wish to estimate the true volume of liquid in the patients bladder using the original (untransformed) data. A spaghetti plot of the untransformed data is displayed in the left side of Figure (#). The right side of Figure (#) displays the estimated coefficients from individual linear regression models with a quadratic term fit to each subject (as before, the predictor  $V$  was centered); the bars represent one-at-a-time 95% confidence intervals

As it turns out, adding a quadratic term to the previously fitted LMM yields a reasonable fit. Since the intervals for coefficient of the quadratic term in Figure (#) mostly overlap, we used the same random effects structure as before with the model for the transformed data.

$$HD_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})V_{ij} + \beta_2 V_{ij}^2 + \epsilon_{ij}$$

$$b_{ki} \sim \mathcal{N}(0, \theta_k^2), k = 0, 1,$$

$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

As before, we assume that the random effects are uncorrelated. Table (#) displays the fixed-effects results from applying this model to the bladder volume data.

Table 5: Fixed-effects t-table for the random intercept and slope model fit to the bladder volume data.

	Estimate	Std. Error	t value
(Intercept)	1.887	1.714	1.1
volume	8.656	0.318	27.2
I(volume^2)	-0.146	0.014	-10.1

The point estimate of  $x_0$  is easily obtained using the quadratic formula:  $\hat{x}_0 = 13.05$ . Recall, from Figure

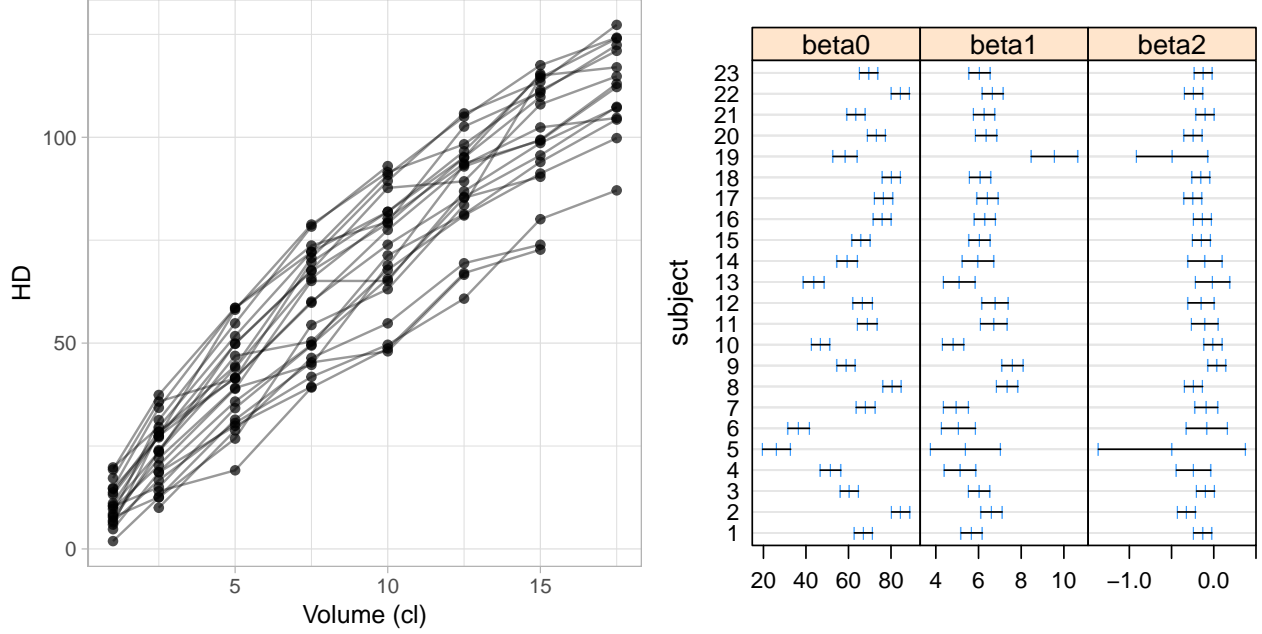


Figure 3: Bladder volume data. Left: spaghettiplot of untransformed data (lines connect measurements belonging to the same subject). Right: Estimated subject specific intercepts and slopes (with one-at-a-time 95% confidence limits).

(#) that each patient has a slightly nonlinear trajectory; thus, there is no reason to expect the sampling distribution of  $\hat{x}_0$  to be normal, or even symmetric in this case. To see that this is indeed the case, we applied the parametric bootstrap. The results are summarized in Figure (#). Clearly, the Wald-based confidence interval (#) will not be accurate in this case. However, the approximate predictive pivot used in the inversion interval (#) appears reasonably normal. Thus, our recommendation is that, if the number of subjects  $m$  is reasonably large (say  $m \geq 30$ ) and the bootstrap replicates are approximately normal (see Figure (#)), then the Wald-based and inversion methods are useful. Otherwise, it is probably best to stick with the parametric bootstrap confidence intervals or, if prior information is available, adopt a fully Bayesian approach.

## Conclusion

We have discussed a number of confidence interval procedures for statistical calibration in linear models with random coefficients with a single level of grouping. We have described two R packages for implementing these procedures: **investr** and **lme4**. The **investr** package can be used for obtaining the asymptotic confidence intervals (i.e., the Wald-based and inversion confidence intervals). We also showed how the **lme4** package can be used to obtain calibration intervals based on a parametric bootstrap using the recently added **bootMer** function. Future work will likely extend the methods discussed in this paper to more complicated cases such as nonlinear mixed-effects models and multi-level hierarchical models (i.e., more than one grouping variable).

## References

- Bates, Douglas, Martin Maechler, Ben Bolker, and Steven Walker. 2014. *Lme4: Linear Mixed-Effects Models Using Eigen and S4*. <http://CRAN.R-project.org/package=lme4>.
- Boos, D.D., and L.A. Stefanski. 2013. *Essential Statistical Inference: Theory and Methods*. Springer Texts in

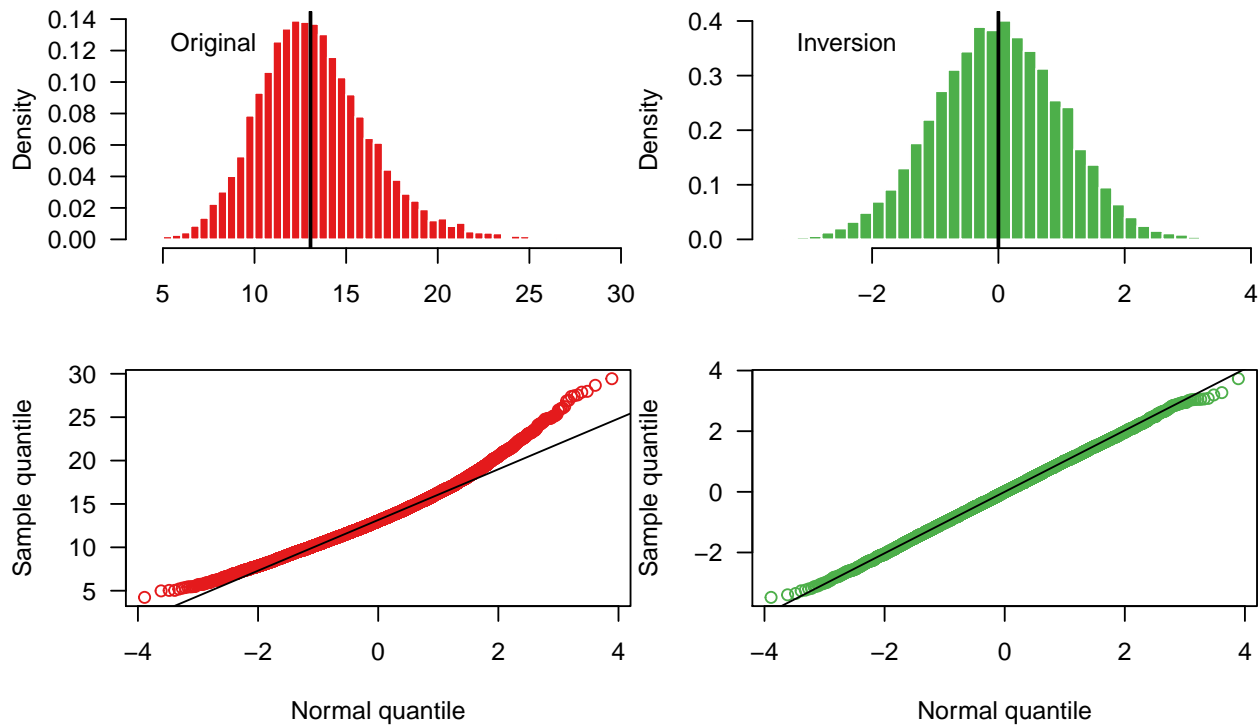


Figure 4: Graphical summary of bootstrap replicates.  $R = 9,999$  bootstrap replicates of  $\hat{x}_0$  (left),  $Q_W$  (middle), and  $Q_I$  (right).

Statistics. Springer London, Limited.

Brown, P.J. 1993. *Measurement, Regression, and Calibration*. Oxford Science Publications. Clarendon Press.

Davison, A. C., and D. V. Hinkley. 1997. *Bootstrap Methods and Their Applications*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press.

Demidenko, E. 2013. *Mixed Models: Theory and Applications with R*. Wiley.

Efron, B. 1979. “Bootstrap Methods: Another Look at the Jackknife.” *The Annals of Statistics* 7 (1): 1–26.

Fox, John, and Sanford Weisberg. 2011. *An R Companion to Applied Regression*. Second. Thousand Oaks CA: Sage. <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>.

Graybill, F. A. 1976. *Theory and Application of the Linear Model*. Duxbury Classic Series. Pacific Grove, CA: Duxbury.

Greenwell, Brandon M. 2013. *Investr: Functions for Inverse Estimation with Linear and Nonlinear Regression Models in R*. <http://CRAN.R-project.org/package=investr>.

———. 2014. “Topics in Statistical Calibration.” PhD thesis, Air Force Institute of Technology.

Greenwell, Brandon M., and Christine M. Schubert Kabban. 2014. “Investr: An R Package for Inverse Estimation.” *The R Journal* 6 (1): ??–??

Hall, Peter. 1986. “On the Bootstrap and Confidence Intervals.” *The Annals of Statistics* 14 (4): 1431–52.

Hoadley, Bruce. 1970. “A Bayesian Look at Inverse Linear Regression.” *Journal of the American Statistical Association* 65 (329): 356–69.

Huet, S. 2004. *Statistical Tools for Nonlinear Regression: A Practical Guide with S-Plus and R Examples*.

Springer Series in Statistics. Springer.

Jones, Geoffrey, and David M. Roche. 1999. "Bootstrapping in Controlled Calibration Experiments." *Technometrics* 41 (3): 224–33.

Laird, Nan M., and James H. Ware. 1982. "Random-Effects Models for Longitudinal Data." *Biometrics* 38 (4): 963–74.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. Wiley.

Oman, Samuel D. 1998. "Calibration with Random Slopes." *Biometrika* 85 (2): 439–49.

Osborne, C. 1991. "Calibration: A Review." *International Statistical Review* 59 (3): 309–36.

Pinheiro, J.C., and D.M. Bates. 2000. *Mixed-Effects Models in S and S-Plus*. Statistics and Computing. Springer.

Pinheiro, Jose, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Core Team. 2013. *Nlme: Linear and Nonlinear Mixed Effects Models*.

Seber, G.A.F., and C.J. Wild. 2003. *Nonlinear Regression*. Wiley Series in Probability and Statistics. Wiley.

Wickham, Hadley. 2011. "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software* 40 (1): 1–29. <http://www.jstatsoft.org/v40/i01/>.