Inverse Estimation with Linear Mixed-Effects Models

with Application in R

Brandon M. Greenwell

Christine M. Schubert Kabban

Andrew J. McCarthy

Bradley C. Boehmke

Inverse estimation, also known as inverse prediction or statistical 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 both asymptotic methods and a parametric bootstrap. The converage probability for the asymptotic procedures is assessed using a small Monte-Carlo simulation. Examples are given using the R programming language with a real data set.

#### Keywords: calibration, regulation, inverse prediction, LMM, parametric bootstrap

## Introduction

Consider an ordinary regression model , where is a known expectation function (called a *calibration curve* in this context) that is monotonic over the range of interest and . A common problem in regression is to predict a future response , or estimate the mean response , for a known value of the explanatory variable . Often, however, there is a need to do the reverse; that is, given an observed value of the response (*calibration*), or a specified value of the mean response (*regulation*), infer the unknown value of the explanatory variable . We refer to both situations (i.e., calibration and regulation) more generally as inverse estimation. Though this paper focuses on calibration (in particular, *controlled calibration*, where the predictor is held fixed by design), the methods can be narrowed to handle regulation.

A thorough overview of the calibration problem is given in Osborne (1991) and Greenwell (2014). Oman (1998) considers the case of a random intercept and slope model and provides an approximate parametric bootstrap algorithm for inferring (the algorithm we present later is fully parametric).

This paper concerns inverse estimation with linear mixed-effects models (LMMs); however, the methods presented here can be extended to generalized least-squares (GLS), nonlinear mixed-effects models (NLMMS) and generalized linear mixed-effects models as well. In particular, we extend the application of calibration to *grouped data*; that is, data in which the observations are grouped into disjoint classes called clusters or groups. Common examples of grouped data include *repeated measures data* and *longitudinal data*. Groups tend to be homogeneous, therefore, observations belonging to the same group cannot be considered independent. (Although, observations between clusters usually are.) Thus, we need to account for within cluster dependence when modeling this type of data. To our knowledge, other than Oman (1998), very little has been done for calibration with grouped data. Oman considered a simpler model that only allowed for the intercept and slope to vary between groups, whereas we take a more general (and practical) approach that allows for an arbitrary random effects structure. Furthermore, while Oman considers only one type of calibration interval, we discuss four different calibration intervals that can be computed for grouped data, along with some adjustments to improve their accuracy. Moreover, the calibration interval considered by Oman was based on an approximate parametric bootstrap that did not account for the variance attributed by the random variable .

### Linear mixed-effects models

In practice, multiple observations are often taken from the same subject or experimental unit. This type of data is called *repeated measures* data. This includes, for example, *longitudinal* data or *panel* data (where experimental units are observed over time). One feature to remember about repeated measures is that the individual observations are not independent. This feature must be taken into account in order to obtain valid standard errors, 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

where

* is an response vector for the -th subject/cluster/group;
* is an design matrix for the fixed effects;
* is an design matrix for the random effects;
* is a vector of fixed effects coefficients;
* is a vector of random effects coefficients with mean zero and variance-covariance matrix ;
* is a variance-covariance matrix for the random effects;
* is an vector of random errors with mean zero and variance-covariance matrix .

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

where , , , , and for . Since , it is assumed that the random vectors are mutually independent.

The additional term in the model imposes a specific variance-covariance structure on the response vector :

Thus, the fixed effects determine the mean of , while the random effects govern the variance-covariance structure of . Different random effects structures impose different variance-covariance structures on the response, resulting in a highly flexible framework for modeling repeated measures.

The random effects variance-covariance matrix has at most unique elements which we represent by the vector . There are a number of methods available for estimating ; see, for example, McCulloch, Searle, and Neuhaus (2008), chap. 6, and Demidenko (2013), chap. 2. Most commonly, the fixed effects are estimated via the method of maximum likelihood (ML), while the variance components are estimated via restricted maximum likelihood (REML). The ML estimator of , given by

depends on the estimated variance components through which makes it difficult to capture the variability of 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 as the true (fixed) value of . 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 rewrite the LMM (1) as

where and are linear in and , respectively. For instance, the random intercept and slope model has and with and .

Assume that, after the data are collected and a model is fitted, we obtain a new observation, denoted , from the same population under study for which the value of the explanatory variable is unknown. We assume that the new observation belongs to a group not included in our analysis. Estimating is rather straightforward. By assumption, the new observation is distributed as a random variable with . A natural estimator for is obtained by inverting to obtain

where is the ML estimator of . We shall refer to Equation (2) as the inverse estimator. Note that the point estimate 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 are given in Greenwell (2014), Section 5.2.

Another approach is to maximize the profiled likelihood for . This can be done simply by updating the original fitted model after adding a new point to the original data. Doing this for a range of values for will give the profile likelihood . The maximum likelihood estimate of is then

For a balanced random intercept model, this coincides with the inverse estimator (2).

## Interval estimation

Before introducing our parametric bootstrap algorithm, we first discuss two asymptotic approaches to interval estimation: (1) using a Taylor series approximation to the standard error of to formulate a Wald-based interval, and (2) inverting an asymptotic prediction interval for . Both methods have the benefit of being computationally fast; but, as we will see in the next two sections, these methods do not take into account the variance of the estimated variance components, leading to confidence intervals for that are too narrow.

### Wald interval

There is no "textbook" formula for the standard error of ---not even in the case of the simple linear regression model with independent and identically distributed (i.i.d.) normal errors. 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 of the standard error, denoted , relies on the variance-covariance matrix of ; namely,

Since is independent of , it is also independent of , hence the diagonal structure of . Recall that our point estimate has the form . Let and denote the partial derivatives of with respect to the parameters and , respectively. A first-order Taylor-series approximation for the variance of is given by

To obtain , we simply replace and with their respective estimates and . Assuming large sample normality for leads to an approximate Wald confidence interval for of

where is the probit function.

Alternatively, one can use a parametric bootstrap estimate of the standard error, say ; this is discussed in a later section. The advantage of using over is that , through simulation, would take into account the variability of the estimated variance components and .

### Inversion interval

In the case of the simple linear regression model with i.i.d. normal errors, an exact confidence interval for can be derived (see, for example, Graybill 1976). This can be generalized to an approximate method in the case of polynomial and nonlinear regression models with i.i.d. normal errors; see, for example, Seber and Wild (2003) and Huet (2004). In a similar fashion, we can generalize the same results to an approximate method for LMMs.

Let be the predicted mean response at . A prediction interval for at with asymptotic coverage probability is

If instead, is observed to be and is unknown, then an asymptotic confidence interval for the unknown can be obtained by inverting :

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. Furthermore, note that is not symmetric about and will not necessarily result in a single finite interval; see, for example, Greenwell and Kabban (2014).

Finally, notice that relies on the quantiles from a normal distribution. While it is likely that a -distribution may be more accurate, it is difficult to determine the appropriate degrees of freedom. Oman (1998) suggests a -distribution with degrees of freedom ( being the total sample size).

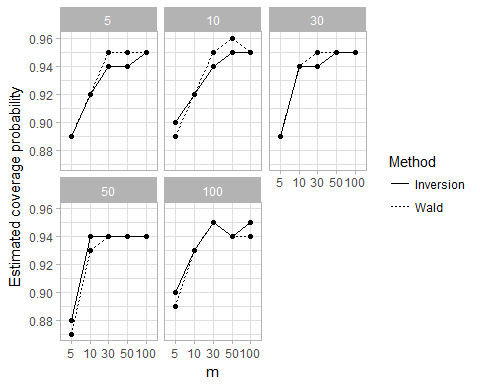
### Monte Carlo study

To assess the empirical performance of and , 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 1 and indicate that both and have asymptotic coverage probability close to with sufficient sample size. The main point of this experiment was to highlight the fact that it is the number of subjects , not the sample size per subject , that has the biggest impact on the asymptotic coverage probability.

We considered the values 5, 10, 30, 50, and 100 for both the number of subjects and and the number of observations per subject . For each combination of sample sizes, we generated 1,000 data sets from a random intercept and slope model with parameters with fixed-effects and . The standard deviations for the (uncorrelated) random intercept and slope were 39.62499, and 14.28841, respectively. The residual standard deviation was . We chose so that the true unknown is . The standard deviation of the coverage estimates is approximately . A trellis plot of the results is given in Figure 1. The coverage estimates are plotted against the number of subjects and paneled by number of observations per subject . The results indicate that with is sufficient for achieving close to the stated coverage probability in this particular example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Method |  |  |  |  |  |
| 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 |

**Table 1.** Estimated coverage probability for and as a function of the number of subjects (rows) and the number of observations per subject (columns).



**Figure 1.** Estimated coverage probability for and as a function of the number of subjects (-axis) and the number of observations per subject (panels).

### Parametric bootstrap

The bootstrap (B. 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 dependent observations, the nonparametric bootstrap does not easily apply, and instead, we adopt a parametric approach. In a parametric bootstrap, new samples are generated from a fitted parametric model, rather than sampling with replacement directly from the data. Fortunately, the 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 be an estimate of the variance of the new observation . An algorithm for bootstrapping in an LMM is given below. Note that step (3) is crucial for calibration problems because we need to treat as a random quantity in the bootstrap simulation, otherwise the variability of will be underestimated; see, for example, Jones and Rocke (1999) and Greenwell and Kabban (2014).

* Fit an LMM---Equation (1)---to the data and obtain estimates , , and .
  1. Define , where and ;
  2. Update the original model using as the response vector to obtain and ;
  3. Generate ;
  4. Define ;
* Repeat steps (1)-(4) times.

**Algorithm 1.** Parametric bootstrap for controlled calibration in LMMs.

There are many bootstrap confidence interval procedures available once a set of bootstrap replicates has been obtained, for instance: the percentile method (B. Efron 1979), the studentized bootstrap method (Bradley Efron 1982), and the double bootstrap method (Hall 1986). For a good overview of these confidence interval procedures and more, see Davison and Hinkley (1997), chap. 5, and Boos and Stefanski (2013), chap. 11. In the next section, we discuss how to use Algorithm 1 to adjust the previously discussed inversion interval.

It is possible to adopt a Bayesian approach to obtain the posterior distribution of . However, as discussed in Hoadley (1970) (and the commenting articles), finding a prior for 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

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 to account for the variability of the estimated variance components. The inversion interval assumes that the *predictive pivot*

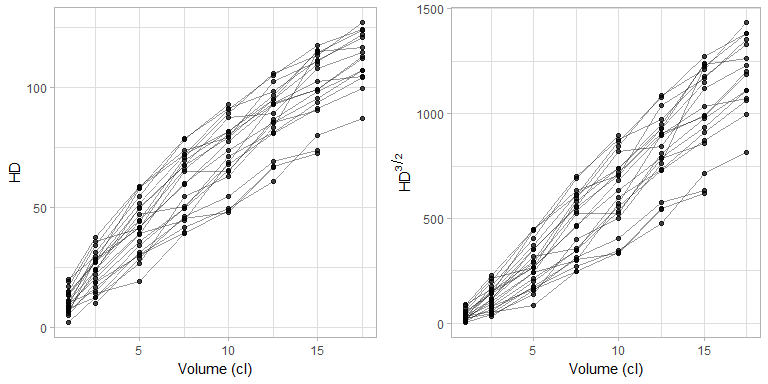
has a standard normal distribution. A bootstrap modified inversion interval would instead use the bootstrap replicates defined by

to estimate the sampling distribution of . If is the empirical distribution function for a sample of bootstrap replicates , then the bootstrap modified inversion interval for is given by

## Bladder volume example

For illustration, we consider the bladder volume data which first appeared in Haylen et al. (1989) and again in Brown (1993), pg. 7, and Oman (1998). The study sample consisted of a series of 23 female patients attending a urodynamic clinic. After successfully voiding their bladder, each subject was injected with sterile water in additions of 1, 1.5, and then 2.5 cL increments up to a final cumulative total of 17.5 cL. At each true volume a measure of height () in mm and depth () in mm of largest ultrasound bladder images were taken. The product was taken as a measure of liquid volume.

A spaghetti plot of the raw data is displayed in the left side of Figure 2. As noted by Brown (1993), should be linearly related to . The transformed data are displayed in the right side of Figure 2.



**Figure 2.** Spaghetti plots of the bladder volume data (lines connect measurements belonging to the same subject). Left: Original data. Right: Transformed data.

As indicated by Figure 2, the intercept and slopes seem to very between subjects for the transformed data; hence, the following random intercept and slope model seems appropriate:

Here we also assume that the random effects are uncorrelated (i.e, ). Table 2 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)))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**Table 2.** Fixed effects *t*-table for the random intercept and slope model fit to the bladder volume data.

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 R package investr

The R package investr (Greenwell and Kabban 2014) facilitates calibration/inverse estimation with linear and nonlinear regression models. The main function, invest, can be used for inverse estimation of given an observed response . 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 3 below. The source code for the package is hosted on GitHub at <https://github.com/bgreenwell/investr>, and the latest stable release can be found on CRAN at <https://CRAN.R-project.org/package=investr>.

Fortunately, can be computed numerically 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 a later section.

|  |  |
| --- | --- |
| Argument | Description |
| object | An object that inherits from class "lm", "glm", "nls", or "lme". |
| y0 | The value of the observed response(s) or specified value of the mean response. For "glm" objects, y0 should be on the scale of the response variable |
| interval | The type of interval required. |
| level | A numeric scalar between 0 and 1 giving the confidence level for the interval to be calculated. |
| mean.response | Logical indicating whether confidence intervals should correspond to an individual response (FALSE) or a mean response (TRUE). For "glm" objects, this is always TRUE. |
| lower | The lower endpoint of the interval to be searched. |
| upper | The upper endpoint of the interval to be searched. |
| tol | The desired accuracy passed on to uniroot. Recommend a minimum of 1e-10. |
| maxiter | The maximum number of iterations passed on to uniroot. |
| q1 | Optional lower cutoff to be used in forming confidence intervals. Only used when object inherits from class "lme". Defaults to stats::qnorm((1+level)/2). |
| q2 | Optional upper cutoff to be used in forming confidence intervals. Only used when object inherits from class "lme". Defaults to stats::qnorm((1-level)/2). |

**Table 3.** The main arguments for the invest function (as it applies to "lme" objects).

Returning to the bladder volume example, suppose we obtained an ultrasound measurement from a new patient for which (that's roughly 63 cL on the original scale). What is the true volume of fluid () 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

This estimate can be obtained in R as follows:

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

## [1] 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 the R function uniroot from the stats package to solve the equation numerically for . 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 confidence interval (where is equal to 1 - level) for is calculated according to Equation (2):

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 as a function of and .

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

If a closed-form formula is available for , then the deltaMethod function from the car package (Fox and Weisberg 2011) can also be used to obtain the Taylor series approximation for :

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, 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

numerically for using the R function uniroot. To use the quantiles from a -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 . Being able to specify specific quantiles via the arguments q1 and q2 will also be useful in constructing .

### The parametric bootstrap

Implementing Algorithm 1 is relatively straightforward using the new bootMer function from the well-known R package lme4 (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 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 for a given value of :

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 , 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 bootstrap replicates of , , and according to the Algorithm 1:

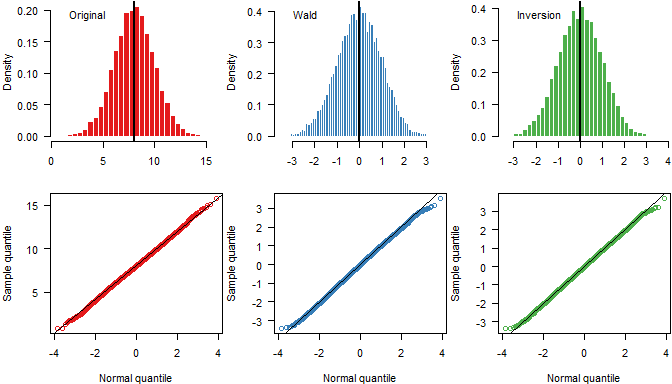
boot.fun <- function(.) { # bootstrap function  
  
 # Point estimate  
 var.y0.boot <- var.y(., x = x0.est) # VAR[Y0]  
 fe.boot <- unname(fixef(.)) # fixed effects  
 if (all(getME(., "y") == bladder$HD^(3/2))) {  
 y0.boot <- 500  
 } else {  
 y0.boot <- rnorm(1, 500, sqrt(var.y0.boot))  
 }  
 x0.boot <- (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 , based on 9999 bootstrap replicates, are 1.974 and 0.052, respectively. The original estimate 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 3. These graphs indicate that the sampling distributions of , , and 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 previously.



**Figure 3.** Graphical summary of bootstrap replicates. bootstrap replicates of (left), (middle), and (right).

To obtain the percentile and studentized intervals, 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

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 , along with a bootstrap estimate of the standard error of . While this may be more accurate than , it still may 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 and :

QI.boot <- pb$t[, 3] # bootsrap 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 4 below. Notice that all of the bootstrap-based confidence intervals for (indicated by a ) 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 |
|  | 8.016 | 1.954 | (4.185, 11.846) | 7.66 |
|  | 8.016 | --- | (4.228, 11.919) | 7.691 |
|  | 8.016 | 1.974 | (4.05, 11.913) | 7.863 |
|  | 8.016 | 1.974 | (4.252, 11.947) | 7.695 |
|  | 8.016 | 1.974 | (4.278, 12.019) | 7.741 |

**Table 4.** Summary of results for the bladder volume example. A 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 a closed-form solution for . 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.

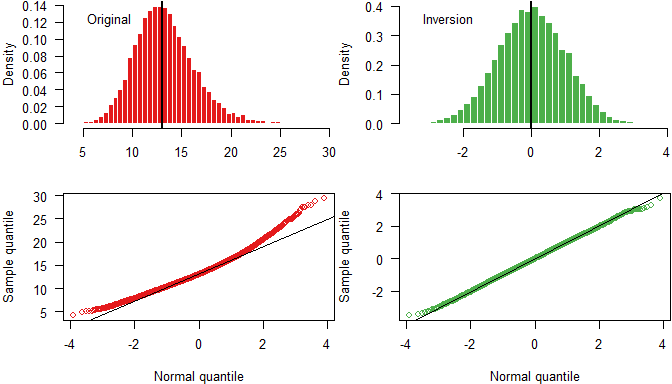
As it turns out, adding a quadratic term to the previously fitted LMM yields a reasonable fit. We used the same random effects structure as before with the model for the transformed data.

As before, we assume that the random effects are uncorrelated. Table 5 displays the fixed effects results from applying this model 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 |

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

The point estimate of is easily obtained using the quadratic formula: . Recall, from Figure 2 that each patient has a slightly nonlinear trajectory; thus, there is no reason to expect the sampling distribution of 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 4. 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 is reasonably large (say ) and the bootstrap replicates are approximately normal (see Figure 3), 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.



**Figure 4.** Graphical summary of bootstrap replicates. bootstrap replicates of (left) and (right).

## Discussion

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).

While most calibration experiments involve only a single predictor, it is not uncommon for there to be additional covariates (e.g., gender, age, race, etc.). For the latter case, all the methods discussed in this paper still apply as long as the additional covariates are held fixed at a constant value (e.g., the mean for continuous covariates and the most frequent level for factors).

## 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 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.

Efron, Bradley. 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society for Industrial; Applied Mathematics.

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. 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): 90–100. <http://journal.r-project.org/archive/2014-1/greenwell-kabban.pdf>.

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

Haylen, B. T., M. I. Frazer, J. R. Sutherst, and C. R. West. 1989. “Transvaginal Ultrasound in the Assessment of Bladder Volumes in Women: Preliminary Report.” *British Journal of Urology* 63 (2): 149–51.

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. Rocke. 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/>.