Inverse Estimation with Linear Mixed-Effects Models

with Application in R

Brandon M. Greenwell

January 15, 2017

Abstract

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 asymptotic methods and a parametric bootstrap. The converage probability for the asymptotic intervals is estimated as a function of sample size using a small Monte-Carlo simulation. Examples are given using the R programming language with a real data set.

Introduction

Consider an ordinary regression model $\mathcal{Y}_i = f\left(x_i; \boldsymbol{\beta}\right) + \epsilon_i$ $(i = 1, ..., n)$, where f is a known expectation function (called a *calibration curve* in this context) that is monotonic over the range of interest and $\epsilon_i \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2\right)$. A common problem in regression is to predict a future response $\mathcal{Y}_0$ (or estimate the mean response $f_0 = E\left[\mathcal{Y}_0\right]$) 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) and Greenwell (2014). Oman (1998) considers the case of a random intercept and slope model and provides an approximate parametric bootstrap algorithm for inferring x_0 (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 $\mathcal{Y}_0$.

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). 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 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

$$\boldsymbol{\mathcal{Y}}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, ..., m, \qquad (1)$$

where

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

The random-effects $\boldsymbol{b}_i$ and errors $\boldsymbol{\epsilon}_i$ are often assumed to follow a normal distribution. By stacking the data, the (normal) LMM can be written concisely as

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

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

The additional term $\boldsymbol{Zb}$ in the model imposes a specific variance-covariance structure on the response vector $\boldsymbol{\mathcal{Y}}$:

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

Thus, the fixed-effects determine the mean of $\boldsymbol{\mathcal{Y}}$, while the random-effects govern the variance-covariance structure of $\boldsymbol{\mathcal{Y}}$. Different random-effects structures impose different variance-covariance structures on the response resulting in a highly flexible framework for modelling repeated measures.

The random-effects variance-covariance matrix $\boldsymbol{D}$ 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 $\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\right)$; 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 $\left(\sigma^2, \boldsymbol{\theta}\right)$ are estimated via restricted maximum likelihood (REML). The ML estimator of $\boldsymbol{\beta}$, given by

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top \widehat{\boldsymbol{V}}^{-1} \boldsymbol{X}\right) \boldsymbol{X}^\top \boldsymbol{\mathcal{Y}},$$

depends on the estimated variance components through $\widehat{\boldsymbol{V}}$ which makes it difficult to capture the variability of $\widehat{\boldsymbol{\beta}}$ 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 $\widehat{\boldsymbol{V}}$ as the true (fixed) value of $\boldsymbol{V}$. 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 for x_0

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

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

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

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 $\mathcal{N}\left\{f\left(x_0; \boldsymbol{\beta}\right), \sigma_0^2\right\}$ random variable with $\sigma_0^2 = VAR\left[R\left(x_0; \boldsymbol{b}_0\right)\right] + \sigma^2$. A natural estimator for x_0 is obtained by inverting f to obtain

$$\widehat{x}_0 = f^{-1}\left(\mathcal{Y}_0; \widehat{\boldsymbol{\beta}}\right), \qquad (2)$$

where $\widehat{\boldsymbol{\beta}}$ is the ML estimator of $\boldsymbol{\beta}$. We shall refer to Equation (2) as the inverse estimator. Note that the point estimate $\widehat{x}_0$ 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 for x_0

There is no "textbook" formula for the standard error of $\widehat{x}_0$—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 $SE\left[\widehat{x}_0\right]$, relies on the variance-covariance matrix of $\left(\mathcal{Y}_0, \widehat{\boldsymbol{\beta}}\right)^\top$; namely,

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

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

$$VAR[\widehat{x}_0] \approx \left[f_1^{-1}\left(\mathcal{Y}_0;\widehat{\boldsymbol{\beta}}\right)\right]^2 \sigma_0^2 + \left[f_2^{-1}\left(\mathcal{Y}_0;\widehat{\boldsymbol{\beta}}\right)\right]^\top \left(\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1} \left[f_2^{-1}\left(\mathcal{Y}_0;\widehat{\boldsymbol{\beta}}\right)\right].$$

To obtain $SE[\widehat{x}_0] = \left\{\widehat{VAR}[\widehat{x}_0]\right\}^{1/2}$, we simply replace σ_0^2 and $\boldsymbol{V}$ with their respective estimates $\widehat{\sigma}_0^2$ and $\widehat{\boldsymbol{V}}$. Assuming large sample normality for $\widehat{x}_0$ leads to an approximate $1-\alpha$ Wald confidence interval for x_0 of

$$CI_W(x_0) = \left(\widehat{x}_0 - SE[\widehat{x}_0]\,\Phi^{-1}(\alpha/2), \widehat{x}_0 - SE[\widehat{x}_0]\,\Phi^{-1}(1-\alpha/2)\right), \qquad (2)$$

where Φ^{-1} is the probit function.

Alternatively, one can use a parametric bootstrap esimate of the standard error, say $SE^\star[\widehat{x}_0]$; this is discussed in a later section. The advantage of using $SE^\star[\widehat{x}_0]$ over $SE[\widehat{x}_0]$ is that $SE^\star[\widehat{x}_0]$, through simulation, would take into account the variability of the estimated variance components $\widehat{\sigma}_0^2$ and $\widehat{\boldsymbol{V}}$.

Inversion interval for x_0

In the case of the simple linear regression model with i.i.d. normal errors, an exact $1-\alpha$ confidence interval for x_0 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 $\widehat{f}_0 = f\left(x_0;\widehat{\boldsymbol{\beta}}\right)$ be the predicted mean response at $x = x_0$. A prediction interval for $\mathcal{Y}_0$ at $x = x_0$ with asymptotic coverage probability $1-\alpha$ is

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

If instead, $\mathcal{Y}_0$ is observed to be y_0 and x_0 is unknown, then an asymptotic $1-\alpha$ confidence interval for the

unknown x_0 can be obtained by inverting $\mathcal{I}_\infty(x_0)$:

$$CI_I(x_0) = \left\{ x : \Phi^{-1}(\alpha/2) \leq \frac{\mathcal{Y}_0 - f\left(x; \widehat{\boldsymbol{\beta}}\right)}{\left\{\widehat{VAR}\left[\mathcal{Y}_0 - f\left(x; \widehat{\boldsymbol{\beta}}\right)\right]\right\}^{1/2}} \leq \Phi^{-1}(1-\alpha/2) \right\}. \qquad (3)$$

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 $CI_I(x_0)$ is not symmetric about $\widehat{x}_0$ and will not necessarily result in a single finite interval; see, for example, Greenwell and Kabban (2014).

Finally, notice that $CI_I(x_0)$ relies on the quantiles from a normal distribution. While it is likely that a t-distribution may be more accurate, it is difficult to determine the appropriate degrees of freedom. Oman (1998), suggests a t-distribution with $N-1$ degrees of freedom (N being the total sample size).

Monte carlo study

To assess the empirical performance of $CI_W(x_0)$ and $CI_I(x_0)$, 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 $CI_W(x_0)$ $CI_I(x_0)$ have asymptotic coverage probability close to $1-\alpha$. 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.

We consider the values 5, 10, 30, 50, and 100 for both the number of subjects m and and the number of observations per subject n. For each combination of sample sizes, we generated 1,000 data sets from a random intercept and slope model with **ADD LATER**. 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 $f(x_0; \boldsymbol{\beta}) = 500$ 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

m	Method	$n = 5$	$n = 10$	$n = 30$	$n = 50$	$n = 100$
	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 $CI_W(x_0)$ and $CI_I(x_0)$ as a function of the number of subjects (rows) and the number of observations per subject (columns).

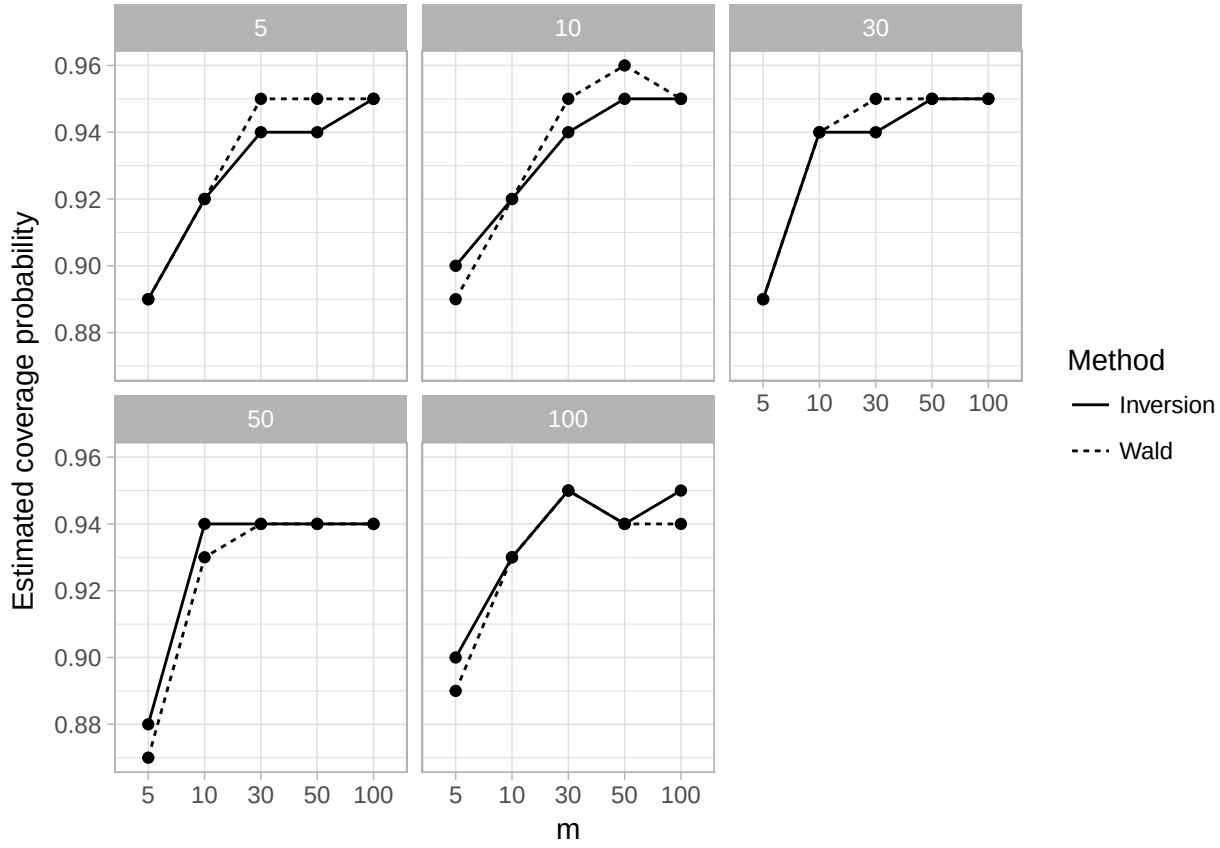

Figure 1: Estimated coverage probability for $CI_W(x_0)$ and $CI_I(x_0)$ as a function of the number of subjects (x-axis) and the number of observations per subject (y-axis).

Parametric bootstrap replicates of $\widehat{x}_0$

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 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 $\widehat{\sigma}_0^2$ be an estimate of the variance of the new observation $\mathcal{Y}_0$. An algorithm for bootstrapping $\widehat{x}_0$ in an LMM is given below. Note that step (3) is crucial for calibration problems because we need to treat y_0 as a random quantity in the bootstrap simulation, otherwise the variability of $\widehat{x}_0$ 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 $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{D}}$, and $\widehat{\sigma}^2$.
  (1) Define $\boldsymbol{y}^\star = \boldsymbol{X}\widehat{\boldsymbol{\beta}} + \boldsymbol{Z}\boldsymbol{b}^\star + \boldsymbol{\epsilon}^\star$, where $\boldsymbol{b}^\star \sim \mathcal{N}_q\left(\boldsymbol{0}, \widehat{\boldsymbol{D}}\right)$ and $\boldsymbol{\epsilon}^\star \sim \mathcal{N}_N\left(\boldsymbol{0}, \widehat{\sigma}_\epsilon^2 \boldsymbol{I}\right)$;
  (2) Update the original model using $\boldsymbol{y}^\star$ as the response vector to obtain $\widehat{\boldsymbol{\beta}}^\star$ and $\widehat{\sigma}_0^{2\star}$;
  (3) Generate $y_0^\star \sim \mathcal{N}\left(y_0, \widehat{\sigma}_0^{2\star}\right)$;
  (4) Define $\widehat{x}_0^\star = f^{-1}\left(y_0^\star; \widehat{\boldsymbol{\beta}}^\star\right)$;
- Repeat steps (1)-(4) R times.

There are many bootstrap confidence interval procedures, for instance: the percentile method (Efron 1979), the studentized bootstrap t method (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 this algorithm to adjust the previously discussed inversion interval.

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 $CI_I(x_0)$ to account for the variability of the estimated variance components. The inversion interval assumes that the *predictive pivot*

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

has a $\mathcal{N}(0,1)$ distribution. A bootstrap modified inversion interval would instead use the bootstrap distribution of

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

to estimate the sampling distribution of Q_I. If $\widehat{F}_{Q_I}$ 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

$$CI_I^\star(x_0) = \left\{x : \widehat{F}_{Q_I}(\alpha/2) \leq Q_I \leq \widehat{F}_{Q_I}(1-\alpha/2)\right\}.$$

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 V a measure of height (H) in mm and depth (D) in mm of largest ultrasound bladder images were taken. The product $HD = H \times D$ was taken as a measure of liquid volume."

A spaghetti plot of the raw data is displayed in the left side of Figure 1. As noted by Brown (1993), $(H \times D)^{3/2}$ should be linearly related to V. The transformed version of the data is displayed in the right side of Figure 1.

As indicated by Figure 1, the intercept and slopes seem to very between subjects for the transfored data; hence, the following random intercept and slope model seems appropriate:

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

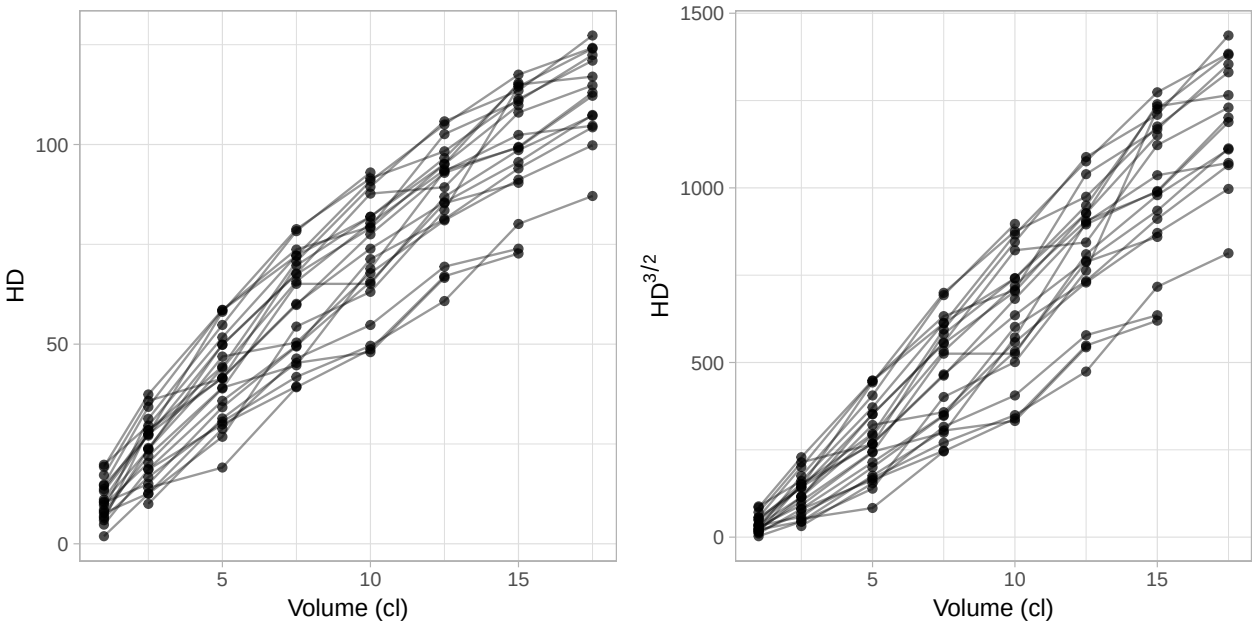

Figure 2: Bladder volume data. Left: spaghettiplot of transformed data (lines connect measurements belonging to the same subject). Right: Estimated subject specific intercepts and slopes (with one-at-a-time 95% confidence limits).

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

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

Here we also assume that the random effects are uncorrelated (i.e, $COV\left[b_{0i}, b_{1i}\right] = 0$). 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 2: 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

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 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 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. An in-depth introduction to the `investr` package is given in Greenwell and Kabban (2014). Though the paper only covers the case with independent observations, the discussion is still relevent to the use of the package with LMMs.

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)`.

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

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

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 the R function `uniroot` from the `stats` package to solve the equation $f\left(x; \widehat{\boldsymbol{\beta}}\right) - y_0 = 0$ 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 ar
```

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 $1 - \alpha$ confidence interval (where α 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 $\widehat{x}_0$ as a

function of y_0 and $\widehat{\boldsymbol{\beta}}$.

```
> 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, $\widehat{x}_0 = f^{-1}\left(y_0; \widehat{\boldsymbol{\beta}}\right)$ 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 (#).

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` (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 speicfying 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 $\widehat{x}_0$ 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 $\widehat{x}_0$, 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 $\widehat{x}_0$, based on $R =$ `9999` bootstrap replicates, are `1.974` and `0.052`, respectively. The original estimate $\widehat{x}_0$ 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 $\widehat{x}_0$, 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 (#)-(#).

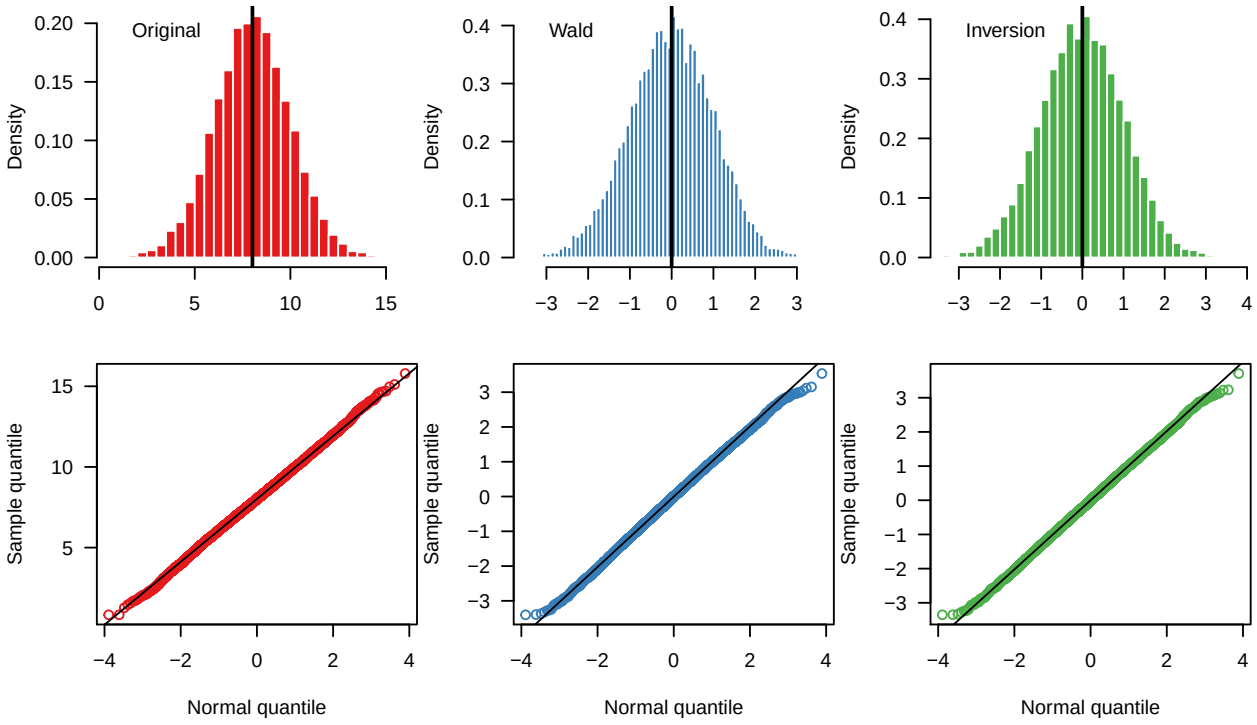

Figure 3: Graphical summary of bootstrap replicates. $R = 9,999$ bootstrap replicates of $\widehat{x}_0$ (left), Q_W (middle), and Q_I (right).

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

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

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 $\widehat{x}_0$. 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 $\widehat{F}_{Q_I}(0.025)$ and $\widehat{F}_{Q_I}(0.975)$:

```
> 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 (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
$CI_W(x_0)$	8.016	1.954	(4.185, 11.846)	7.66
$CI_I(x_0)$	8.016	—	(4.228, 11.919)	7.691
$CI^{\star}_{percentile}(x_0)$	8.016	1.974	(4.05, 11.913)	7.863
$CI^{\star}_W(x_0)$	8.016	1.974	(4.252, 11.947)	7.695
$CI^{\star}_I(x_0)$	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 tranform the response so that a straight line provided a reasonable fit. This simple form led to a closed-form solution for $\widehat{x}_0$. 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.

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.

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

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

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

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: $\widehat{x}_0 = 13.05$. Recall, from Figure 1 that each patient has a slightly nonlinear trajectory; thus, there is no reason to expect the sampling distribution of $\widehat{x}_0$ 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 ($C_W(x_0)$) will not be accurate in this case. However, the approximate predictive pivot used in the inversion interval ($C_I(x_0)$) 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.

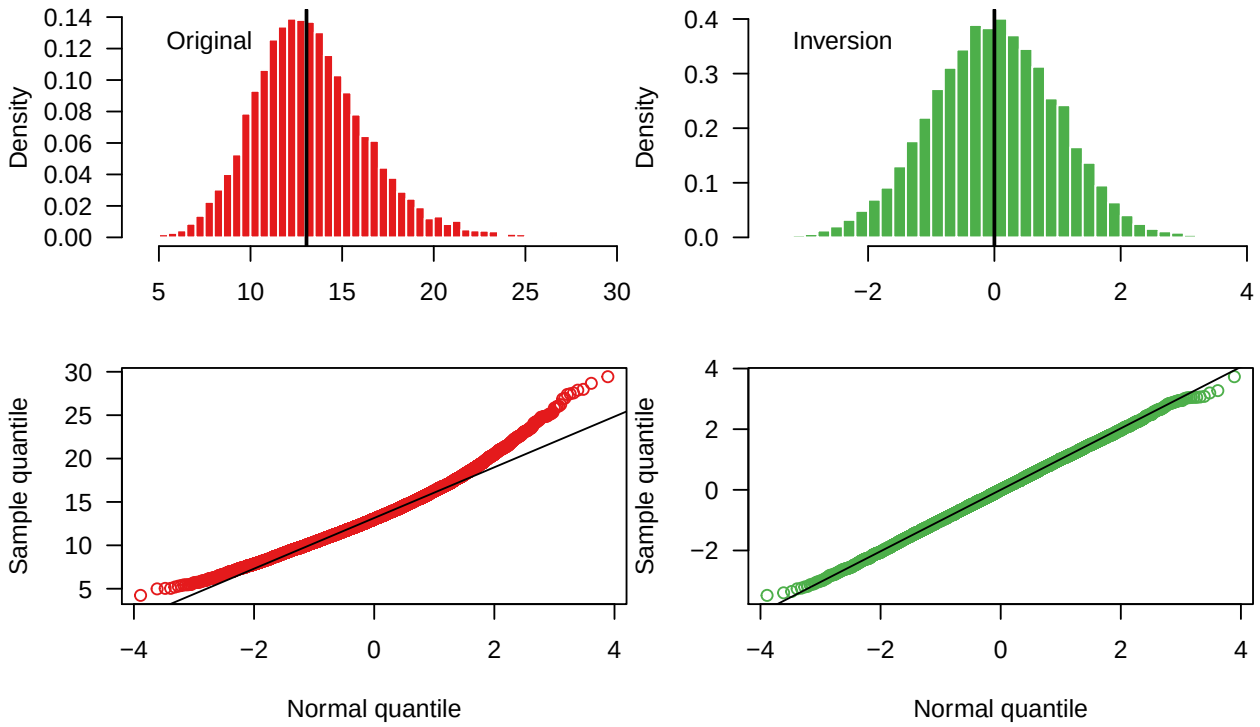

Figure 4: Graphical summary of bootstrap replicates. $R = 9,999$ bootstrap replicates of $\widehat{x}_0$ (left), Q_W (middle), and Q_I (right).

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

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.

———. 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/.