5-2024

A Comprehensive Uncertainty Quantification Methodology for Metrology Calibration and Method Comparison Problems Via Numeric Solutions to Maximum Likelihood Estimation and Parametric Bootstrapping

Aloka B. S. N. Dayarathne
Utah State University, a02384453@usu.edu

A COMPREHENSIVE UNCERTAINTY QUANTIFICATION METHODOLOGY FOR METROLOGY CALIBRATION AND METHOD COMPARISON PROBLEMS VIA NUMERIC SOLUTIONS TO MAXIMUM LIKELIHOOD ESTIMATION AND PARAMETRIC BOOTSTRAPPING

by

Aloka B S N Dayarathne

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTERS OF SCIENCE

in

Statistics

Approved:

Stephen J. Walsh, Ph.D. Major Professor	Brennan Bean, Ph.D. Committee Member
Yan Sun, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan,Utah

2024

ABSTRACT

A comprehensive uncertainty quantification methodology for metrology calibration and method comparison problems via numeric solutions to maximum likelihood estimation and parametric bootstrapping

by

Aloka B.S.N Dayarathne, Master of Science

Utah State University

Major Professor: Dr. Stephen J. Walsh

Department: Mathematics and Statistics

Straight line calibration models are extremely common in metrology. Ordinary least squares and generalized least squares are the widely used techniques to fit the calibration curve but they can produce biased estimates of the slope and intercept if the calibrants (x-values) are subject to uncertainty. Ripley and Thompson (1987) identified Functional Relationship Estimation by Maximum Likelihood (FREML) as the better fit for this particular type of problem. FREML accounts for both x and y variable uncertainties and therefore can produce unbiased estimates of the slope and intercept of the calibration curve. The objective of this project is to construct a proper mechanism of calculating the uncertainty of the final measurement quantity which accounts for and propagates all known input uncertainties. To achieve this, we apply international standard guidelines, specifically the GUM (Guide to uncertainty in Metrology) guidelines in conjunction with parametric bootstrapping to account for uncertainties in both x and y of the calibration points, and propagate them into the standard errors of the calibration line parameter estimates. This is achieved by assuming that the reported uncertainties represent the known population standard deviations. The approach is validated in a computational study where we benchmark the statistical coverage of the method and show it achieves the designed confidence. Last, we extend the model to the case where the uncertainties of the calibration inputs are no longer known, but estimated standard deviations and we

account for their uncertainties, i.e. degrees-of-freedom, and thus include these uncertainty sources in the propagation calculation.

(83 pages)

PUBLIC ABSTRACT

A comprehensive uncertainty quantification methodology for metrology calibration and method comparison problems via numeric solutions to maximum likelihood estimation and parametric bootstrapping

Aloka B.S.N Dayarathne

In metrology, the science of measurements, straight line calibration models are frequently employed. These models help understand the instrumental response to an analyte, whose chemical constituents are unknown, and predict the analyte's concentration in a sample. Techniques such as ordinary least squares and generalized least squares are commonly used to fit these calibration curves. However, these methods may yield biased estimates of slope and intercept when the calibrant, substance used to calibrate an analytical procedure with known chemical constituents (x-values), carries uncertainty. To address this, Ripley and Thompson (1987) proposed functional relationship estimation by maximum likelihood (FREML), which considers uncertainties in both x and y variables, providing unbiased estimates of slope and intercept. This project aims to develop a robust mechanism for calculating uncertainty in final measurement quantities, integrating international standard guidelines such as GUM and parametric bootstrapping to handle uncertainties in both x and y calibration points. Initial validation follows Ripley and Thompson's assumptions of known population standard deviations. Subsequently, the model is extended to accommodate cases where calibration input uncertainties are estimated, incorporating their degrees-of-freedom and propagating these uncertainties into parameter estimates.

ACKNOWLEDGMENTS

I express my deepest gratitude to my supervisor, Dr. Stephen J. Walsh, for his invaluable guidance and mentorship throughout this research. His expertise and constructive feedback were instrumental in shaping the direction and quality of this thesis. I am also thankful to my thesis committee members, Dr. Yan Sun and Dr. Brennan Bean, for their insightful comments and scholarly input.

I acknowledge the faculty members and researchers at Utah State University for fostering a stimulating academic environment and providing resources for this study. My heartfelt appreciation goes to the university staff for their efficient support in accessing research materials.

I am grateful to my family and friends for their unwavering love and encouragement. Their constant motivation has been a source of strength throughout this endeavor. I also appreciate the camaraderie of my colleagues and fellow researchers, whose collaborative spirit enriched my research experience.

Lastly, I extend my thanks to all the participants who generously shared their experiences. Their contributions were invaluable to this research. In conclusion, the completion of this thesis would not have been possible without the support and contributions of the individuals and institutions mentioned above.

Thank you all for your invaluable contributions.

Aloka B. S. N. Dayarathne

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
LITERATURE REVIEW: FUNCTIONAL RELATIONSHIP ESTIMATION BY MAXIMUM-LIKELIHOOD FOR METROLOGY CALIBRATION AND METHOD COMPARISON PROBLEMS

1.1
The Role of Simple Linear Regression in Metrology

Calibration is the process of determining the relationship between the measurement results of an instrument response or system and the corresponding known values of the measured quantity (e.g. a chemical compound concentration). According to International Vocabulary of Metrology (VIM), there are three major concepts associated with calibration i.e. the measurand, calibration specific conditions and the relationship between measured or indicated values and the reference values. Describing the relationship between measured or indicated values and those of the reference values includes calibration equations, curves, or mathematical models that establish the connection between the measured/indicated values and the reference values. The ultimate goal of determining the mathematical relationships between standard quantities and the indications of instruments is evaluating new unknown quantities regards to the new indications. Typically, understanding this relationship enables the assessment of accuracy, determination of measurement uncertainties, and traceability to established standards [6].

1.1.1
As a Component of the Measurement Procedure

One aspect of calibration curves is establishing instrument linearity. Calibration curves are used to assess and correct for any non-linearities in the response of measuring instruments. By plotting a calibration curve, which involves taking multiple measurements at different known values of the measured quantity, it becomes possible to determine if the instrument exhibits any deviations from linearity. Non-linearities can then be accounted for during subsequent measurements,

leading to improved accuracy. On the other hand, calibration curves can be used to determine the instrument sensitivity and the resolution. The slope of a calibration curve indicates the sensitivity of the instrument, i.e., the change in instrument output per unit change in the measured quantity. Additionally, the calibration curve can reveal the resolution of the instrument, which is the smallest change in the measured quantity that can be detected and quantified. Also they are used to determine measurement uncertainty, which quantifies the confidence or range of plausible values associated with a measured quantity. By considering the scatter or dispersion of calibration points around the curve, it is possible to estimate the measurement uncertainty associated with the measurand. Calibration curves provide valuable information for uncertainty evaluation and can be used to assign appropriate measurement uncertainties to measurement results.

1.1.2

Comparing Relative Method Bias

Method comparison is the process of comparing the results obtained from different methods of measurement. In the scenario where the reference values for the samples are unknown but have been measured using an established method, one approach to comparing measurement methods is to perform a simple linear regression analysis by comparing the results obtained from a new method with those obtained from the established method. In such cases, it is often observed that the regression line has a smaller slope, indicating a systematic deviation between the two methods. This discrepancy might imply that the new method exhibits a bias, for example showing higher values at low concentrations and lower values at high concentrations compared to the old method. One component of method comparison studies is the assessment of relative bias. Most authors use ordinary regression techniques for this purpose which is incapable of handling both response and explanatory variable uncertainties at the same time. Since the most common regression techniques such as simple and weighted linear regression are not symmetrical techniques (since both the methods are designed for predicting y on x), these methods are usually accurate for predicting the bias of new variable (Y) relative to the existing reference variable (X) [8].

1.2

Linear Calibration Curves

When seeking to estimate the functional relationship between two variables, it is common to assume a model that describes their interdependency. This model can take a linear or non-linear

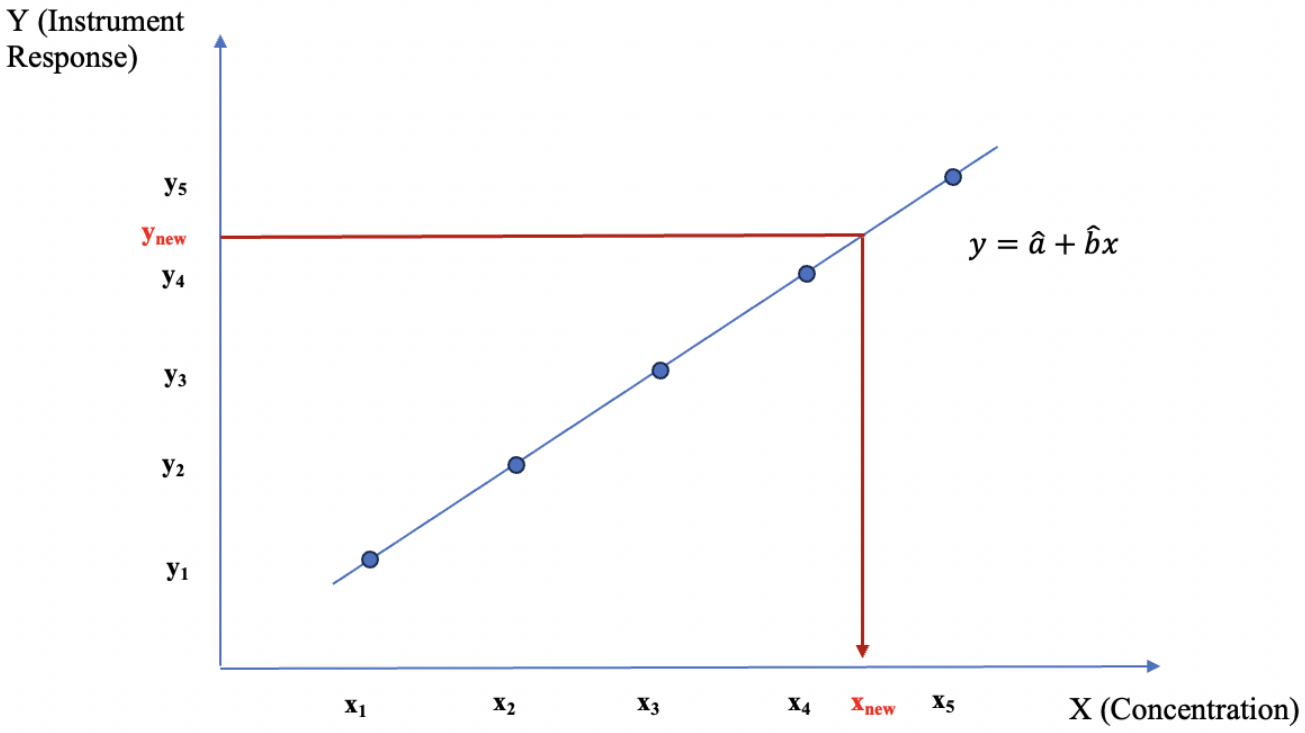


Figure 1.1: An illustration of the functional Relationship between measurand concentration and the instrument response.

form, depending on the nature of the relationship between the variables. Once a suitable model is selected, the parameters of the model need to be estimated. This is often accomplished using Maximum Likelihood Estimation (MLE). In the calibration exercise, the analyst prepares n samples with known concentrations, denoted as x_i. Then they submit the samples to the instrument to attain a response y_i. This yields a data set $\{x_i,\ y_i\}_{i=1}^{n}$ to which the analyst fits a calibration curve of the type depicted in Eq. (1.1). Then for a new sample with an unknown concentration x_{new}, they retrieve the sampling response y_{new} and estimate concentration by inverting the calibration curve as illustrated in Eq. (1.2).

$$y = \hat{a} + \hat{b}x \tag{1.1}$$

$$\hat{x}_{new} = \frac{y_{new} - \hat{a}}{\hat{b}} \tag{1.2}$$

The challenge in calibration problems arises from chemists attempting inverse regression. Consequently, the uncertainty of the final measurement x_{new} is influenced by the uncertainty stemming from the linear functional relationship as well as the newly measured value y_{new}. Hence, there should be a method of incorporating all these uncertainties into the variance of x_{new}. Conversely,

in method comparison problems, where both X and Y variables are subject to measurement errors, it is essential to account for all these measurement errors in the model coefficients. This ensures unbiased estimates and enables the identification of relative method bias [10]. Failure to propagate both X and Y variable uncertainties into the model coefficients can lead to erroneous conclusions when quantifying the bias of the new method compared to the reference method.

Figure 1.1 illustrates the functional relationship of concentration and the instrument response for the calibration problem. MLE involves identifying the parameter values that maximize the likelihood function. The likelihood function represents the probability of obtaining the observed data given the chosen model and its corresponding parameters. This function is dependent on both the model parameters and the observed data. By maximizing the likelihood function, MLE provides an effective means of reducing the uncertainties associated with the variables being studied, namely X and Y. MLE enables the determination of parameter values that optimize the fit between the model and the observed data, thereby enhancing our understanding of the relationship between the variables and reducing uncertainties in X and Y.

Within the field of metrology, the likelihood function is frequently constructed by comparing the instrument response with the corresponding known values. The aim is to maximize this likelihood function by adjusting the model parameters, thereby obtaining the most accurate estimation of the functional relationship between the variables being studied. The MLE method is highly prevalent in metrology, particularly in calibration and method comparison scenarios. Its appeal lies in its rigorous and statistically grounded approach to estimating the functional relationship between variables. By employing MLE, metrologist can determine the best parameter estimates while also quantifying the uncertainty associated with these estimates [10]. This ability to assess uncertainty is crucial in metrology as it enables the determination of measurement uncertainties for calibrated instruments or systems, leading to more reliable and accurate measurements. In essence, the MLE serves as a valuable tool in metrology by providing a robust framework for estimating functional relationships, allowing for uncertainty evaluation, and enhancing the overall quality of measurements and calibration processes.

1.2.1 Ordinary Least Squares Regression - OLS

Ordinary least squares regression is the most often used technique in linear pattern recognition. OLS is used when the x variable is known exactly without uncertainties and the y variable's uncertainty is unknown and assumed to be constant. Therefore, OLS regression line of y on x min-

imizes the sum of the squares of the residuals, where the vertical distances between the observed data points and the predicted values on the line. So, in this method the measurement model is of the form; $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where, $\mathbf{y} = \begin{pmatrix} y_1 \; y_2 \; \dots \; y_n \end{pmatrix}'$ is an $n \times 1$ vector of the dependent variable y , $\mathbf{X} = \begin{bmatrix} \mathbf{1}_n & \mathbf{x} \end{bmatrix}$ is an $(n \times 2)$ model matrix, $\boldsymbol{\beta} = \begin{pmatrix} a & b \end{pmatrix}'$ is a vector of regression coefficients a and b and $\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \; \epsilon_2 \; \dots \; \epsilon_n \end{pmatrix}'$ is an $(n \times 1)$ vector of errors. Major assumption of OLS regression is that the $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$. Moreover, the estimated vector of regression coefficients $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, $Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ and $\hat{\sigma}^2 = \frac{\mathbf{y}'(\mathbf{I_n} - \mathbf{H})\mathbf{y}}{n-2}$ where, $(\mathbf{I_n} - \mathbf{H})_{\mathbf{n}\times\mathbf{n}}$ is the annihilator matrix and $\mathbf{H}_{\mathbf{n}\times\mathbf{n}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the hat matrix and p is the number of parameters. The major drawback of using OLS model in calibration problems is that in OLS regression, the explanatory variable x is assumed to be exactly known without uncertainties.

1.2.2

Generalized Least Squares Regression - GLS

Generalized Least Squares Regression is frequently used when the response data violates the major least squares regression assumption of homoscedasticity (having equal variances). In GLS regression, the model assumes that the errors follow a specific covariance structure, often represented by a variance-covariance matrix. The estimated coefficients are obtained by minimizing the weighted sum-of-squared residuals, where the weights are derived from the estimated covariance structure. So, in this method the measurement model is of the form, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ .Where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ and;

$$\boldsymbol{\Sigma} = \begin{bmatrix} {\sigma_1}^2 & 0 & \dots & \\ 0 & {\sigma_2}^2 & & \\ \vdots & & \ddots & \\ & & & {\sigma_n}^2 \end{bmatrix}$$

Each σ_i represents the known measurement uncertainty for response y_i [7]. GLS model satisfies all the OLS regression assumptions except homoscedasticity [14]. Also in this model, the estimated vector of regression coefficients $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{Y}$ and their variance is given by; $Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}$. σ^2 can be estimated by using, $\hat{\sigma}^2 = \frac{\mathbf{y}'\boldsymbol{\Sigma}^{-1}(\mathbf{I_n} - \mathbf{P})\mathbf{y}}{n-2}$ where, $\mathbf{P} = \mathbf{X}(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}$. Major drawback of using GLS model in calibration problems is that in GLS regression, the explanatory variable x is assumed to be exactly known without uncertainties, and this assumption is often violated with chemistry and metrology data and applications.

1.2.3

FREML - Functional Relationship Estimation by Maximum Likelihood

Functional relationship estimation through maximum likelihood is indeed a widely used statistical approach in metrology calibration and method comparison problems where, the uncertainties occur on both x and y ; $\boldsymbol{\sigma_y^2} = \begin{pmatrix} \sigma_{y_1}^2 & \sigma_{y_2}^2 & \dots & \sigma_{y_n}^2 \end{pmatrix}'$ and $\boldsymbol{\sigma_x^2} = \begin{pmatrix} \sigma_{x_1}^2 & \sigma_{x_2}^2 & \dots & \sigma_{x_n}^2 \end{pmatrix}'$ are known. By maximizing the likelihood function, which represents the probability of obtaining the observed data given the model and its parameters, MLE identifies the parameter values that are most consistent with the observed data. This estimation process allows for the determination of the functional relationship between variables and provides a statistical foundation for calibration and method comparison in metrology. In fact, closed form expressions for maximum likelihood estimators do not exist and must be found numerically [8]. Section 1.3 discusses the derivation of the FREML model in detail.

1.3

The FREML Model

1.3.1

The Model

The FREML model is presented in the perspective of the calibration problem. Let x_i be the measurand value of the i^{th} calibrant, and y_i be the instrument response to this calibrant. We assume that there are n samples in the calibration and so $i \in \{1, 2, \dots, n\}$. These are measured values and we assume that the true unknown values are u_i and v_i respectively. We assume a simple-linear relationship between the true values as given in Eq. (1.3), where a and b represent the intercept and slope of the true calibration line, respectively. Consequently, the measurement-error models can be

derived as depicts in Eq. (1.4) and Eq. (1.5).

$$v_i = a + bu_i \tag{1.3}$$

$$y_i = v_i + \epsilon_i = a + bu_i + \epsilon_i \tag{1.4}$$

$$x_i = u_i + \eta_i \tag{1.5}$$

$$\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{y_i}^2)$$

$$\eta_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{x_i}^2)$$

ϵ_i and η_i are independent.

In FREML model we assume that the errors in both directions are normally distributed with zero mean and non-constant variances. Additionally, we assume that all measurement uncertainties, namely $\boldsymbol{\sigma_y^2} = \begin{pmatrix} \sigma_{y_1}^2 & \sigma_{y_2}^2 & \dots & \sigma_{y_n}^2 \end{pmatrix}'$ and $\boldsymbol{\sigma_x^2} = \begin{pmatrix} \sigma_{x_1}^2 & \sigma_{x_2}^2 & \dots & \sigma_{x_n}^2 \end{pmatrix}'$ are known. Therefore, the probability distribution functions for this study are given by Eq. (1.6) and Eq. (1.7).

$$f(x_i|u_i, \sigma_{x_i}^2) = \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} \exp\left\{-\frac{1}{2\sigma_{x_i}^2}(x_i - u_i)^2\right\} \tag{1.6}$$

$$f(y_i|a, b, u_i, \sigma_{y_i}^2) = \frac{1}{\sqrt{2\pi\sigma_{y_i}^2}} \exp\left\{-\frac{1}{2\sigma_{y_i}^2}(y_i - (a + bu_i))^2\right\} \tag{1.7}$$

1.3.2

Parameter Estimation Via Maximum Likelihood

For a vector of true calibration values $\boldsymbol{u} = \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix}'$ which are now viewed as parameters to be estimated and slope and intercept a and b, the likelihood function is given by Eq. (1.9) [8].

$$\begin{aligned} L(a, b, \boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\sigma_x^2}, \boldsymbol{y}, \boldsymbol{\sigma_y^2}) &:= f(\boldsymbol{x}, \boldsymbol{\sigma_x^2}, \boldsymbol{y}, \boldsymbol{\sigma_y^2}|a, b, \boldsymbol{u}) \qquad (1.8) \\ &= \prod_{i=1}^{n} f(x_i|\boldsymbol{u}, \boldsymbol{\sigma_x^2}) f(y_i|a, b, \boldsymbol{u}, \boldsymbol{\sigma_y^2}) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} \exp\left\{-\frac{1}{2\sigma_{x_i}^2}(x_i - u_i)^2\right\} \frac{1}{\sqrt{2\pi\sigma_{y_i}^2}} \exp\left\{-\frac{1}{2\sigma_{y_i}^2}(y_i - (a + bu_i))^2\right\} \\ &= \prod_{i=1}^{n} \left(\frac{1}{(4\pi\sigma_{x_i}^2\sigma_{y_i}^2)^{1/2}}\right) \exp\left\{\frac{-\sum_{1=1}^{n}(x_i - u_i)^2}{2\sigma_{x_i}^2} + \frac{-\sum_{1=1}^{n}(y_i - (a + bu_i))^2}{2\sigma_{y_i}^2}\right\} \qquad (1.9) \end{aligned}$$

The negative log-likelihood is then;

$$-l(a,b,\boldsymbol{u}|\boldsymbol{x},\boldsymbol{\sigma_x^2},\boldsymbol{y},\boldsymbol{\sigma_y^2}) := -\log L(a,b,\boldsymbol{u}|\boldsymbol{x},\boldsymbol{\sigma_x^2},\boldsymbol{y},\boldsymbol{\sigma_y^2}) \tag{1.10}$$

$$= -\log\left(\prod_{i=1}^{n}\frac{1}{(4\pi\sigma_{x_i}^2\sigma_{y_i}^2)^{1/2}}\exp\left\{\frac{-\sum_{1=1}^{n}(x_i-u_i)^2}{2\sigma_{x_i}^2}+\frac{-\sum_{1=1}^{n}(y_i-(a+bu_i))^2}{2\sigma_{y_i}^2}\right\}\right)$$

$$= -\sum_{i=1}^{n}\log\left[\frac{1}{(4\pi\sigma_{x_i}^2\sigma_{y_i}^2)^{1/2}}\right]+\frac{\sum_{1=1}^{n}(x_i-u_i)^2}{2\sigma_{x_i}^2}+\frac{\sum_{i=1}^{n}(y_i-(a+bu_i))^2}{2\sigma_{y_i}^2} \tag{1.11}$$

$$\{\hat{a},\hat{b},\hat{\boldsymbol{u}}\} := \underset{\substack{a,b\in(-\infty,\infty)\\ u_i\in(0,\infty)\forall i}}{\operatorname{argmin}} -l(a,b,\boldsymbol{u}|\boldsymbol{x},\boldsymbol{\sigma_x^2},\boldsymbol{y},\boldsymbol{\sigma_y^2}) \tag{1.12}$$

Given the negative log-likelihood, the maximum-likelihood parameter estimates are defined as shown in Eq. (1.12). Previous studies have endeavored to derive mathematical formulas to evaluate the parameter estimates of the FREML model. However, due to the likelihood function for calibration problems involving $n+2$ parameters to be estimated by minimizing the negative log likelihood function, finding a closed-form mathematical solution for the estimation process is challenging [8]. Therefore, in this study, we propose to employ numerical maximum likelihood estimation using R programming software to determine the parameter estimates.

1.4

Conclusions

In the calibration or method comparison problems the commonly found issue is having the x variable uncertainties reported as data. In such problems the usual regression techniques such as ordinary least squares or generalized least squares are not appropriate. Maximum likelihood estimation constructs the joint probability model that has complete information on Y as well as X which is an applicable method in these specific cases. An issue arises when estimating the parameters since calibration/method comparison problem may have more than two parameters in general (e.g. slope, intercept and the true u values). Finding a mathematical solution to minimize the negative log likelihood is challenging, but computational solutions can be easily obtained. FREML gives estimates for $\hat{a},\hat{b},\hat{u}_i, i=1,\ldots,n$. But it fails to give uncertainties of $\hat{a},\hat{b},\hat{u}_i$, SE($\hat{a}$), SE($\hat{b}$) and SE($\hat{u}_i$). Though FREML evaluates the point estimates for parameters, the derivation is quite complex [8].

Once the model coefficient estimates are determined, my next objective is to establish a suitable method for assessing the uncertainty associated with the final measurement of the calibration

problem, namely 'x_{new}'. This entails incorporating the final uncertainty arising from both the model coefficients and the newly measured value 'y_{new}'. Chapter 2 focuses on implementing an appropriate approach(Monte-Carlo simulation) to evaluate the final uncertainty and to estimate the uncertainty arising from the model coefficients.

CHAPTER 2
LITERATURE REVIEW: GUM, THE GUIDE TO UNCERTAINTY IN METROLOGY

2.1
Introduction to GUM

The Guide to Uncertainty in Metrology (GUM) defines uncertainty as "a parameter associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand" [3]. Therefore, errors in measurements can be explained by uncertainty of measurements and uncertainty of analysis. GUM provides a framework for estimating and expressing uncertainty, including a description of the sources of uncertainty, the methods for combining them, and the means for expressing the results. It also provides guidance on the reporting of uncertainty in measurement results, including the use of probability distributions, coverage factors, confidence intervals, and error bars.

GUM is widely used in fields such as chemistry, physics, engineering, and metrology (the science of measurement). It is also an important reference for accreditation bodies, regulators, and others who require measurements to be traceable and of known quality. The final measurand, the quantity of interest, depends on several measurements that are not always exact. When a quantity is measured, the measurand depends upon the measuring system, the measurement procedure, the skill of the operator, the environment, and other random and systematic effects. Even if repetitions are made (i.e. the quantity is measured several times), a slightly different measurement value is usually obtained each time [4]. Measurement errors in GUM are categorized into two main types: systematic errors and random errors. Systematic errors, also known as measurement biases, arise from a consistent offset present in the measured quantity value. On the other hand, random errors occur when repeated measurements yield different values, making the next measurement unpredictable based solely on previous values. Both types of errors can stem from various contributing factors.

Measurement presents a challenge in determining the best way to communicate the information obtained about the quantity being measured, known as the measurand. Before the introduction

of GUM, one approach was to express the measurement result by providing a best estimate of the measurand, accompanied by information about systematic and random errors (through an error analysis). However, GUM introduced a new perspective on measurement by emphasizing the expression of the perceived quality of the measurement result. This is achieved by providing a best estimate of the measurand, along with a measurement uncertainty value, rather than relying solely on an error analysis. Figure 2.1 depicts the GUM uncertainty framework for the calibration curve model via FREML as shown in Chapter 1. The GUM uncertainty framework is a versatile and widely applicable approach for estimating and expressing uncertainty in various measurement scenarios. While it can be applied in many circumstances, there are specific conditions where the GUM framework yields exact and precise results. One such circumstance is when the measurement function exhibits linearity with respect to the input quantities, and the probability distributions of these quantities follow a Gaussian distribution. In this case, the GUM uncertainty framework provides exact and rigorous uncertainty statements. When the measurement function is linear, it means that the relationship between the input quantities and the measurand can be expressed as a linear combination of the input quantities. This simplifies the uncertainty analysis, as the propagation of uncertainties can be performed using straightforward mathematical calculations. However, it is important to note that the GUM framework can also be applied in more general cases, where non-linear relationships or non-Gaussian distributions are present, although more advanced techniques and approximations may be required in such situations. Producing measurements via inverting a linear calibration curve is an example of a non-linear measurement equation, as will be shown in Section 2.2.

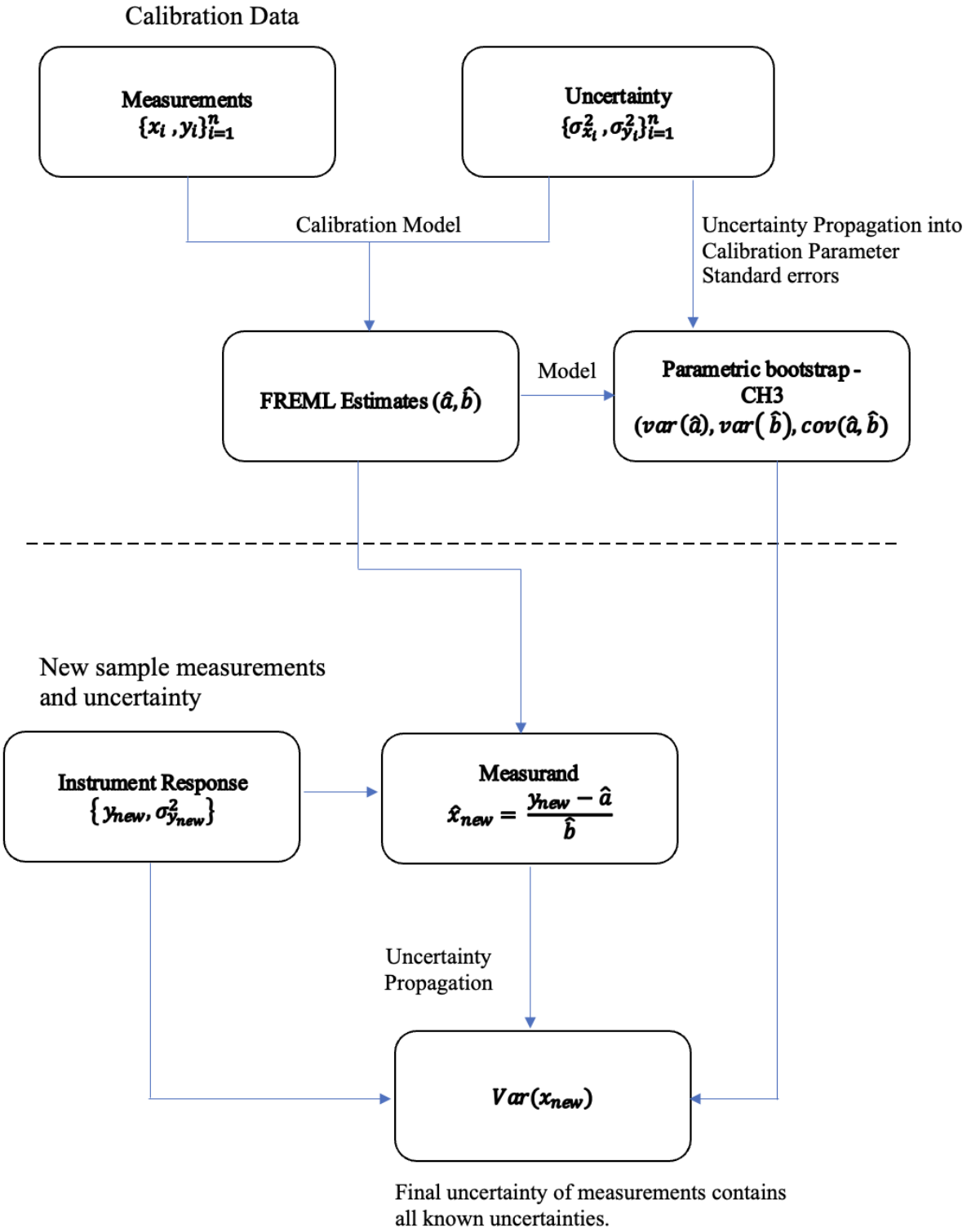


Figure 2.1: Measurement uncertainty evaluation using the GUM uncertainty framework for the calibration curve model.

2.2

Mathematical and Statistical Perspective of the GUM

In a measurement model, the factors necessary to define a measurand are referred to as input quantities. These input quantities, denoted as $X_1, X_2, \ldots, X_N$, are part of a functional relationship expressed as $Y = f(X_1, X_2, \ldots, X_N)$ where Y is the measurement result. The measurand represents the output quantity in this model. However, the true values of the input quantities $(X_1, X_2, \ldots, X_N)$ are generally unknown. In the GUM approach, the input quantities $(X_1, X_2, \ldots, X_N)$ are treated as random variables and characterized by probability distributions. This means that their values are not precisely known. Uncertainties of those unknown $u(X_i)$ are produced under two major concepts usually referred as "type A and type B". Information about an input quantity, $u(X_i)$, is derived either from repeated measurements or observations (referred to as Type A evaluation of uncertainty), or through scientific judgment and other relevant information regarding the possible values of the quantity (known as Type B evaluation of uncertainty) [1]. These methods help to establish knowledge about the input quantities and their associated uncertainties.

When conducting Type A evaluations of measurement uncertainty, it is common to assume that the distribution that best describes an input quantity X, based on repeated independent indication values, is a Gaussian distribution (also known as a normal distribution). In this case, the expectation of X is equal to the average repeated measurements, and the standard deviation of X is equal to the standard deviation of the repeated measurements, and so the uncertainty of the mean is $u(\bar{x}) = \frac{S_x}{\sqrt{n}}$. There are three forms of uncertainties: standard, combined and expanded uncertainty. Standard uncertainty is often calculated using the observed data (Type A) or expert opinion (Type B). The combined uncertainty requires a technique of propagating uncertainty in measurements on the model $Y = f(X_1, X_2, \ldots, X_N)$. Expanded uncertainty quantifies confidence by an interval about the results of a measurement. It is calculated using the standard or combined standard uncertainty multiplied by the coverage factor which is determined by degrees-of-freedom and a required significance [13].

When uncertainty is evaluated based on a small number of indication values, which can be seen as instances of an indication quantity characterized by a Gaussian distribution, a t-distribution can be used. The t-distribution takes into account the smaller sample size and tends to have fatter tails compared to the Gaussian distribution, accommodating the increased uncertainty associated with limited data. Therefore, the t-distribution is employed when the number of indication values is limited in order to provide a more appropriate representation of the uncertainty. Moreover,

degrees-of-freedom is another important concept that comes along with the t-distribution. Degrees-of-freedom refers to the number of values in a calculation that are free to vary. The degrees-of-freedom in the t-distribution are associated with the number of independent pieces of information available in the data. Specifically, in the context of hypothesis testing or constructing confidence intervals, degrees-of-freedom are used to determine the critical values or probabilities associated with the t-distribution. As the degrees-of-freedom increase, the t-distribution approaches the shape of the standard normal distribution. For large sample sizes (typically considered as degrees-of-freedom greater than 30), the t-distribution becomes virtually indistinguishable from the standard normal distribution [3].

2.2.1 Measurement Equation

Recall in Eq. (1.2) which represents the measurement equation of the calibration problem. Here, $\hat{x}_{new}$ denotes the new concentration of the compound for a measurement index of the machine y_{new}. $\hat{a}$ and $\hat{b}$ are the estimates of intercept and slope of the calibration curve, which are estimated via maximum likelihood under the FREML model.

2.2.2 Propagation of Uncertainty

This section briefly describes how GUM recommends quantifying uncertainty of measurement Y given uncertainties of input quantities $X_1, X_2, \ldots, X_N$. Let the measurement equation be:

$$Y = f(X_1, X_2, \ldots, X_n) \tag{2.1}$$

According to the law of uncertainty propagation stated in GUM, the variance of the predicted value can be written as:

$$u^2(y) = \sum_i \left[\frac{\delta\ Y}{\delta\ X_i}\right]^2 u^2(x_i) + 2\sum_i \sum_{j \neq i} \frac{\delta\ Y}{\delta\ X_i} \frac{\delta\ Y}{\delta\ X_j} cov(x_i, x_j) \tag{2.2}$$

In calibration problem, the variance of the X_{new} is:

$$var(\hat{x}_{new}) = var\left(\frac{y_{new} - \hat{a}}{\hat{b}}\right) \tag{2.3}$$

In this equation, the partial derivatives are determined using an analytical or numerical method. Now that the measurement equation is a function of 3 random variables, Y, $\hat{a}$, and $\hat{b}$, the law of error propagation is used to derive the variance formula. According to the Eq. (2.3),

$$
\begin{aligned}
var\left(\hat{x}_{new}\right) =& var\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right) \qquad (2.4)\\
&= \left[\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta y_{new}}\right]^2 var(y_{new}) + \left[\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta \hat{a}}\right]^2 var(\hat{a}) + \left[\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta\,\hat{b}}\right]^2 var(\hat{b})\\
&+ 2\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta y_{new}}\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta \hat{a}} cov(y_{new},\hat{a})\\
&+ 2\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta y_{new}}\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta \hat{b}} cov(y_{new},\hat{b}) + 2\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta \hat{a}}\frac{\delta\left(\frac{y_{new}-\hat{a}}{\hat{b}}\right)}{\delta \hat{b}} cov(\hat{a},\hat{b})\\
&= \frac{1}{\hat{b}^2}\left[var(y_{new}) + var(\hat{a}) + \frac{(y_{new}-\hat{a})^2}{\hat{b}^2} var(\hat{b}) + 2\frac{(y_{new}-\hat{a})}{\hat{b}} cov(\hat{a},\hat{b})\right] \qquad (2.5)
\end{aligned}
$$

The derivation of the Eq. (2.5) removes $cov(y_{new},\hat{a})$ and the $cov(y_{new},\hat{b})$ as it is commonly assumed that y is independent of both $\hat{a}$ and $\hat{b}$. According to Eq. (2.5), $var(y_{new})$, $var(\hat{a})$, $var(\hat{b})$ and $cov(\hat{a},\hat{b})$ are required to estimate the $var(\hat{x}_{new})$. In calibration problem, $var(y_{new})$ is a given data by the experimenter and a methodology is required to produce $var(\hat{a})$, $var(\hat{b})$ and $cov(\hat{a},\hat{b})$. We will provide such a methodology in Chapter 3.

2.3

GUM guidelines regarding the FREML model used in linear calibration

Primarily, when applying the FREML model it is mandatory to check the randomness of the input data. If every input is random, or if each X_i, X_j are constants or else if there is insufficient information to evaluate the covariances between each pair of inputs, the covariance associated with the estimates of two input quantities X_i and X_j may be taken to be zero or treated as insignificant. In real-world scenarios, input quantities frequently exhibit correlation due to factors such as the use of common physical measurement standards, measuring instruments, reference datums, or measurement methods with notable uncertainties. In such cases, the associated uncertainty should be calculated using the Eq. (2.3). Variances of $\hat{a}$, $\hat{b}$ and covariance of $\hat{a}$ and $\hat{b}$ can be obtained from FREML and parametric bootstrapping. Starting with the FREML model, Monte Carlo simulation and bootstrap procedure can generate thousands of similar data of the existing data set, and then the empirical sampling distributions of respective regression coefficients. Afterwards, the uncertain-

ties of $\hat{a}$ and $\hat{b}$ can be derived from those sampling distributions. We will develop and validate this approach in the Chapter 3.

CHAPTER 3

UNCERTAINTY PROPAGATION FOR THE FREML MODEL VIA PARAMETRIC BOOTSTRAPPING FOR KNOWN AND REPORTED MEASUREMENT UNCERTAINTIES

3.1

Introduction

This chapter briefly discuss about the summary of FREML model and maximum likelihood function and then, the implication of these two methods in calibration is illustrated by an example problem taken from Ripley & Thompson (1987) [8]. Then uncertainty propagation using parametric bootstrapping is discussed and applied the new mechanism to the illustrative example to check whether bootstrap technique gives exact same results in the Ripley & Thompson (1987)[8]. End of the chapter discuss about the validation of the application of parametric bootstrap into FREML and the final conclusion.

3.1.1

FREML Model

Recall the Eq. (1.3) in Chapter 1. In the presence of both x and y variable uncertainties, the model that accounts for both variable uncertainties to address the relationship of x and y is the FREML model. Therefore, the true linear relationship lies between the expected $x(u)$ and expected $y(v)$, where a and b represent the intercept and slope of the true calibration line, as shown in Eq. (1.3). Thus, the measurement-error models are

$$y_i = v_i + \epsilon_i = a + bu_i + \epsilon_i$$

$$x_i = u_i + \eta_i$$

Where, ϵ is the error of observed y and the actual y (v) while η is the error of the observed x and the actual x (u). The assumptions of the FREML model are

$$\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{y_i}^2)$$
$$\eta_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{x_i}^2)$$
$$\epsilon_i \text{ and } \eta_i \text{ are independent.}$$

This chapter focuses on determining the uncertainty of the parameter estimates of the FREML model; $\hat{a}$ and $\hat{b}$, when the x and y uncertainties are known.

3.1.2

Maximum Likelihood Estimates

According to this problem, there are n+2 parameters to be estimated (a, b and $\{u_1, u_2, \ldots, u_n\}$). Maximum Likelihood Estimation (MLE) is used to estimate the parameters of the FREML model. MLEs maximizes the likelihood function (or minimize the negative log-likelihood function). For the calibration problem, the negative log likelihood function is of the form,

$$\begin{aligned} -l(a, b, \boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\sigma_x^2}, \boldsymbol{y}, \boldsymbol{\sigma_y^2}) &:= -\log L(a, b, \boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\sigma_x^2}, \boldsymbol{y}, \boldsymbol{\sigma_y^2}) \\ &= -\sum_{i=1}^{n} \log \left[\frac{1}{(4\pi\sigma_{x_i}^2\sigma_{y_i}^2)^{1/2}} \right] + \frac{\sum_{1=1}^{n}(x_i - u_i)^2}{2\sigma_{x_i}^2} + \frac{\sum_{1=1}^{n}(y_i - (a + bu_i))^2}{2\sigma_{y_i}^2} \end{aligned} \tag{3.1}$$

In order to illustrate the concept of FREML in mass spectrometry we now present an example. Table 3.1 depicts the Arsenic(V) concentration of 30 natural water determined by two different methods. The data are taken from Ripley & Thompson (1987) [8]. Method 1 used for determining x values is selective reduction and atomic absorption spectrometry where as the method 2 used to determine y is cold trapping and atomic emission spectrometry. x and y are the means of Arsenic(V) concentrations and σ_x and σ_y are the standard deviations of x and y respectively. This is and example of method comparisons in the field of mass spectrometry.

Comparing the two methods using a FREML model evaluates a model of expected concentrations which minimizes relative method bias. Figure 3.1 illustrates the fitted ordinary least square regression and FREML linear models for this scenario. It can be seen that the FREML is resulting a model that is different from the OLS model. As the figure indicates, both x and y variables have uncertainties in them and OLS regression does not propagate x variable uncertainty while FREML does. Also among the two models, FREML does the better job propagating heterogeneous uncertain-

Table 3.1: Arsenic(V) concentration in 30 natural waters determined by two methods. Data from Ripley & Thompson (1987) [8].

x(Method 1)/μgl^{-1}	$\sigma(x_i)$	y(Method 2)/μgl^{-1}	$\sigma(y_i)$
8.71	1.92	7.35	2.07
7.01	1.56	7.92	2.23
3.28	0.76	3.4	0.96
5.6	1.26	5.44	1.53
1.55	0.39	2.07	0.59
1.75	0.43	2.29	0.65
0.73	0.22	0.66	0.19
3.66	0.84	3.43	0.97
0.9	0.25	1.25	0.36
9.39	2.07	6.58	1.85
4.39	1	3.31	0.93
3.69	0.84	2.72	0.77
0.34	0.13	2.32	0.66
1.94	0.47	1.5	0.43
2.07	0.5	3.5	0.99
1.38	0.36	1.17	0.33
1.81	0.45	2.31	0.66
1.27	0.33	1.88	0.54
0.82	0.23	0.44	0.13
1.88	0.46	1.37	0.4
5.66	1.27	7.04	1.98
0	0.06	0	0.01
0	0.06	0.49	0.15
0.4	0.15	1.29	0.37
0	0.06	0.37	0.12
1.98	0.48	2.16	0.62
10.21	2.24	12.53	3.51
4.64	1.05	3.9	1.1
5.66	1.27	4.66	1.31
19.25	4.18	15.86	4.45

ties and therefore FREML is a good approach in the method comparison problem. Hypothetically, if the two methods gives identical results without concerning the chance error, the fitted model should be inline with $y = x$. Therefore, in method comparison problems, for a fitted line of $y = a + bx$, it is needed to check the hypothesis of

$$H_{01} : Intercept(a) = 0 \text{ and } H_{02} : Slope(b) = 1$$

These hypotheses may be assessed in several ways, one of which includes calculating the following confidence intervals shown in Eq. (3.3) and Eq. (3.4).

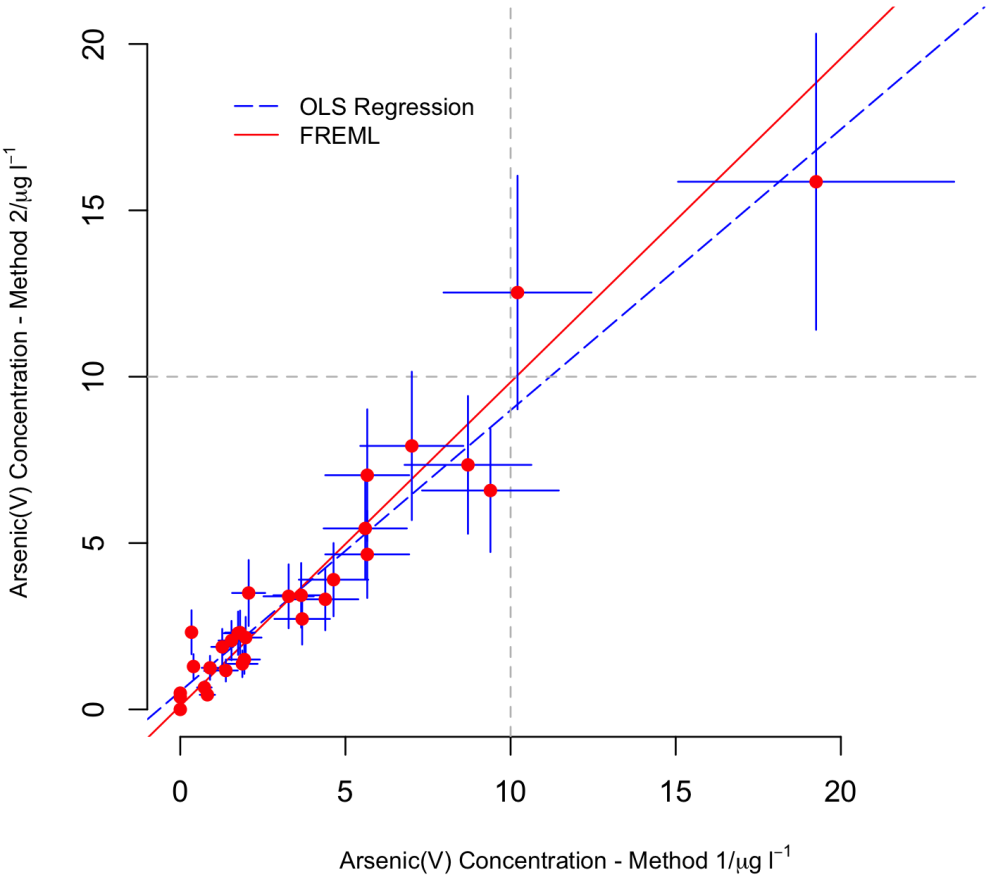


Figure 3.1: Simple Linear regression and FREML models in the process of method comparison in Calibration Metrology.

$$\text{Confidence Interval for Intercept } (a) := \hat{a} \pm \text{coverage factor} \times SE(\hat{a}) \tag{3.2}$$

$$\text{Confidence Interval for Slope } (b) := \hat{b} \pm \text{coverage factor} \times SE(\hat{b}) \tag{3.3}$$

Where, Ripley & Thompson (1987) [8] use 1.96 as the coverage factor assuming that a and b have from normal distributions. According to Eq. (3.3) and Eq. (3.4), variances of $\hat{a}$ and $\hat{b}$ are required and these variances are not immediately available from the FREML calculation. Therefore, we propose the use of parametric bootstrapping to produce $var(\hat{a}), var(\hat{b})$ and also $cov(\hat{a}, \hat{b})$. Hence $var(\hat{a}), var(\hat{b})$ and $cov(\hat{a}, \hat{b})$ are required in both calibration and method comparison perspective.

The next section is about calculating variance-covariance matrix using parametric bootstrapping.

3.2

Parametric Bootstrapping

Parametric bootstrapping is a resampling technique used in statistics to estimate the sampling distribution of a statistic or to make inferences about a population parameter. It is a variation of the bootstrap method, which involves sampling with replacement from the observed data to create new datasets [11]. Parametric bootstrapping is useful when the underlying population distribution can be reasonably assumed to follow a specific parametric form. It allows for making inferences and estimating population parameters even when the underlying distribution is not known or is difficult to determine. An important fact of using parametric bootstrapping is that the validity of the results relies on the accuracy of the chosen parametric model. Four major steps of parametric bootstrapping are: parameter estimation, resampling, calculation of statistic on the bootstrap samples and analysis of resampled statistics. For the calibration problem, the parameter estimation is done using maximum likelihood function and the resulting parameter estimates are used as set of initial values to initiate the resampling procedure. An algorithm shows the procedure followed for to obtain the bootstrap sampling distributions.

Algorithm 1 Parametric bootstrapping procedure for generating empirical sampling distributions of parameters

1: **Inputs:** $\hat{a}$:= Point estimate of intercept, $\hat{b}$:= point estimate for the slope, $\{\hat{u}_i\}_{i=1}^{n}$:= Vector of point estimates for true $\hat{u}_i$, M := Number of simulation runs

//Simulate x, y data

2: **For** s in $1:M$ with M large, do

3: $x_i{}^{(s)} = \hat{u}_i + \mathcal{N}(0, \sigma_{x_i}^2)$ for $i = 1, \ldots, n$

4: $v_i^{(s)} = \hat{a} + \hat{b}\hat{u}_i$

5: $y_i^{(s)} = v_i^{(s)} + \mathcal{N}(0, \sigma_{y_i}^2)$ for $i = 1, \ldots, n$

6: $\{\hat{a}^{(s)}, \hat{b}^{(s)}, \{\hat{u}_i^{(s)}\}_{i=1}^{n}\} := \operatorname{argmin} -l(a, b, u_1, \ldots, u_n | x_i^{(s)}, \sigma_{xi}^2, y_i^{(s)}, \sigma_{yi}^2)$

7: **End For**

//Output bootstrap sampling distributions for parameters

8: **Output:** $\{\hat{a}^{(s)}, \hat{b}^{(s)}, \{\hat{u}_i^{(s)}\}_{i=1}^{n}\}_{s=1}^{M}$

In this study, M = 5000 simulation runs were conducted on the inputs $\hat{a}, \hat{b}$ and $\hat{u}_i$'s. In each trial, a new sample data set was generated that has different plausible set of errors but the same characteristics as the parent data set. For each sample data set, maximum likelihood estimation

Table 3.2: Variance covariance matrix of the parameter estimates of the illustrative example addressed by Ripley & Thompson (1987) [8].

	$\hat{\mathbf{a}}$	$\hat{\mathbf{b}}$
$\hat{\mathbf{a}}$	0.00232	-0.00071
$\hat{\mathbf{b}}$	-0.00071	0.00610

Table 3.3: Comparison of the standard errors of the regression coefficients of the illustrative example evaluated by Parametric bootstrapping and by Ripley & Thompson (1987) [8].

	SE($\hat{a}$)	**SE($\hat{b}$)**
Parametric Bootstrap	0.0482	0.0781
Ripley & Thomson (1987)	0.048	0.076

was then applied contemporary with the generation of the data sets. Hence, the final outcome of the bootstrap procedure was 5000 of possible values for the parameters a, b and $\{u_i\}_{i=1}^{n}$. Figure 3.2 shows the empirical distributions of estimated coefficients $\hat{a}$ and $\hat{b}$ by the FREML model for the illustrative example addressed by Ripley & Thompson (1987) [8]. Table 3.2 shows the spread of these sampling distributions equivalently the variance of the estimates and Table 3.3 compares the results obtained by parametric bootstrap to findings of Ripley & Thompson (1987) [8]. It is evident that, bootstrapping evaluates almost exact same results as Ripley & Thompson have obtained in their analysis (The methodology used to evaluate the $var(\hat{a})$ and $var(\hat{b})$ is not revealed in Ripley & Thompson (1987) paper [8]).

3.3

Method Validation

3.3.1

Coverage Estimation

For the validation assessment, this study uses coverage estimation process. Coverage estimation refers to the process of assessing how well a statistical model's predictions or confidence intervals match the actual outcomes or true values in a dataset [12]. In the context of calibration, coverage estimation pertains to the assessment of how effectively the calculated confidence intervals for the model's parameters encompass the actual true values. FREML, MLE and bootstrap are utilized to provide confidence intervals for the $n + 2$ parameters of the study. Coverage estimation involves in examining whether these confidence intervals encompass the true values of these parameters. If the coverage is close to the nominal confidence level, i.e. 95%, it indicates that the method accurately quantifies the uncertainty associated with the estimated parameters.

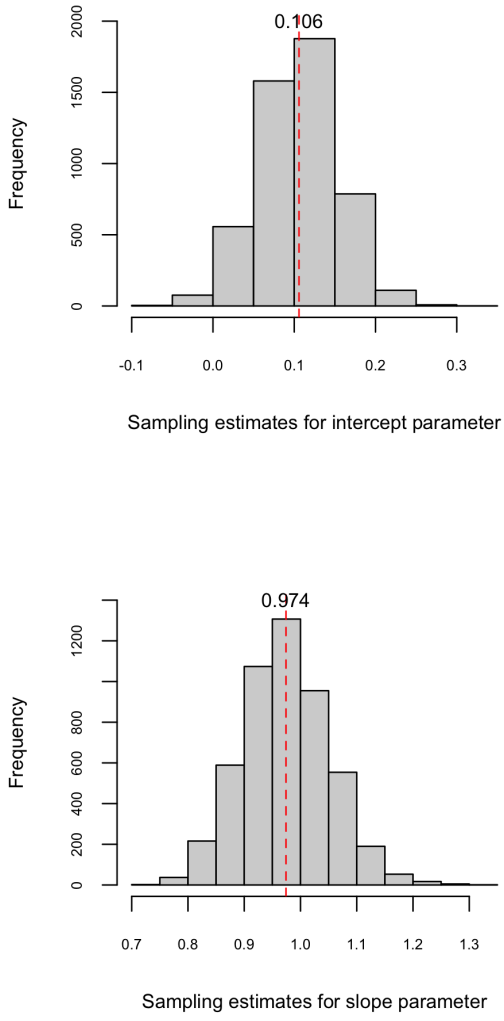


Figure 3.2: Sampling distributions of parameters by parametric bootstrapping.

For the calibration problem the coverage estimation is of the form:

$$\hat{p} = \frac{Number\ of\ Confidence\ Intervals\ which\ include\ the\ population\ parameter}{Total\ number\ of\ confidence\ intervals} \tag{3.4}$$

3.3.2

Validation of implication of Parametric Bootstrapping to FREML and MLE

For the validation of the concept of applying bootstrapping to the FREML model, this study uses 19 lists of datasets where as each list contains 2000 datasets with same number of parameters, same initial values for $\hat{a}$ and $\hat{b}$ but different $\hat{u}_i$ values. We selected these 19 cases to illustrate the efficacy of our approach for different number of parameters and with varying relative sizes between parameters. Table 3.4 shows the initial true parameter values used for generating the sampling data sets. n is the number of observations per data set, a is the intercept parameter and b is the slope parameter, $u.min$ and $u.max$ are the minimum and maximum values of actual u_i's of that particular data list, In each data set, the uncertainty of x and y are subject to change. Relative sd(x) is the expected relative standard deviation of x is allowed to vary by $\pm rsd.x.r$ which refers to range of relative standard deviation (x). Also, y variable measurement uncertainty is generated by using the relative sd(y) and the $rsd.y.r$ or range of relative standard deviation of (x).

For each case depicted in Table 3.4, a total of 2000 sample datasets were simulated. Applying parametric bootstrapping to each dataset results in a single list containing 2000 elements, with each element comprising 5000 bootstrap datasets. Subsequently, applying maximum likelihood estimation (MLE) to each list element yields 5000 estimates for each of the parameters a, b, and $\{u_i\}_{i=1}^{n}$. Next, the 2.5% and 97.5% quantiles are calculated for each of these 2000 sampling distributions. The proportion of confidence intervals containing their true population parameters is then calculated list-wise. This constitutes an estimate of empirical statistical coverage, which should be close to 0.95 if the parametric bootstrap method is consistent. To ensure this, interval estimates of the proportions are calculated [2].

Table 3.5 and Figure 3.3 illustrate the coverage results for Study Case number eight out of the 19 cases tested for statistical validation. "$\hat{p}$" is an estimate of the coverage of the parametric bootstrap procedure. "n" is the number of vectors of $\hat{p}$'s per list. "$SE(\hat{p})$" is the standard error of $\hat{p}$ and "lb" and "ub" represent the lower bound and the upper bound of the estimated $\hat{p}$. According to Table 3.5, all the confidence interval for the proportions calculated include 95%. Therefore, it can be concluded that 95% calculated confidence intervals for the parameters include their respective

Table 3.4: Master parameter set used for data generation.

case	n	a	b	u.min	u.max	rsd(x)	rsd.x.r	rsd(y)	rsd.y.r
1	5	0	1	0.1	5	0.05	0.02	0.15	0.03
2	5	0	1	0.1	5	0.15	0.03	0.05	0.02
3	5	0	1	0.1	5	1.00E-07	0	0.1	0
4	5	10	0.5	0.1	5	0.01	0.005	0.05	0.02
5	5	10	0.5	0.1	5	0.05	0.02	0.15	0.03
6	10	0	1	0.01	50	0.05	0.02	0.15	0.03
7	10	0	1	0.01	50	0.15	0.03	0.05	0.02
8	10	0	1	0.01	50	1.00E-07	0	0.1	0
9	10	10	0.5	0.01	50	0.01	0.005	0.05	0.02
10	10	10	0.5	0.01	50	0.05	0.02	0.15	0.03
11	20	0.001	1.00E-07	0.1	10000	0.05	0.01	0.12	0.06
12	20	0.001	1.00E-07	0.1	10000	0.15	0.1	0.2	0.06
13	20	1.00E+20	1.00E-07	0.1	10000	0.15	0.1	0.2	0.06
14	40	0	1	0.1	5	0.05	0.02	0.15	0.03
15	40	0	1	0.1	5	0.15	0.03	0.05	0.02
16	40	0	1	0.1	5	1.00E-07	0	0.1	0
17	100	0	1	0.1	5	0.05	0.02	0.15	0.03
18	100	0	1	0.1	5	0.15	0.03	0.05	0.02
19	100	0	1	0.1	5	1.00E-07	0	0.1	0

population parameters.

Table 3.6 summarizes the coverage results of the 19 cases studied. In each case, the number of parameters studied equals the total number of observations plus 2. According to Table 3.6, there are 593 total coverage assessments and 25 of them haven't captured by the expected coverage of 95%. Moreover, among those 25 coverage assessments, for 11 cases coverage was over achieved and for the remaining 14 cases the coverage was under achieved. Therefore, is can be concluded that parametric bootstrapping application in calibration and in method comparison problems give highly accurate results.

Table 3.5: Proportional Comparison of Estimated VS. True Parameters Case 8 with 10 Parameters.

	p.hat	SE.p.hat	lb	ub
a	0.942	0.00525	0.931	0.952
b	0.95	0.00487	0.940	0.960
u1	0.946	0.00505	0.936	0.956
u2	0.948	0.00496	0.938	0.958
u3	0.946	0.00505	0.936	0.956
u4	0.942	0.00525	0.931	0.952
u5	0.953	0.00476	0.943	0.962
u6	0.954	0.00468	0.945	0.963
u7	0.948	0.00496	0.938	0.958
u8	0.951	0.00483	0.942	0.961
u9	0.949	0.00494	0.939	0.958
u10	0.954	0.00468	0.945	0.963

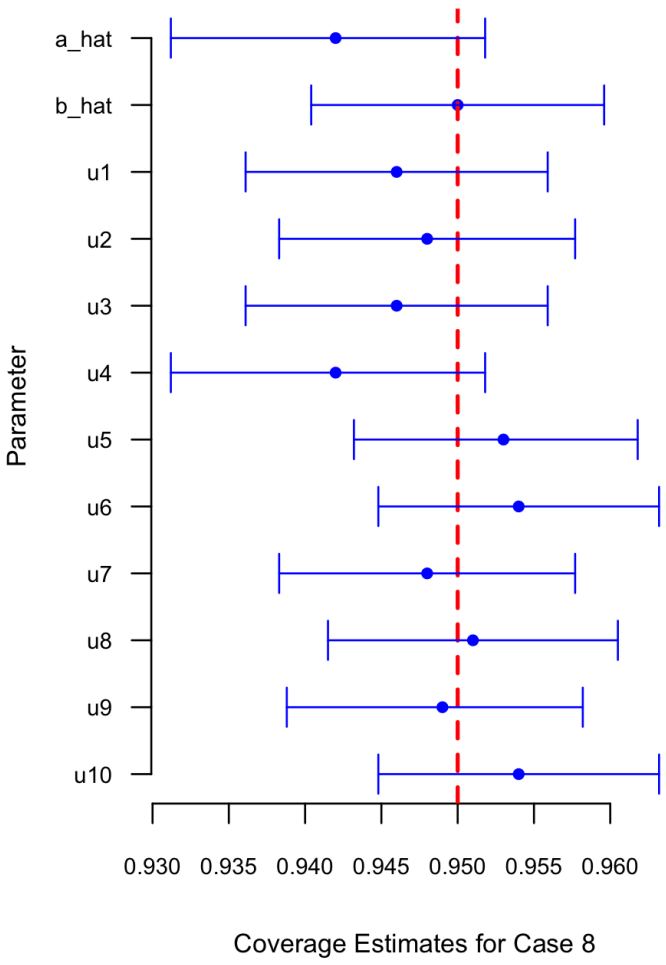


Figure 3.3: Estimated Confidence Intervals for Coverage Case 8 with 10 Parameters.

Table 3.6: Summary of coverage estimation that does not capture the target coverage of 95%.

Case number	**Parameters coverage not achieved**	**Observations per data set**
1	**None**	5
2	**None**	5
3	**None**	5
4	**u4**	5
5	**u4**	5
6	**u6**	10
7	**None**	10
8	**None**	10
9	**u5**	10
10	**u1**	10
11	**None**	20
12	**a** **b** **u5**	20
13	**None**	20
14	**a** **u8** **u40**	40
15	**None**	40
16	**u4** **u13** **u21**	40
17	**u3** **u34** **u41** **u69**	100
18	**u73**	100
19	**u10** **u15** **u36** **u52** **u78** **u89**	100
Total number of parameters which the coverage achieved 95%		593
Parameters for which the coverage not achieved 95%	Total number of parameters coverage under achieved	14
	Total number of parameters coverage over achieved	11

3.4 Conclusion

In this chapter, we illustrated the FREML model application in method comparison in calibration and then it was found that in both calibration and method comparison problems variance-covariance matrix of model parameters need to be estimated. Since FREML fails to generate the variance-covariance matrix, parametric bootstrap method was introduced to generate the $var(\hat{a})$, $var(\hat{b})$ and the $cov(\hat{a}, \hat{b})$. The parametric bootstrap implication in FREML was validated using manually generated data sets and using the concept of "coverage estimates" and found that uncertainties calculated with the aid of bootstrap technique reaches 95% significance level. Therefore, it can be concluded that the parametric bootstrap implication in FREML evaluates the uncertainties of the model coefficients with significant accuracy. Chapter 4 is focused on modeling and calculating the model uncertainties when the initial input variable uncertainties are unknown.

CHAPTER 4
EXTENSION OF THE FREML MODEL AND MAXIMUM-LIKELIHOOD ESTIMATION PROCEDURE WHEN MEASUREMENT UNCERTAINTIES ARE ESTIMATED AND REPORTED WITH A STATED DEGREES-OF-FREEDOM

4.1
Introduction

In the preceding section, we delved into the utilization of the parametric bootstrap technique in scenarios where the FREML model is employed for both straight line calibration and method comparison problems. In both contexts, the estimation of model coefficients was predicated upon the assumption that the population variances of variables x and y are known and non negligible. In this chapter, we shift focus to an extension of the model in situations where the population variances of the variables are unknown and instead replaced by their respective sample variances along with their corresponding degrees-of-freedom. We construct the joint likelihood function to provide accurate estimates for the unknown parameters and extend the parametric bootstrap process to the resulting model. We end with a comprehensive study on the degrees-of freedom space to discover when the extended FREML approach is necessary, or when degrees-of-freedoms are large enough so that FREML is safe to use.

4.2
Likelihood Function for the Extended Model

Ripley & Thompson (1987) suggest that FREML model is more appropriate in straight line calibration problem when x and y have known population uncertainties [8]. Therefore, for this study we uses FREML approach along with maximum likelihood estimation process to determine model coefficients of the linear calibration curve. The scenario involves several unknown parameters that need to be estimated through the analysis process. The linear model coefficients, a and b, true u_i values, population uncertainty of x namely $\{\sigma_{x_1}, \sigma_{x_2}, \ldots, \sigma_{x_n}\}$ and population uncertainties of y

namely $\{\sigma_{y_1}, \sigma_{y_2}, \ldots, \sigma_{y_n}\}$ are now viewed as unknown parameters that need to be estimated given data. All of these unknown parameters are estimated using an extension of FREML and MLE approach. This entails finding the values of a, b and $\{u_i, \sigma_{x_i},\ \sigma_{y_i}\}_{i=1}^{n}$ that minimize the negative log likelihood of observing the given data points while considering the uncertainties associated with both x and y.

As per the structure of FREML:

$$v_i = a + bu_i$$

where v is the true responses, u is the true x_i values and a and b are the estimates for intercept and slope parameters. Also,

$$x_i = u_i + \epsilon_i$$
$$y_i = v_i + \eta_i$$

where, ϵ_i is the deviation of true u_i from observed x_i and η_i is the deviation of true v_i from observed y_i. The assumptions of the FREML model for the extended case are

$$\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{y_i}^2)$$
$$\eta_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_{x_i}^2)$$
$$\epsilon_i \text{ and } \eta_i \text{ are independent.}$$

But now population variances $\{\sigma_{x_1}^2, \sigma_{y_i}^2\}$ are unknown but estimated and their estimates are reported in the data, specially we assume $\{s_{x_i}^2, df_{x_i}, s_{x_i}^2, df_{y_i}\}$ are given. To extend the model and UQ procedure, we will exploit the following well known distribution assumptions[2].

$$\frac{s_{y_i}^2 \times df_{y_i}}{\sigma_{y_i}^2} \sim \chi_{df_{y_i}}^2$$
$$\frac{s_{x_i}^2 \times df_{x_i}}{\sigma_{x_i}^2} \sim \chi_{df_{x_i}}^2$$

We also use the fact that $x_i \perp\!\!\!\perp s_{x_i}^2$ and $y_i \perp\!\!\!\perp s_{y_i}^2$,[2] which are reasonable assumptions in this scenario. Therefore, the likelihood equation for the extended version of the calibration problem can

be written as:

$$L(a, b, \boldsymbol{u}, \boldsymbol{\sigma_x^2}, \boldsymbol{\sigma_y^2} | \boldsymbol{x}, \boldsymbol{s_x^2}, \boldsymbol{df_x}, \boldsymbol{y}, \boldsymbol{s_y^2}, \boldsymbol{df_y}) := f(\boldsymbol{x}, \boldsymbol{s_x^2}, \boldsymbol{df_x}, \boldsymbol{y}, \boldsymbol{s_y^2}, \boldsymbol{df_y} | a, b, \boldsymbol{u}, \boldsymbol{\sigma_x^2}, \boldsymbol{\sigma_y^2}) \tag{4.1}$$

$$= \prod_{i=1}^{n} f(x_i | u_i, \sigma_{x_i}^2) f(s_{x_i}^2 | \sigma_{x_i}^2, df_{x_i}) f(y_i | a, b, u_i, \sigma_{y_i}^2) f(s_{y_i}^2 | \sigma_{y_i}^2, df_{y_i}) \tag{4.2}$$

4.2.1

Maximum Likelihood Estimates for Population Variances

The likelihood function in Eq. (4.1) shows that one can use a hierarchical approach for estimating the model parameters. Therefore, first we will estimate the unknown variances of x and y and then we will substitute these estimates into the likelihood function to estimate the $a, b,$ and $\boldsymbol{u}$. In this case we assume that the sample variance is a function of chi-square distributed and the sample variance only depends upon the population variance and therefore independent of sample mean. Therefore, the joint distribution function is the product of individual probability distributions Eq. (4.2).

$$\{\hat{\sigma}_{x_i,ML}^2, \hat{\sigma}_{y_i,ML}^2\} = \text{argmin}\{L(\sigma_x^2 | s_x^2, df_x) \times L(\sigma_y^2 | s_y^2, df_y)\} \tag{4.3}$$

$$= \text{argmin}\{-\log \prod_{i=1}^{n} f(s_{x_i}^2 | \sigma_{x_i}^2, df_{x_i}) \times f(s_{y_i}^2 | \sigma_{y_i}^2, df_{y_i})\} \tag{4.4}$$

The next task will be to find the point estimates for the population variances. From the previous section we know that $s^2 \times df/\sigma^2 \sim {\chi^2}_{df}$. Therefore, this formulas is used to derive the probability distribution function of s^2 using transformations. The formula in Eq. (4.5) shows the transformed formula to calculate the probability distribution function of s^2.

$$\text{Let } W = \frac{s^2 \times df}{\sigma^2} \text{ where, } W \sim \chi_{df}^2$$

$$\begin{aligned} F_{S^2}\left(s^2\right) &= P\left(S^2 \leq s^2\right) \\ &= P\left(\frac{W\sigma 2}{df} \leq s^2\right) \\ &= P\left(W \leq \frac{s^2 df}{\sigma^2}\right) \\ &= F_W\left(\frac{s^2 df}{\sigma^2}\right) \end{aligned}$$

Therefore,

$$\begin{aligned} f_{S^2}\left(s^2\right) =& \frac{d}{ds^2}\left(F_W\left(\frac{s^2 df}{\sigma^2}\right)\right) \\ =& f_W\left(\frac{s^2 df}{\sigma^2}\right) \times \frac{d}{ds^2}\left(\frac{s^2 df}{\sigma^2}\right) \\ \therefore f_{S^2}\left(s^2\right) =& \frac{df}{\sigma^2} f_W\left(\frac{s^2 df}{\sigma^2}\right) \end{aligned} \tag{4.5}$$

According to above formula,

$$\begin{aligned} f_{S^2}(s^2) =& \frac{df}{\sigma^2}\left[\frac{1}{2^{df/2}\Gamma(df/2)}\left[\frac{s^2 df}{\sigma^2}\right]^{\frac{df}{2}-1} e^{-\frac{s^2 df}{2\sigma^2}}\right] \\ \propto& \frac{df}{\sigma^2}\left[\frac{s^2 df}{\sigma^2}\right]^{\frac{df}{2}-1} e^{-\frac{s^2 df}{2\sigma^2}} \\ -logL(\sigma^2|s^2) =& -\log\left(\frac{df}{\sigma^2}\right) - \left(\frac{df}{2}-1\right)\log\left(\frac{s^2 df}{\sigma^2}\right) - \frac{s^2 df}{s\sigma^2} \\ \frac{d}{d\sigma^2} - \log L(\sigma^2|s^2) =& \frac{1}{\sigma^2} - \left(\frac{df}{2}-1\right)\left[-\frac{1}{\sigma^2}\right] - \frac{s^2 df}{2\sigma^4} \end{aligned}$$

To find the critical points, let $\frac{d}{d\sigma^2} - \log L(\sigma^2|s^2) = 0$

$$\implies \hat{\sigma^2}_{MLE} = s^2$$

The second derivative test is used to check whether the negative likelihood function reach a minimum when $\hat{\sigma}^2 = s^2$.

$$\begin{aligned} \frac{\partial^2(-\log L(\sigma^2|s^2))}{\partial {\sigma^2}^2} =& \frac{\partial[\frac{1}{\sigma^2} - (\frac{df}{2}-1)[-\frac{1}{\sigma^2}] - \frac{s^2 df}{2\sigma^4}]}{\partial\sigma^2} \\ =& \frac{-df/2}{\sigma^4} + \frac{s^2 df}{\sigma^6} \end{aligned} \tag{4.6}$$

when $\hat{\sigma^2} = s^2$,

$$\begin{aligned} \frac{\partial^2(-\log L(\sigma^2|s^2))}{\partial {\sigma^2}^2} =& \frac{df}{s^4} - \frac{df}{2s^4} \\ =& \frac{df}{2s^4} > 0 \end{aligned} \tag{4.7}$$

Therefore, $-\log\{L(\sigma^2|s^2)\}$ reaches minimum when $\sigma^2 = s^2$. Since the maximum likelihood estimator for population variance is the sample variance, we can get rid of the optimizing the likelihood formula shown in Eq. (4.1) and can directly use sample variance. Now the original

likelihood function for the extended model can be revised as:

$$
\begin{aligned}
-\log L(a,b,\boldsymbol{u}|\boldsymbol{x},\hat{\boldsymbol{\sigma}}=\boldsymbol{s_x^2},\boldsymbol{y},\hat{\boldsymbol{\sigma}}=\boldsymbol{s_y^2}) &:= -\log\{f(\boldsymbol{x},\boldsymbol{s_x^2},\boldsymbol{y},\boldsymbol{s_y^2}|a,b,\boldsymbol{u})\} \\
&= -\log\left\{\prod_{i=1}^{n} f(x_i|u_i,\hat{\sigma}^2{}_{x_i,MLE}) f(y_i|a,b,u_i,\hat{\sigma}^2{}_{y_i,MLE})\right\}
\end{aligned} \tag{4.8}
$$

Then we need to minimize the negative log likelihood function shown is Eq. (4.8) to obtain the parameter estimates. We will be using the likelihood function in Eq. (4.8) which can produce estimates for $n+2$ parameters (a, b and $\{\mathbf{u}\}$). The function is optimized via the R function optim and the method Nelder-Mead. The estimates that minimizes the negative log-likelihood function will be using as the initial values for the bootstrapping. We will be using parametric bootstrapping along with numerical maximum likelihood and propagate all reported uncertainties, s_{x_i}, s_{y_i} and df_{x_i}, df_{y_i} through the system of measurement equations aiming to obtain the uncertainties of model coefficient estimates.

$$
\{\hat{a},\hat{b},\hat{u}_{i,LM}|\hat{\sigma}^2_{x_i,ML},\hat{\sigma}^2_{y_i,ML}\} = \text{argmin}\left\{-\log\prod_{i=1}^{n} f(x_i|\hat{\sigma}^2_{x_{i,ML}},u_i)\times f(y_i|\hat{\sigma}^2_{y_{i,ML}},a,b,u_i)\right\} \tag{4.9}
$$

Optimizations are done using R optim function. The outcomes of the likelihood functions are then used to build the bootstrap algorithm to estimate the sampling distributions of a, b and $\boldsymbol{u}$ which we show in Algorithm 2. According to Algorithm 2, it can be seen that for each bootstrap data set, the population variance is substituted using sample variance. And for generating each x_i, random values from normal distribution were used and the substituted population variances are again generated by using chi-square distribution. For a single initial data set, this procedure generates 5000 possible sampling data sets and per each data set maximum likelihood estimates are then calculated using R optim function. Final outcome of the procedure is 5000 possible $\hat{a},\hat{b},\{\hat{u}\}$, yielding Monte Carlo approximations of their sampling distributions.

4.3

Statistical Validation - Extended Approach

For the validation of this procedure, coverage calculations were used. Since the problem consists with both x and y sample uncertainties and the respective degrees-of-freedoms, validation was done considering all possible combinations of degrees of freedoms. In this study, 5 major cases each with 5 possible combinations were considered for coverage estimation. Table 4.1 shows all

Algorithm 2 Parametric bootstrapping procedure for generating empirical sampling distributions of parameters - Extended model approach.

Inputs: $\hat{a}$:= Point estimate of intercept, $\hat{b}$:= point estimate for the slope, $\{\hat{u}_i\}_{i=1}^n$:= Vector of point estimates for true $\hat{u}_i$, $\{\hat{\sigma}_{xi}{}^2\}_{i=1}^n$:= sample variance for x data,$\{df_{x_i}\}_{i=n}^n$:= degrees-of-freedom of x,$\{\hat{\sigma}_{y_i}{}^2\}_{i=1}^n$:= sample variance for y data,$\{df_{y_i}\}_{i=n}^n$:= degrees-of-freedom of y, M := Number of simulation runs

For s in $1:M$ with M large, do

$v_i = \hat{a} + \hat{b}\hat{u}_i$ for $i = 1, \ldots, n$

//Simulate new sample standard deviations for bootstrap data sets

$s_{xi}{}^{2(s)} \sim \chi^2_{df_{x_i}} \frac{\hat{\sigma}_{xi}{}^2{}_{MLE}}{df_{xi}}$ for $i = 1, \ldots, n$

$s_{y_i}{}^{2(s)} \sim \chi^2_{df_{y_i}} \frac{\hat{\sigma}_{y_i}{}^2{}_{MLE}}{df_{y_i}}$ for $i = 1, \ldots, n$

//Account for uncertainty in true $\sigma^2_{x_i}, \sigma^2_{y_i}$ by simulating new possible values given $s^2_{x_i}$ and $s^2_{y_i}$

$\tilde{\sigma_{x_i}}^2 \sim \frac{s^2_{x_i} df_{x_i}}{\chi^2_{df_{x_i}}}$ for $i = 1, \ldots, n$

$\tilde{\sigma_{y_i}}^2 \sim \frac{s^2_{y_i} df_{y_i}}{\chi^2_{df_{y_i}}}$ for $i = 1, \ldots, n$

//Simulate x, y data.

$x_i{}^{(s)} \sim \hat{u}_i + \mathcal{N}(0, \tilde{\sigma_{x_i}}^2)$ for $i = 1, \ldots, n$

$y_i^{(s)} \sim v_i^{(s)} + \mathcal{N}(0, \tilde{\sigma_{y_i}}^2)$ for $i = 1, \ldots, n$

$\{\hat{a}^{(s)}, \hat{b}^{(s)}, \{\hat{u}_i^{(s)}\}_{i=1}^n, \{\hat{\sigma_{x_i}}^{2(s)}\}_{i=1}^n, \{\hat{\sigma_{y_i}}^{2(s)}\}_{i=1}^n\} :=$

argmin $-l(a, b, u_1, \ldots, u_n | x_i^{(s)}, s^2_{x_i}{}^{(s)}, df_{x_i}, y_i^{(s)}, s^2_{y_i}{}^{(s)}, df_{y_i})$

End For

//Output bootstrap sampling distributions for parameters

Output: $\{\hat{a}^{(s)}, \hat{b}^{(s)}, \{\hat{u}_i^{(s)}\}_{i=1}^n\}$

Table 4.1: Case combinations studied in coverage estimation process.

Case Number	Procedure	df_x	df_y
1	Fixed y at low degrees of freedom while changing x degrees of freedom	{5,10,30,50,100}	5
2	Both x and y degrees of freedom vary from smaller to larger	{5,10,30,50,100}	{5,10,30,50,100}
3	Fixed x at low degrees of freedom while changing y degrees of freedom	5	{5,10,30,50,100}
4	Fixed y at low degrees of freedom while changing x degrees of freedom	{5,10,30,50,100}	100
5	Fixed x at high degrees of freedom while changing y degrees of freedom from low to high values.	100	{5,10,30,50,100}

Table 4.2: Comparison of Coverage Assessment with dfx = 5 and dfy = 5 (EFREML vs. FREML Approach).

	Results of Coverage using EFREML				**Results of Coverage using FREML**			
	p.hat	**SE.p.hat**	**lb**	**ub**	**p.hat**	**SE.p.hat**	**lb**	**ub**
a.hat	0.963	0.00425	0.954	0.971	0.897	0.0068	0.884	0.91
b.hat	0.959	0.00443	0.95	0.968	0.848	0.00803	0.832	0.864
u1	0.947	0.00501	0.937	0.957	0.886	0.00711	0.872	0.9
u2	0.951	0.00483	0.942	0.96	0.889	0.00701	0.876	0.903
u3	0.955	0.00464	0.946	0.964	0.883	0.00719	0.869	0.897
u4	0.956	0.00459	0.947	0.965	0.876	0.00736	0.862	0.891
u5	0.956	0.00461	0.946	0.965	0.884	0.00716	0.87	0.898
u6	0.953	0.00473	0.944	0.962	0.882	0.00721	0.868	0.896
u7	0.943	0.00518	0.933	0.953	0.878	0.00731	0.864	0.893
u8	0.955	0.00464	0.946	0.964	0.883	0.00719	0.869	0.897
u9	0.959	0.00446	0.95	0.967	0.882	0.0072	0.868	0.897
u10	0.947	0.00503	0.937	0.956	0.877	0.00733	0.863	0.892

possible case combinations studied for the coverage estimations. For each of these cases, 2000 data sets were generated using the same true parameter values and at the end of the bootstrap procedure, sampling distributions for each model parameter estimated each with 5000 estimates were obtained. Assuming binomial distribution, coverage intervals were calculated for each of these data set to check whether how many intervals include the true population parameter with significance level of 0.05. For each case we end up with $n + 2$ coverage intervals and check whether the interval includes 0.95.

For each of the cases presented in Table 4.1, coverage calculations were completed. Table 4.2 and Table 4.3 depict two of these outputs generated using the EFREML approach, as well as using the FREML approach. Upon comparing both FREML and extended FREML outputs it can be seen that EFREML catch the true parameters as expected and reach 0.95 coverage all most all of the time compared to FREML outputs. These results suggest that even with smaller degrees-of-freedom, employing the extended approach alongside the FREML model leads to calculated confidence intervals for most parameters capturing the true population parameter 95% of the time.

Table 4.3: Comparison of Coverage Assessment with dfx = 100 and dfy = 5 (EFREML vs. FREML Approach).

	Results of Coverage using EFREML				**Results of Coverage using FREML**			
	p.hat	**SE.p.hat**	**lb**	**ub**	**p.hat**	**SE.p.hat**	**lb**	**ub**
a.hat	0.956	0.0046	0.946	0.965	0.903	0.0066	0.891	0.916
b.hat	0.948	0.0050	0.938	0.957	0.847	0.0081	0.831	0.863
u1	0.949	0.0049	0.939	0.958	0.947	0.0050	0.937	0.956
u2	0.959	0.0044	0.95	0.968	0.937	0.0054	0.926	0.948
u3	0.959	0.0045	0.95	0.967	0.933	0.0056	0.923	0.944
u4	0.96	0.0044	0.951	0.968	0.928	0.0058	0.917	0.94
u5	0.964	0.0042	0.956	0.972	0.941	0.0053	0.931	0.951
u6	0.952	0.0048	0.943	0.961	0.922	0.0060	0.91	0.934
u7	0.953	0.0047	0.944	0.962	0.932	0.0056	0.921	0.943
u8	0.946	0.0051	0.936	0.955	0.918	0.0061	0.907	0.93
u9	0.958	0.0045	0.949	0.967	0.937	0.0054	0.926	0.948
u10	0.944	0.0051	0.934	0.954	0.928	0.0058	0.917	0.939

Except for the coverage assessment, we have graphed the variation of the coverage estimation over the degrees of freedoms of x and y to see how well the previous and current procedure capture the true parameter values over the sample size. For each of the cases mentioned in Table 4.1, $var(\hat{a}), var(\hat{b})$ and the $cov(\hat{a}, \hat{b})$ are drawn against the degrees-of-freedom (Figure 4.1 - Figure 4.5)

4.3.1

When should degrees-of-freedom be accounted for as an uncertainty source

Our final effort is to generate and compare the sampling distributions for $\hat{a}, \hat{b}$ using original FREML model assuming the given uncertainty is equivalent to the population uncertainty and also using the extended approach assuming the given uncertainties are the sample uncertainties. Primary goal of doing this is to check whether there is a significant difference between the outputs of the two methods and if so that difference will be affecting the variance of predicted new unknown compound concentration. For each case mentioned in Table 4.1, we calculate the following ratios and compare between the degrees of freedoms. These ratios quantify the percentage that model parameter uncertainties will be underestimated if degrees- of-freedom is ignored as an uncertainty source.

$$\text{Factor } var(\hat{a}) \text{ underestimated} = \frac{var(\hat{a}|FREML)}{var(\hat{a}|\text{Extended FREML})}$$

$$\text{Factor } var(\hat{b}) \text{ underestimated} = \frac{var(\hat{b}|FREML)}{var(\hat{b}|\text{Extended FREML})}$$

$$\text{Factor } cov(\hat{a}, \hat{b}) \text{ underestimated} = \frac{cov(\hat{a}, \hat{b}|FREML)}{cov(\hat{a}, \hat{b}|\text{Extended FREML})}$$

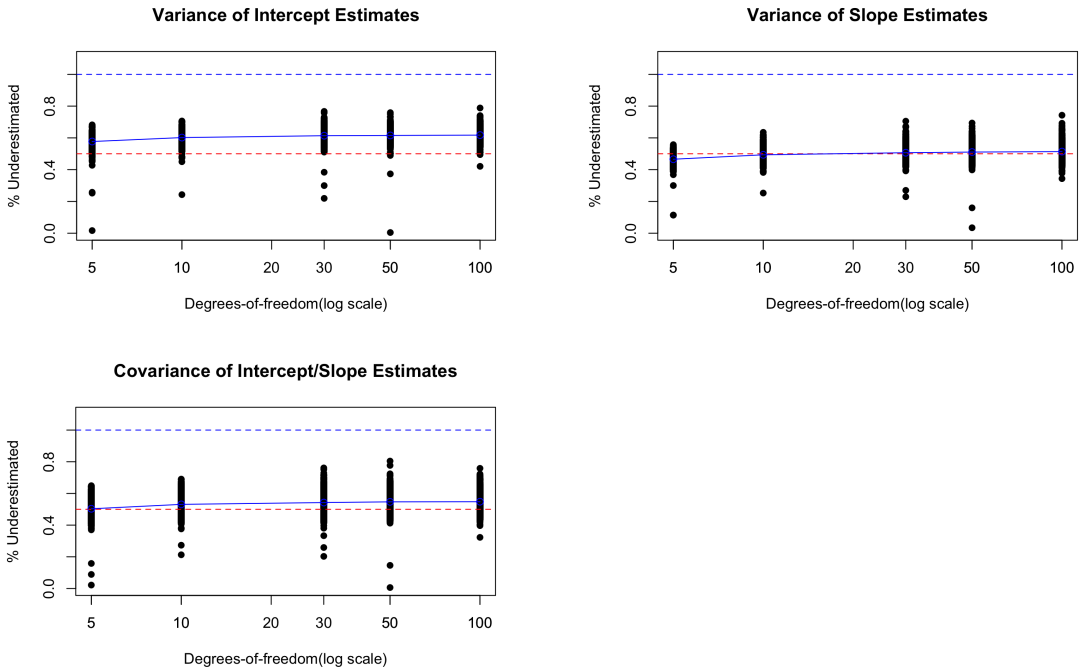


Figure 4.1: FREML Vs. Extended FREML estimated variance covariance comparison - Case 1.

Figure 4.1 - Figure 4.5 show the output for this calculation and according to that the uncertainties calculated for $\hat{a}, \hat{b}$ using EFREML is two times higher than the uncertainties calculated from the original FREML model. Therefore, it can be concluded that taking the degrees of freedom and the sample uncertainties into account when the experiment is done using small sample sizes is important for determining the final outcome of the calibration process.

In Case 1 (Figure 4.1), when the degrees-of-freedom of $\{x, y\}$ is {5,5}, the percentage underestimated is 0.5. which means, the variance calculated using EFREML is two times than the variance calculated using FREML model. When the x degrees-of-freedom increases, underestimation seemingly getting reduced for the intercept parameter but can not see that deviation in the other two graphs indicating that FREML underestimates the uncertainty when the y degrees of freedom is smaller. According to Case 2 shown in Figure 4.2, when the x and y degrees-of-freedom changes sequentially from lower to higher, it can be seen that in the the underestimated ratio is about 0.5 when $\{df_x, df_y\}$ is {5,5} and eventually the underestimation declined with the number of replicates increases. That means when we have larger degrees-of-freedom for both x and y, both FREML and EFREML give similar results. Figure 4.3 shows the underestimation of estimated variance for slope intercept and covariance between slope and intercept when the df_x is 5 and held constant and the df_y takes values from 5 to 100. For the intercept, underestimation ratio reaches 0.5 when both

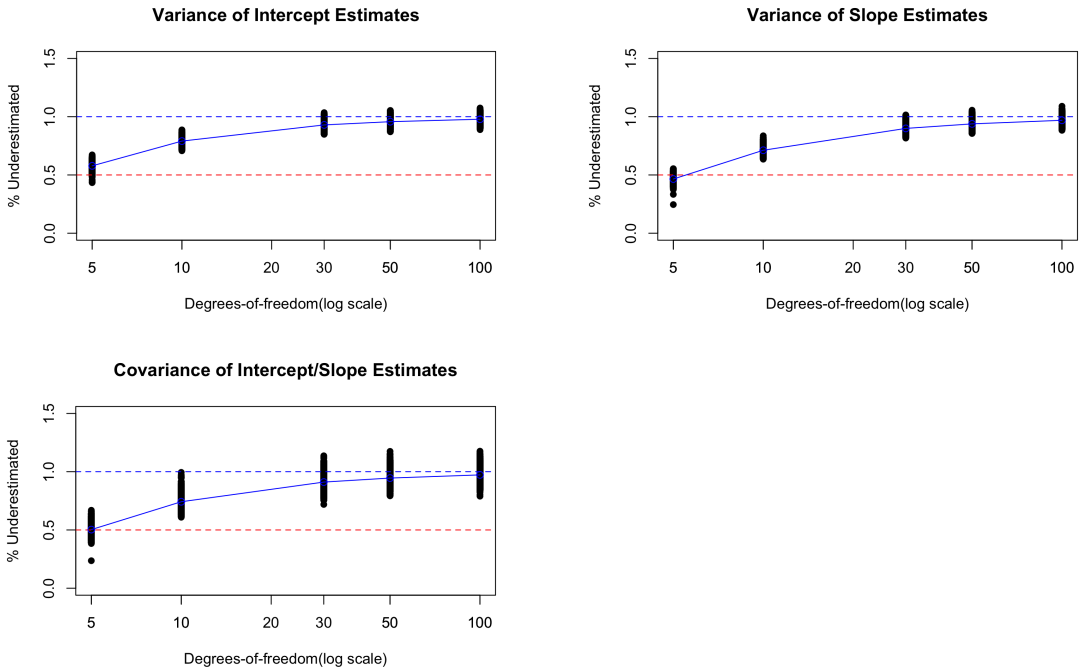


Figure 4.2: FREML Vs. Extended FREML estimated variance covariance comparison - Case 2.

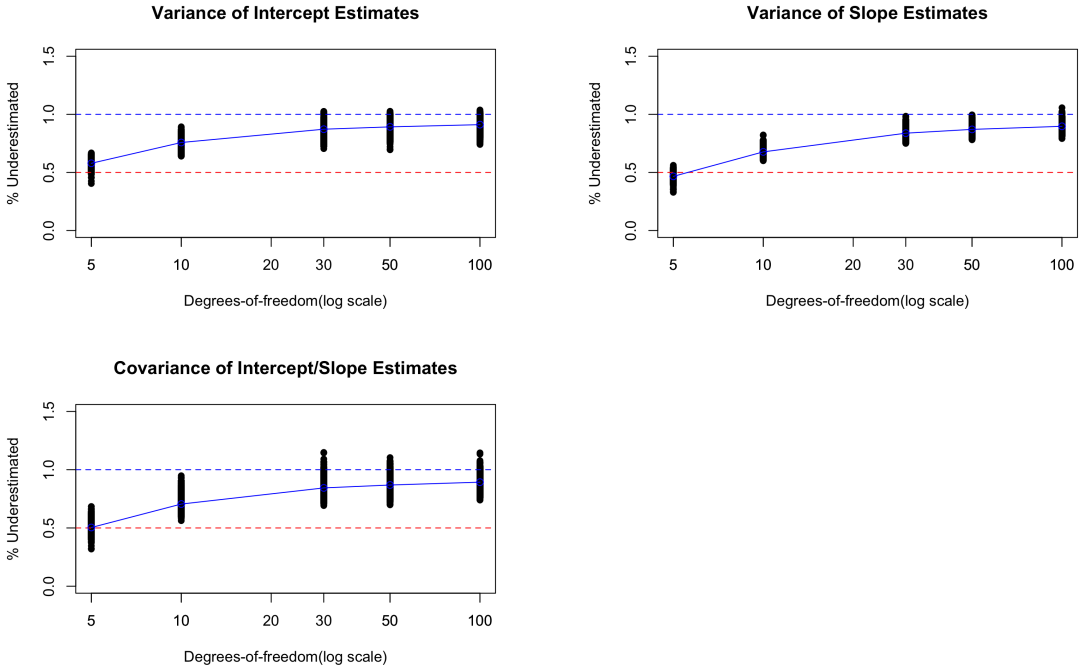


Figure 4.3: FREML Vs. Extended FREML estimated variance covariance comparison - Case 3.

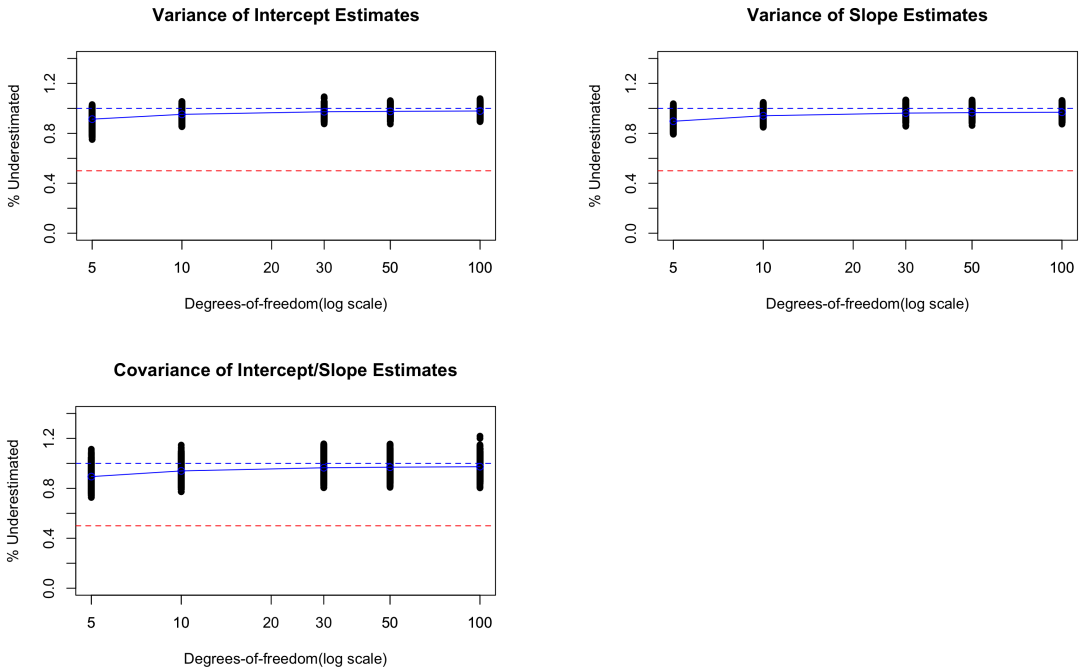


Figure 4.4: FREML Vs. Extended FREML estimated variance covariance comparison - Case 4.

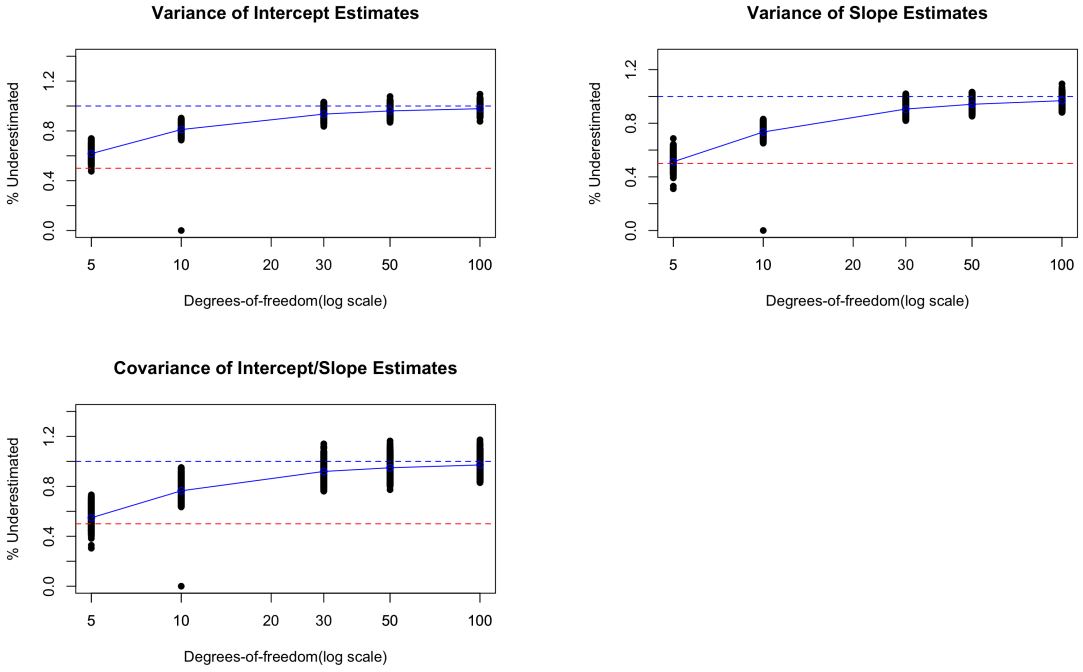


Figure 4.5: FREML Vs. Extended FREML estimated variance covariance comparison - Case 5.

degrees-of-freedoms are smaller and the ratio held consistent when from $df_x = 10$ to 30 and thereafter the ratio get closer to 1. The remaining two plots show the same pattern; ratio is about 0.5 when both degrees-of-freedoms equal to 5 and eventually increases when the df_y gets bigger. Figure 4.4 shows variance covariance comparison results for Case 4 where the the df_x changes between 5 and 100 while the df_y is consistent at 100. According to the results, the underestimation ration is closer to 1 all the time with few miner deviations indicating that FREML and EFREML approach gives similar results when the df_y is a higher value. Figure 4.5 show the behavior of the underestimation ratio when the df_x is held constant at 100 and df_y changes between 5 to 100. It can be seen that the underestimation ratio is almost 0.5 when the df_y is 5 and getting reduced when the df_y increases. Therefore, it can be concluded that the FREML underestimates the variance when either of the df_x or df_y are lower. Therefore using EFREML gives reasonable estimates when the number of replicates are low and the population variances are unknown.

4.4

Conclusion

Calibration problem is usually carried out using small number of replicates and the reported uncertainties can not be considered as population variances. Therefore, the extended approach is to evaluate the uncertainty of the model coefficients when the population variances are unknown. In this process we are propagating both sample variances and the degrees-of-freedoms into the likelihood function. This study derives a mathematical formula for new likelihood function and provides the maximum likelihood estimation of the population variance using reported data. One of the most important finding of the mathematical exploration was finding of the maximum likelihood estimate of the population variance as the sample variance itself which is an unbiased estimator for this problem. Based on the coverage calculations, it can be seen that all most all the calculated coverage intervals capture 0.95 indicating the extended procedure accurately predict the uncertainty of the model coefficients. Moreover, variance-covariance comparison between FREML and EFREML shows that the extended method has higher variances and covariances compared to original method indicating that extended solutions are more accurate when the number of replicates are smaller. Ultimately, the EFREML model is shown to be necessary when $df_x, df_y \leq 30$ to ensure that $var(\hat{a}), var(\hat{b})$ and $cov(\hat{a}, \hat{b})$ are not significantly underestimated.

CHAPTER 5

AN IN−DEPTH DESCRIPTION OF PROGRAMMING CONSTRUCTS FOR THE STUDY

Calibration issue occurs in metrology calibration problem when fitting a linear calibration curve to detect an unknown chemical compound concentration. This challenge can manifest across various sectors, including healthcare, environmental monitoring, manufacturing, and scientific research [5]. This study primarily revolves around leveraging numerical methods to optimize the likelihood function and employing the bootstrap method to generate sampling distributions of model coefficients. Consequently, the study necessitates the development of three key R functions: one for constructing the likelihood function (`nLogLik_FREML_knownPopVar`), another for optimizing the likelihood function (`mmEst_FREML_knownPopVar`) and obtaining estimates, and a third for bootstrapping data generation (`pbs_FREML_knownPopVar`). This investigation focuses on two scenarios: firstly, estimating the uncertainties associated with model coefficients when the error population variances are known, and secondly, when they are assumed to be unknown and replaced by sample variances along with the respective degrees of freedom. Initially, we constructed the aforementioned functions and subsequently modified them to cater to the second scenario. All programming tasks were executed utilizing the R statistical software environment. These functions are designed to handle inputs in the form of data frames. It is essential that all input data be numeric and devoid of categorical or Boolean data. At the outset of each script, we elucidate the functions employed for the FREML and EFREML procedures. Additionally, all parameter and variable definitions are written within the code.

```
RT_data <- list(data = data.frame(x, sd_x_pop = sx, y, sd_y_pop = sy))
print{RT_data}


## a) MLE estimates for original data set
ml_est <- mmEst_FREML_knownPopVar(RT_data)
ml_est$coefficients
```

x	sx	y	sy
8.71	1.92	7.35	2.07
7.01	1.56	7.92	2.23
3.28	0.76	3.4	0.96
5.6	1.26	5.44	1.53
1.55	0.39	2.07	0.59
1.75	0.43	2.29	0.65
0.73	0.22	0.66	0.19
3.66	0.84	3.43	0.97
0.9	0.25	1.25	0.36
9.39	2.07	6.58	1.85
4.39	1	3.31	0.93
3.69	0.84	2.72	0.77
0.34	0.13	2.32	0.66
1.94	0.47	1.5	0.43
2.07	0.5	3.5	0.99
1.38	0.36	1.17	0.33
1.81	0.45	2.31	0.66
1.27	0.33	1.88	0.54
0.82	0.23	0.44	0.13
1.88	0.46	1.37	0.4
5.66	1.27	7.04	1.98
0	0.06	0	0.01
0	0.06	0.49	0.15
0.4	0.15	1.29	0.37
0	0.06	0.37	0.12
1.98	0.48	2.16	0.62
10.21	2.24	12.53	3.51
4.64	1.05	3.9	1.1
5.66	1.27	4.66	1.31
19.25	4.18	15.86	4.45

	$\hat{\mathbf{a}}$	$\hat{\mathbf{b}}$
$\hat{\mathbf{a}}$	0.00232	-0.00071
$\hat{\mathbf{b}}$	-0.00071	0.00610

```
## b) get list of PBS datasets
pbs_data_list <- pbs_FREML_knownPopVar(ml_est)

## c) apply MLE estimation for each data set
Sys.time()
pbs_param_est <- lapply(pbs_data_list, mmEst_FREML_knownPopVar_forPBS)
Sys.time()

## Uncertainties estimated
pbs_param_est_final <- do.call(rbind, pbs_param_est)
cov(pbs.param.est_final[,1:2])
```

5.1

Limitations of the study

This procedure is intended for application with small sample sizes and can be used for large samples even though it is not a practical situation. The primary challenge encountered lies in the time-consuming nature of generating Monte Carlo simulations, which may require a significant amount of time for a personal computer to produce results. Apart from this, the method is not subject to any other limitations.

CHAPTER 6
CONCLUSION

This study primarily focuses on quantifying the uncertainty associated with the model coefficients of the calibration curve. It addresses a significant issue often overlooked by chemists, which involves using a linear regression model to fit the calibration curve while both the x and y variables contain measurement errors. To tackle this challenge, the study adopts the FREML model introduced by scientists Ripley and Thompson. The main objective is to develop a method for quantifying the uncertainties associated with the model coefficients using a numerical technique called parametric bootstrapping. In the initial approach, the study considers a scenario where the reported data exhibit population variances and employs likelihood estimation to determine the model coefficients. However, a practical limitation arises as reported uncertainties are typically calculated from around five replicates, rendering them unreliable as representations of population uncertainties. Consequently, the study shifts its focus to a second approach, where reported uncertainties represent sample variances. In this case, the study incorporates degrees-of-freedom as a source of uncertainty, as the variance of the sample variance is dependent on degrees-of-freedom.

For validation purposes, the study employs a coverage assessment process for both methods. The coverage calculation reveals that the original approach yields lower coverage compared to the proposed new method, particularly in cases with lower degrees-of-freedom. Furthermore, when comparing the calculated variances for a reported dataset with a smaller number of replicates, the study finds that the estimated uncertainties using the second approach are approximately two times larger than those using the first approach, particularly when degrees-of-freedom are limited. Notably, both methods yield similar results when the number of replicates is larger. In conclusion, the findings suggest that the second approach should be favored for estimating associated uncertainties when degrees-of-freedom are lower, while both methods perform comparably when the number of replicates is larger.

6.1

REFERENCES

[1] Sergei F. Boulyga, J. Alan Cunningham, Zsuzsanna Macsik, Joe Hiess, Maxim V. Peńkin, and Stephen J. Walsh. Development, validation and verification of an icp-ms procedure for a multi-element analysis of uranium ore concentrates. *J. Anal. At. Spectrom*, 32(11):2226–2237, 2017. doi:https://doi.org/10.1039/C7JA00180K.

[2] Roger L. Berger George Casella. *Statistical Inference*. CENGAGE, 2002.

[3] JCGM. Evaluation of measurement data — guide to the expression of uncertainty in measurement. *JCGM*, 100, 2008. doi:https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6.

[4] JCGM. Evaluation of measurement data , an introduction to the "guide to the expression of uncertainty in measurement" and related documents. *ISO/IEC GUIDE*, 98(1), 2009. https://www.iso.org/standard/46383.html.

[5] Myer Kutz. *Metrology and Measurement Systems*, volume 3. April 2016.

[6] Phillips SD Estler WT Doiron T Eberhardt KR Levenson MS. A careful consideration of the calibration concept. *J Res Natl Inst Stand Technol*, 106(2):371–379, 2001. doi:https://doi.org/10.6028/jres.106.014.

[7] Marlene Müller. *Generalized Linear Models. In: Gentle, J., Härdle, W., Mori, Y. (eds)*. Handbook of Computational Statistics. Springer Handbooks of Computational Statistics. Springer, Berlin, Heidelberg., 2012.

[8] Brian D. Ripley and Michael J. Thompson. Regression techniques for the detection of analytical bias. *Analyst*, 112:377–383, 1987. doi:https://doi.org/10.1039/AN9871200377.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023.

[10] M Thompson. Uncertainties in concentrations estimated from calibration experiments. Technical report, amc technical brief, March 2006. https://www.rsc.org/images/concentrations-calibration-experiments-technical-brief-22_tcm18-214840.pdf.

[11] Onghena P. Van Den Noortgate, W. Parametric and nonparametric bootstrap methods for meta-analysis. *Behavior Research Methods*, 37:11–22, 2005. doi:https://doi.org/10.3758/BF03206394.

[12] Stephen J. Walsh. Revisiting the fission track method for the analysis of particles in safeguards environmental samples. *Talanta*, pages 583–592, 2017. doi:https://doi.org/10.1016/j.talanta.2017.01.090.

[13] Stephen J. Walsh, Naida Dzigal, Ernesto Chinea-Cano, and Andreas Limbeck. Simple robust estimation of uranium isotope ratios in individual particles from la-icp-ms measurements. *J. Anal. At. Spectrom*, 32(6):1155–1165, 2017. doi:https://doi.org/10.1039/C7JA00089H.

[14] Stephen J. Walsh, Zsuzsanna Macsik, Dariusz Wegrzynek, Thomas Krieger, and Sergei Boulyga. Model diagnostics for detecting and identifying method repeatability outliers in precision studies: application to a homogeneity study under a two-stage nested anova. *J. Anal. At. Spectrom.*, 31(3):686–699, 2016. doi:https://doi.org/10.1039/C5JA00380F.

APPENDICES

APPENDIX 1 -VALIDATION OF NUMERICAL MLE AND PARAMETRIC BOOTSTRAP METHODS FOR UNCERTAINTY QUANTIFICATION USING FREML MODEL: COVERAGE PLOTS.

This appendix contains the results of Monte Carlo estimates of the statistical coverage of the FREML procedure described in Chapter 3 as per the Table 3.4.

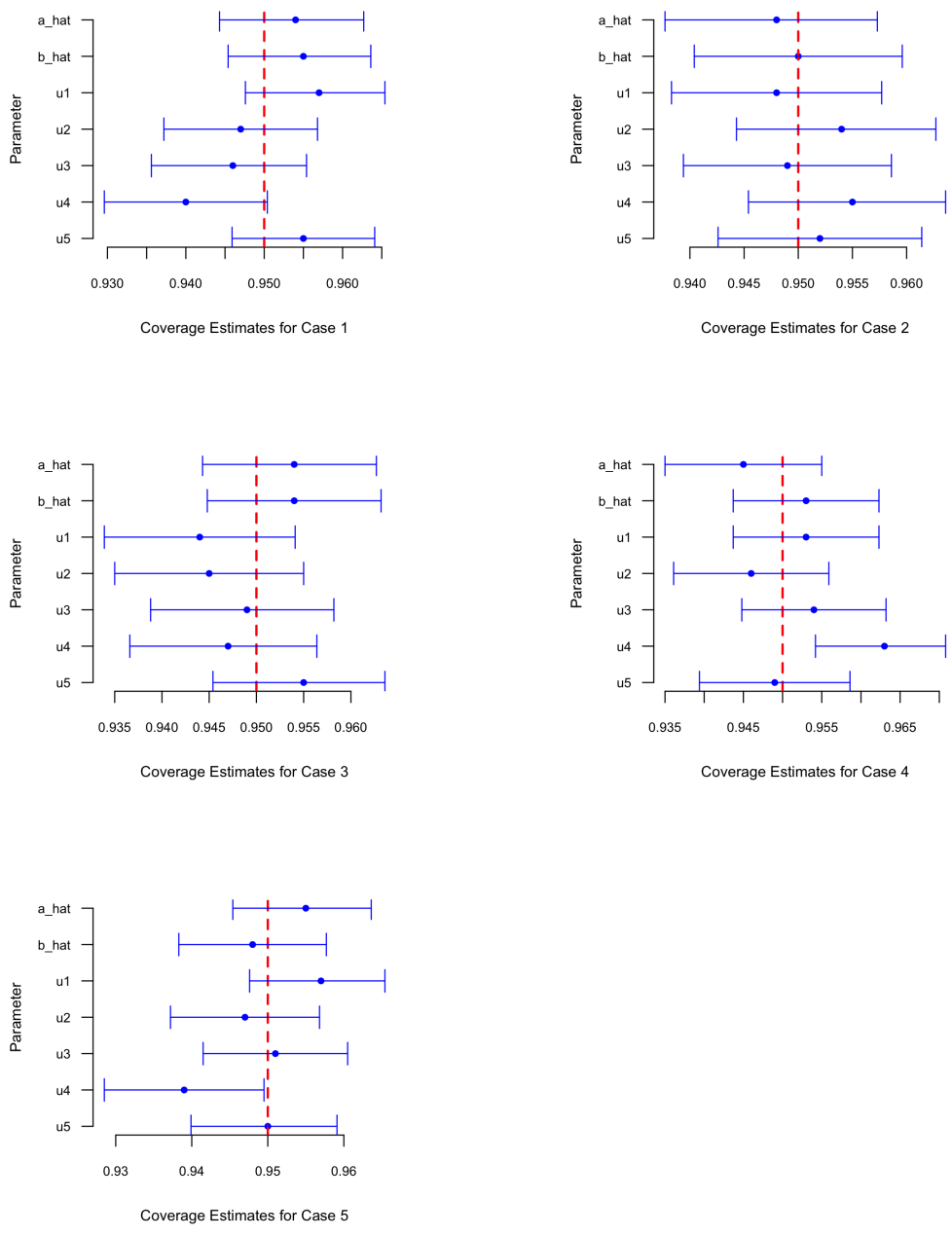


Figure 6.1: Estimated Confidence Intervals of Coverage for Study Cases 1-5 with Seven Parameters.

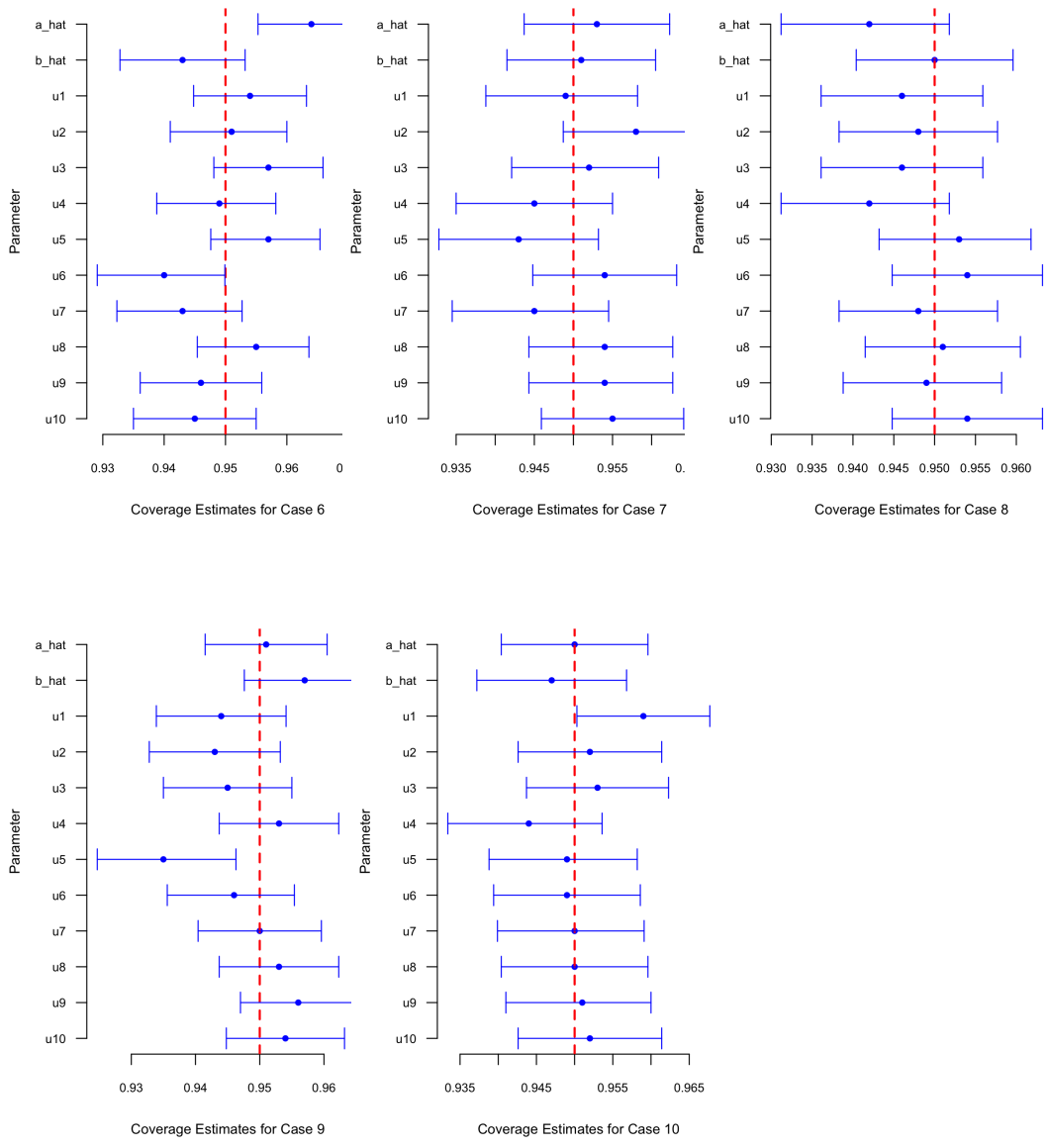


Figure 6.2: Estimated Confidence Intervals of Coverage for Study Cases 6-10 with 12 Parameters.

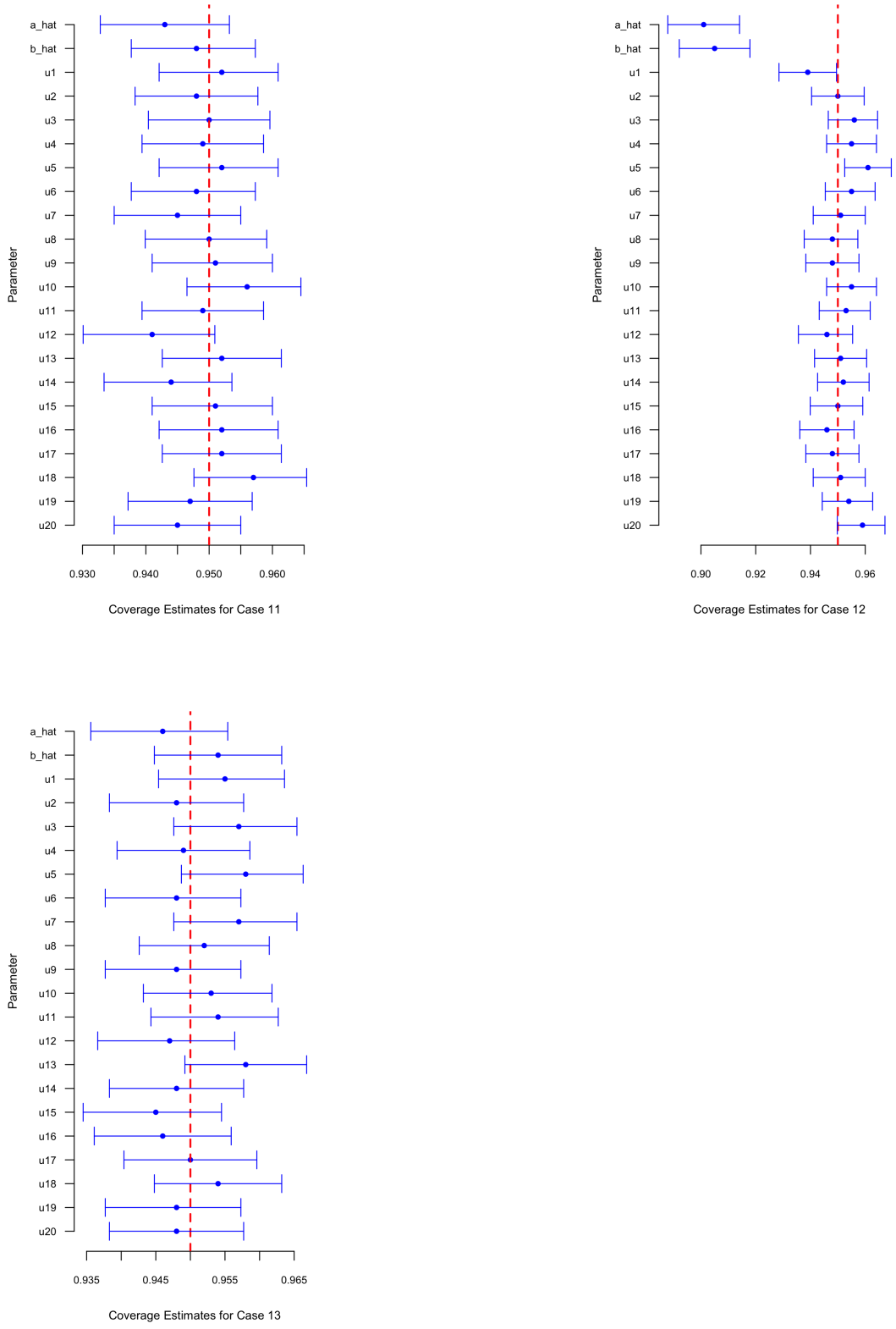


Figure 6.3: Estimated Confidence Intervals of Coverage for Study Cases 11-13 with 22 Parameters.

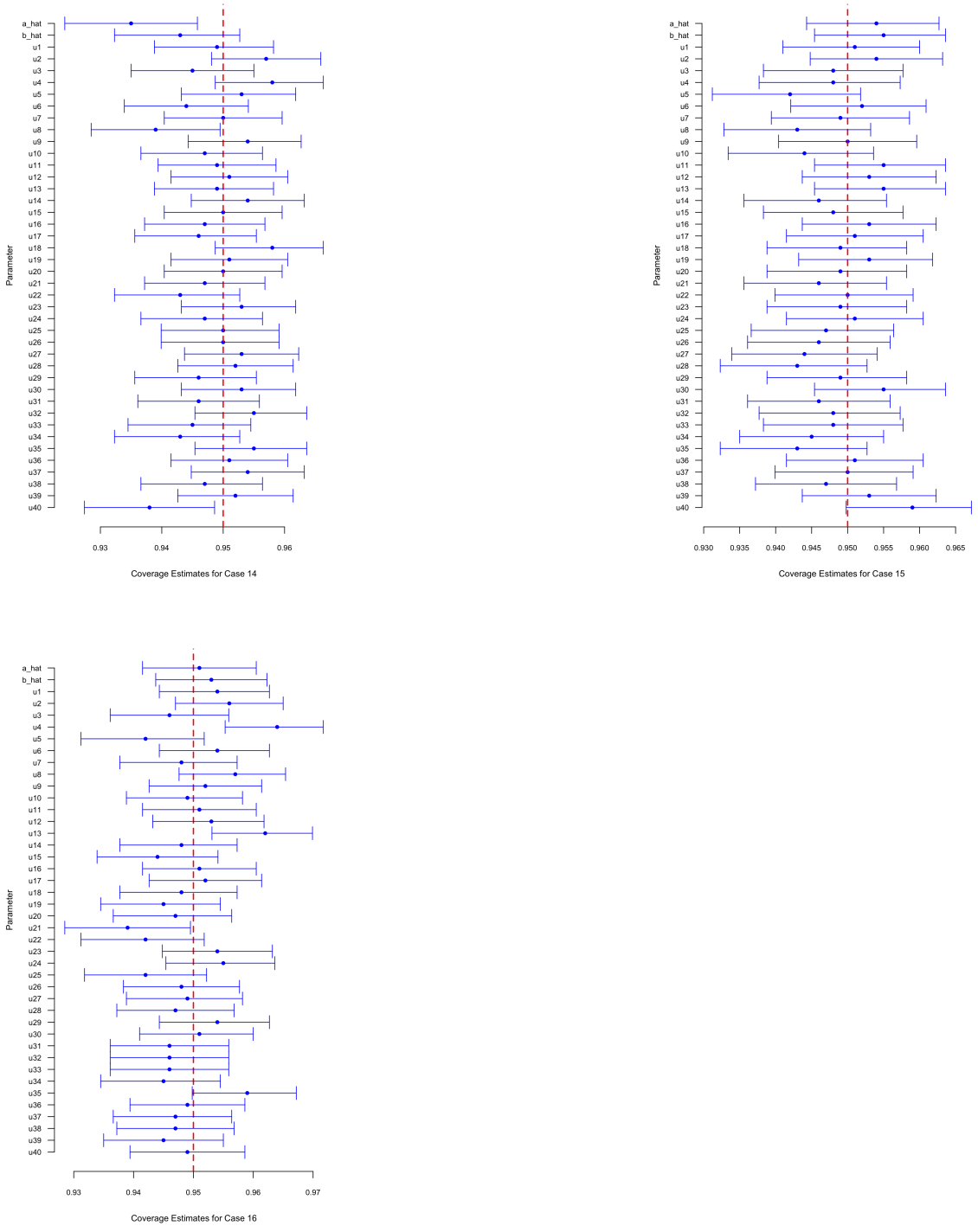


Figure 6.4: Estimated Confidence Intervals of Coverage for Study Cases 14-16 with 42 Parameters.

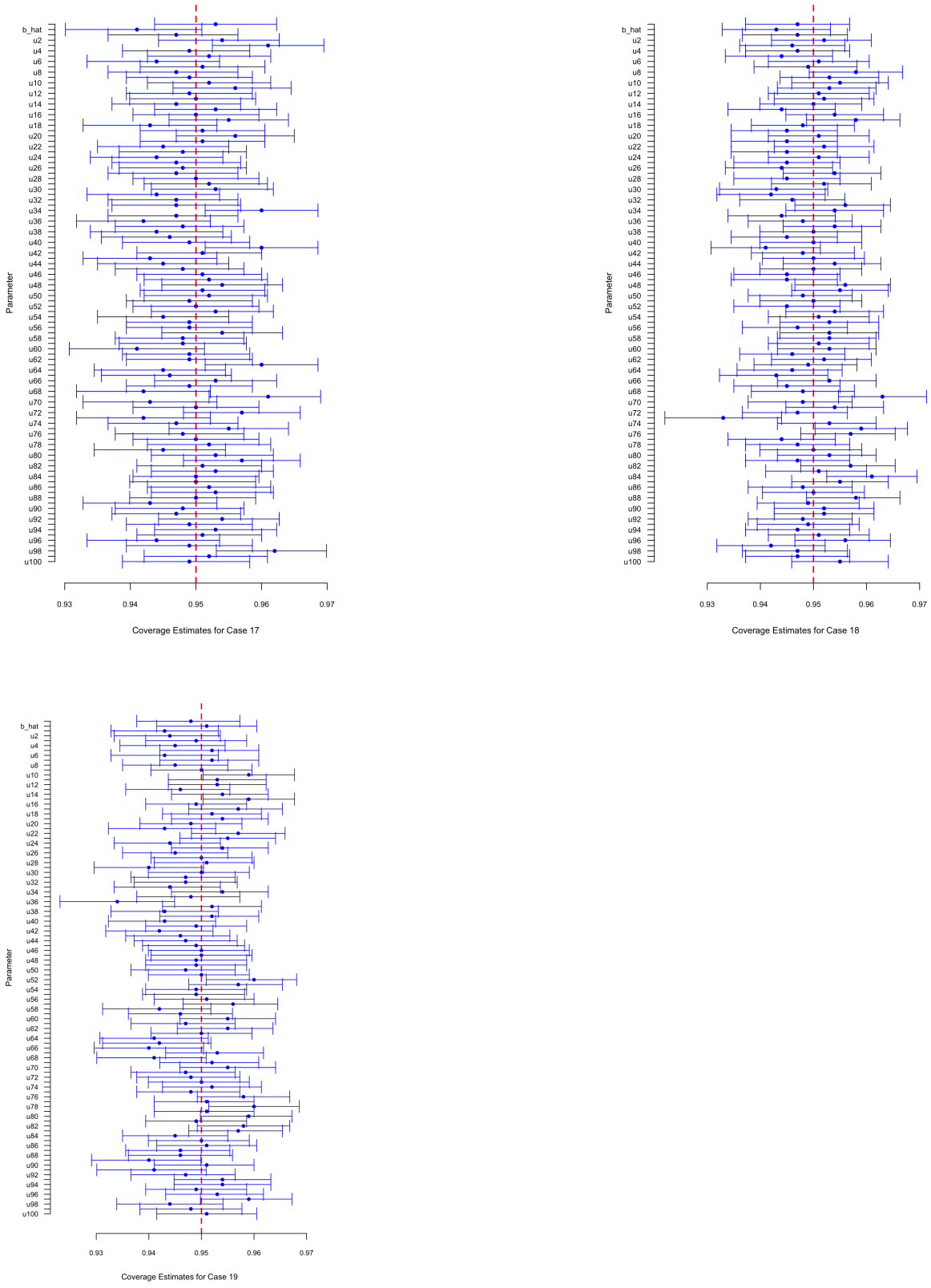
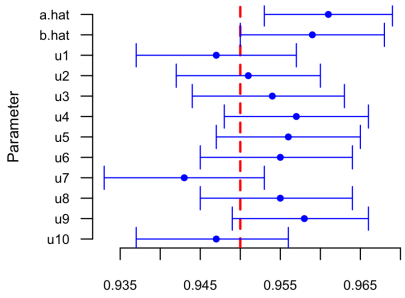


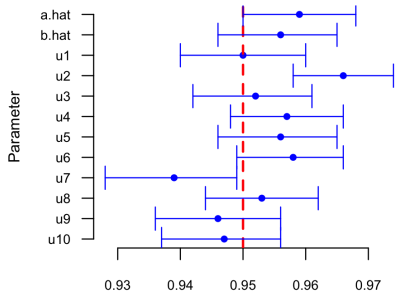
Figure 6.5: Estimated Confidence Intervals of Coverage for Study Cases 17-19 with 102 Parameters

APPENDIX 2 - VALIDATION OF NUMERICAL MLE AND PARAMETRIC BOOTSTRAP METHODS FOR UNCERTAINTY QUANTIFICATION USING EFREML MODEL: COVERAGE PLOTS.

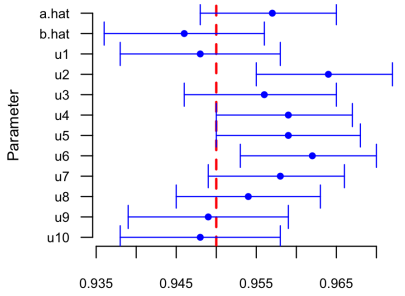
This appendix contains Monte Carlo estimates of the statistical coverage of the EFREML procedure as described in Chapter 4 for the cases shown in Table 4.1



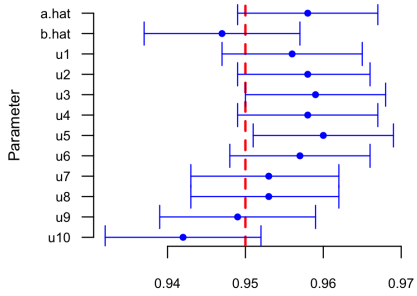
Coverage Estimates for Case 1.1: dfx = 5 and dfy = 5



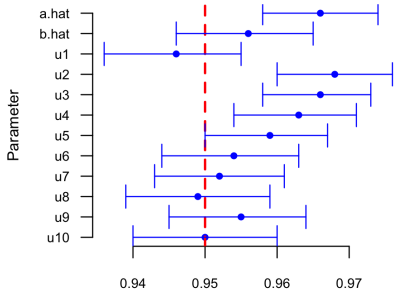
Coverage Estimates for Case 1.1: dfx = 10 and dfy = 5



Coverage Estimates for Case 1.3: dfx = 30 and dfy = 5

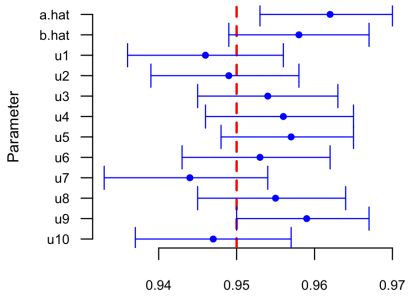


Coverage Estimates for Case 1.4: dfx = 50 and dfy = 5

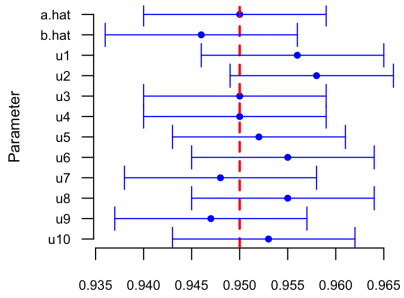


Coverage Estimates for Case 1.5: dfx = 100 and dfy = 5

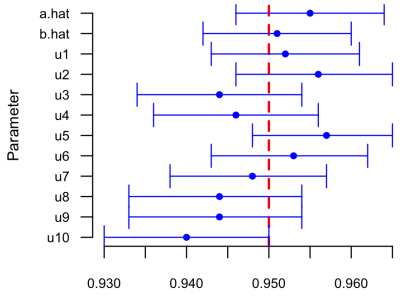
Figure 6.6: Estimated Confidence Intervals of EFREML via Parametric Bootstrap Procedure - Case 1.



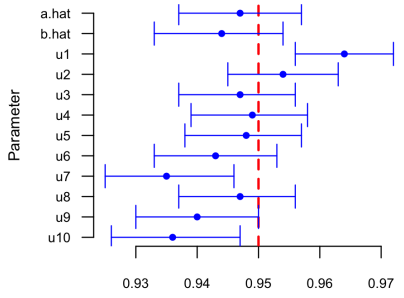
Coverage Estimates for Case 2.1: dfx = 5 and dfy = 5



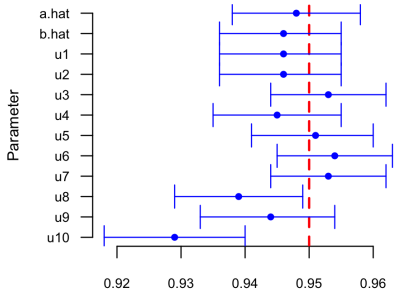
Coverage Estimates for Case 2.2: dfx = 10 and dfy = 10



Coverage Estimates for Case 2.3: dfx = 30 and dfy = 30

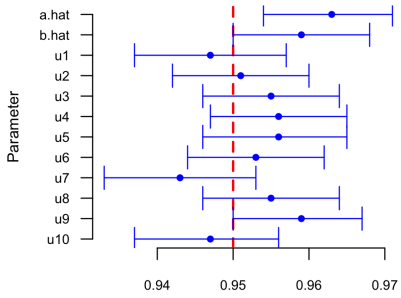


Coverage Estimates for Case 2.4: dfx = 50 and dfy = 50

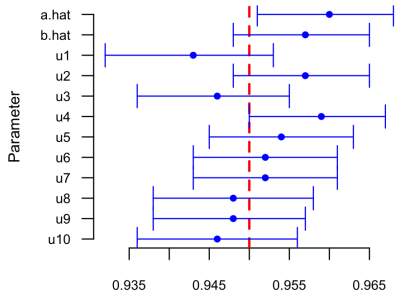


Coverage Estimates for Case 2.5: dfx = 100 and dfy = 100

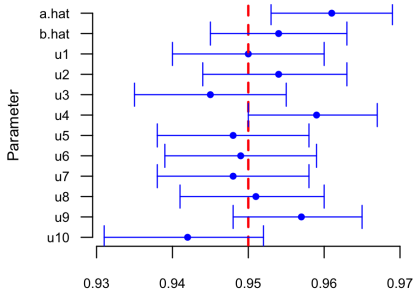
Figure 6.7: Estimated Confidence Intervals of EFREML via Parametric Bootstrap Procedure - Case 2.



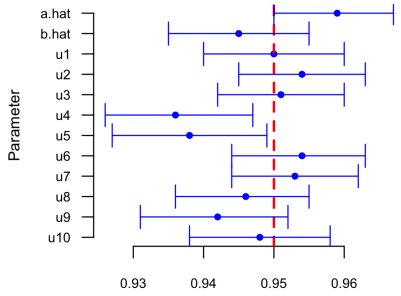
Coverage Estimates for Case 3.1: dfx = 5 and dfy = 5



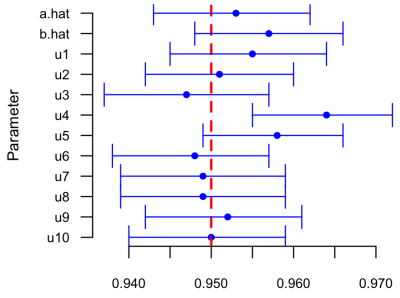
Coverage Estimates for Case 3.2: dfx = 5 and dfy = 10



Coverage Estimates for Case 3.3: dfx = 5 and dfy = 30

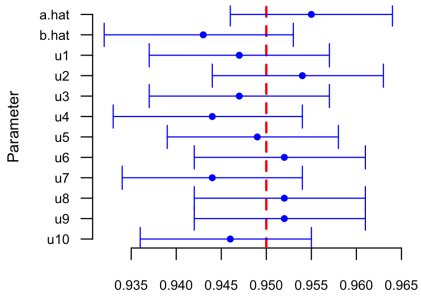


Coverage Estimates for Case 3.4: dfx = 5 and dfy = 50

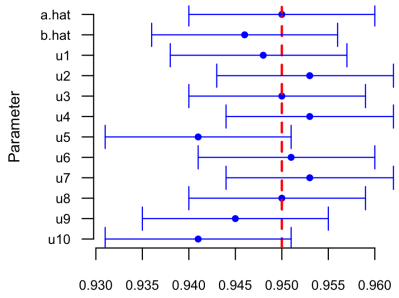


Coverage Estimates for Case 3.5: dfx = 5 and dfy = 100

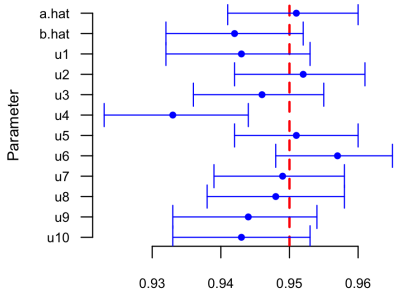
Figure 6.8: Estimated Confidence Intervals of EFREML via Parametric Bootstrap Procedure - Case 3.



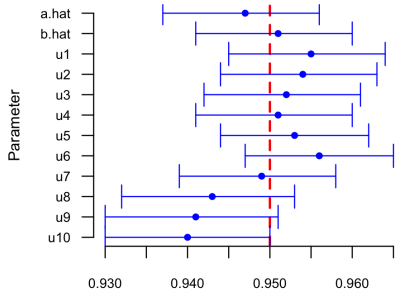
Coverage Estimates for Case 4.1: dfx = 5 and dfy = 100



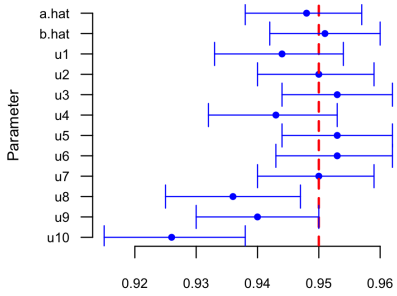
Coverage Estimates for Case 4.2: dfx = 10 and dfy = 100



Coverage Estimates for Case 4.3: dfx = 30 and dfy = 100

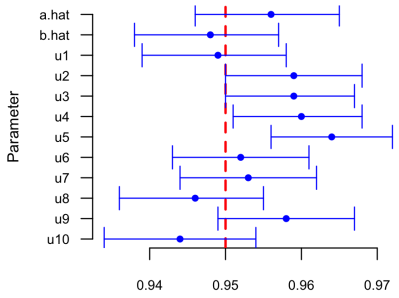


Coverage Estimates for Case 4.4: dfx = 50 and dfy = 100

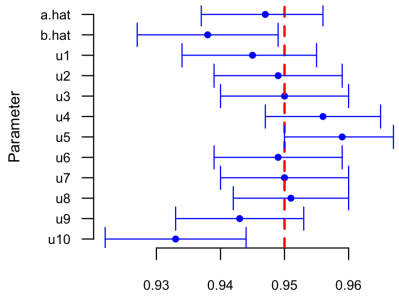


Coverage Estimates for Case 4.5: dfx = 100 and dfy = 100

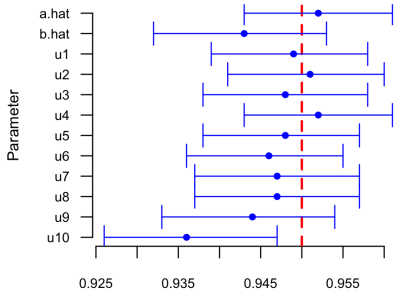
Figure 6.9: Estimated Confidence Intervals of EFREML via Parametric Bootstrap Procedure - Case 4.



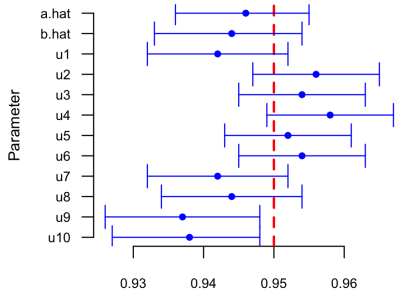
Coverage Estimates for Case 5.1: dfx = 100 and dfy = 5



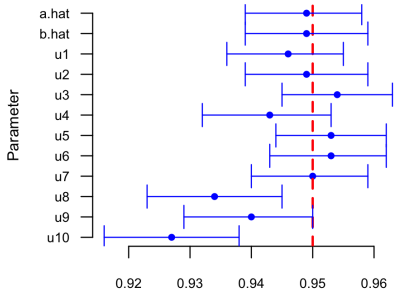
Coverage Estimates for Case 5.2: dfx = 100 and dfy = 10



Coverage Estimates for Case 5.3: dfx = 100 and dfy = 30



Coverage Estimates for Case 5.4: dfx = 100 and dfy = 50



Coverage Estimates for Case 5.5: dfx = 100 and dfy = 100

Figure 6.10: Estimated Confidence Intervals of EFREML via Parametric Bootstrap Procedure - Case 5.

APPENDIX 3 - INVESTIGATING THE UNDERESTIMATION OF ESTIMATED UNCERTAINTY IN CASES WHERE THE MEASUREMENT UNCERTAINTY OF X IS GREATER THAN THAT OF Y.

In this Chapter 2 appendix , we investigate the underestimation of $var(\hat{a})$, $var(\hat{b})$, and $cov(\hat{a}, \hat{b})$ calculated using FREML compared to those calculated using EFREML when the measurement uncertainty of x is larger than that of y under Case 1 in Table 4.1(degrees-of-freedom of y is fixed at 5 with degrees-of-freedom of x is varying). The study ratios are detailed in Section 4.3.1. From these findings, it is evident that when both the degrees-of-freedom for x and y are lower, the average underestimation ratios for all three comparisons are 0.5, and they remain below 1 even as the degrees-of-freedom of x increase. Consequently, these results suggest that when the degrees-of-freedom of y are lower, EFREML should be preferred for uncertainty quantification, even in cases where the measurement uncertainty of x exceeds that of y.

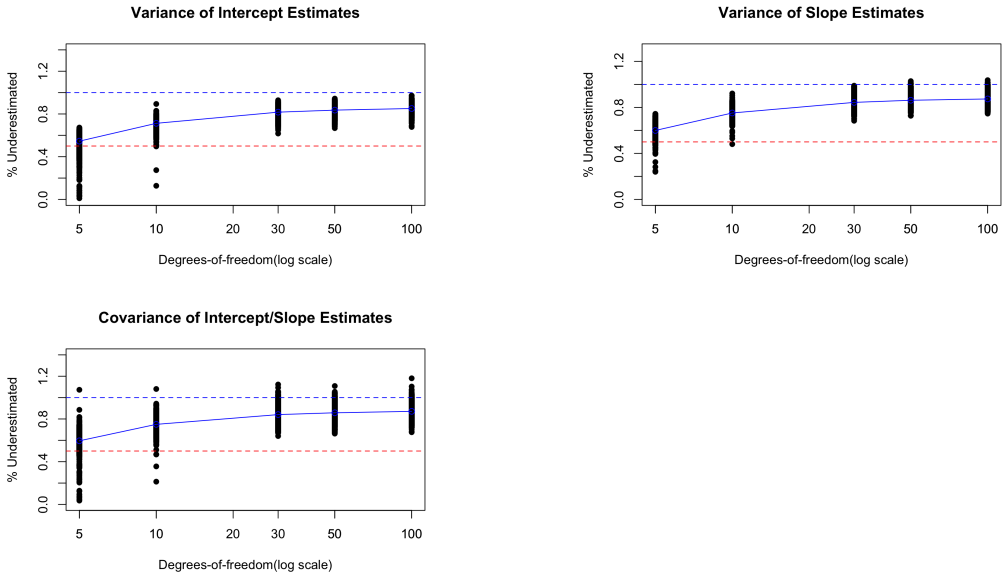


Figure 6.11: Comparison of Estimated Variance-Covariance using FREML and EFREML when the Measurement Uncertainty of x is greater than that of y.

APPENDIX 4 - FREML R FUNCTIONS AND EXAMPLE USAGE

This appendix contains code used to produce results in Chapter 3. Program was developed using R version 4.3.2 [9].

```
## ========================================================================
#
#   Authors: Aloka Dayarathne and Stephen J. Walsh
#
#   January 2024, Utah State University,
#                 Department of Mathematics and Statistics
#
#
#   README: ---------------------------------------------------------------
#   Contents: this script contains the full set of functions and an
#             updated workflow for implementing studies under the
#             FREML model where population variances are known.
#
#
#   Functions:
#   1.) simFREML_knownPopVar             --- simulate data from FREML model
#   2.) nLogLik_FREML_knownPopVar        --- FREML model negative log-likelihood
#   3.) mmEst_FREML_knownPopVar          --- optimization wrapper to perform ML
#                                            estimation given data
#   4.) mmEst_FREML_knownPopVar_forPBS   --- a simplified ML estimation routine
#                                            for use inside parametric
#                                            bootstrapping
#   5.) pbs_FREML_knownPopVar            --- the parametric bootstrap procedure,
#                                            output is a list of potential
#                                            (bootstrapped)
#                                            datasets that are supported by the
#                                            initial data sample
#   6.) sampDist_study_FREML_knownPopVar --- build sampling distributions given
#                                            a PBS dataset list
## ========================================================================

rm(list = ls())

## 1.0 Simulate data from FREML model with known parameters --------------------

simFREML_knownPopVar <- function(n = 10, a = 0, b = 1, u_min = 1, u_max = 1000,
                                 rsd_x = 0.05, rsd_x_r = 0.02,
                                 rsd_y = 0.15, rsd_y_r = 0.03) {
  # chemist will space these regularly
  u_vc <- seq(u_min, u_max, length = n)
  # make true v's
  v_vc <- a + b * u_vc
  sigma_rsd_x <- runif(n, min = rsd_x - rsd_x_r, max = rsd_x + rsd_x_r)
```

```
  sigma_x <- u_vc * sigma_rsd_x
  x_vc <- u_vc + rnorm(n, mean = 0, sd = sigma_x)
  # make y's
  sigma_rsd_y <- runif(n, min = rsd_y - rsd_y_r, max = rsd_y + rsd_y_r)
  sigma_y <- v_vc * sigma_rsd_y
  y_vc <- v_vc + rnorm(n, mean = 0, sigma_y)
  # pack into data frame
  D <- data.frame(x = x_vc, sd_x_pop = sigma_x, y = y_vc, sd_y_pop = sigma_y)
  true_params <- list(a = a, b = b, u = u_vc)
  result <- list(data = D, params = true_params)

  return(result)
}

## === STANDARD FREML FUNCTIONS ================================================

# evaluates the FREML negative log-likeihood given parameters and data
nLogLik_FREML_knownPopVar <- function(
    theta, # parameter vector: (a,b,u_1,...,u_n)
    y, # measurements on Y
    x, # measurements on X
    sigma_x, # pop SD of Y
    sigma_y # popSD of X
    ) {
  # separate parameters
  a <- theta[1]
  b <- theta[2]
  u <- theta[3:length(theta)]
  # Compute SSD_x
  x_err <- sum((x - u)^2 / sigma_x^2)
  # compute SSD_y
  y_err <- sum((y - (a + b * u))^2 / sigma_y^2)
  # return negative log likelihood
  res <- 0.5 * (x_err + y_err)
  return(res)
}



# apply ML estimation to a data set
mmEst_FREML_knownPopVar <- function(Dlist, # Dlist contains $data and $params
                                    allResult = F) {
  ## NOTE: this function is applied to original/master data
  #   to give ML estimates

  D <- Dlist$data
  # pick off variables
  x <- D$x
  sigma_x <- D$sd_x_pop
  y <- D$y
  sigma_y <- D$sd_y_pop

  theta <- c(lm(y ~ x)$coefficients, x) # select starting values for
  # optimization by OLS approach
```

```
  mle <- optim(
    fn        = nLogLik_FREML_knownPopVar,
    par       = theta,
    x         = x,
    y         = y,
    sigma_x   = sigma_x,
    sigma_y   = sigma_y,
    method    = "BFGS",
    control   = list(maxit = 50000)
  )

  if (allResult) {
    nlmOut <- mle
  } else {
    (
      nlmOut <- NULL
    )
  }
  # grab estimates
  estimates <- mle$par
  # organize output
  ans <- list(
    # needed to add data to keep known pop vars
    "data" = D,
    "true.params" = Dlist$params,
    "estimates" = list(
      "coefficients"    = estimates[1:2],
      "u_i"             = estimates[3:length(estimates)]
    ),
    "optim Output" = nlmOut
  )
  ans # Return the Results
}


mmEst_FREML_knownPopVar_forPBS <- function(D, # PBS Bootstrap FREML dataset
                                           allResult = F) {
  ## NOTE: This is FREML MLE but simplified for application
  #    to the bootrap data sets
  # pick off variables
  x <- D$x
  sigma_x <- D$sd_x_pop
  y <- D$y
  sigma_y <- D$sd_y_pop
  n <- length(x)
  theta <- c(lm(y ~ x)$coefficients, x) # select starting values for
  # optimization by OLS approach

  mle <- optim(
    fn        = nLogLik_FREML_knownPopVar,
    par       = theta,
    x         = x,
    y         = y,
    sigma_x   = sigma_x,
```

```
    sigma_y   = sigma_y,
    method    = "BFGS",
    control   = list(maxit = 50000)
  )

  # grab estimates
  estimates <- mle$par
  # organize output
  res <- as.data.frame(t(estimates))
  cnames <- c("a_hat", "b_hat", paste0("u", 1:n))
  names(res) <- cnames
  return(res) # Return the Results
}


# ------------------------------------------------------------------------------

## Parametric Bootstrap --------------------------------------------------------
#  given ML parameter estimates, use FREML model to make M new potential dataset
#  -s supported by the mother data

pbs_FREML_knownPopVar <- function(p_list, M = 5000) {
  data_t <- p_list$data
  sd_x_pop <- data_t$sd_x_pop
  sd_y_pop <- data_t$sd_y_pop
  coef_t <- p_list$estimates$coefficients
  a_hat <- coef_t[1]
  b_hat <- coef_t[2]
  ui_hat <- p_list$estimates$u_i
  n <- length(ui_hat)
  # PBS data list
  pbs_d_list <- list()
  for (i in 1:M) {
    # simulate new data sets
    x_new <- ui_hat + rnorm(n, 0, sd_x_pop)
    v <- a_hat + b_hat * ui_hat
    y_new <- v + rnorm(n, 0, sd_y_pop)
    # parametric bootstrapped data set
    data_pbs <- data.frame(
      x = x_new, sd_x_pop = sd_x_pop, y = y_new,
      sd_y_pop = sd_y_pop
    )
    pbs_d_list[[i]] <- data_pbs
  }
  return(pbs_d_list)
}

# this function applies ML estimation to a list of PBS bootstrapped
#  datasets
sampDist_study_FREML_knownPopVar <- function(Dl) {
  rt <- lapply(Dl, mmEst_FREML_knownPopVar_forPBS)
  res <- do.call(rbind, rt)
  return(res)
}
```

```
#-------------------------------------------------------------------------------

## -----------------------------------------------------------------------------
## Example Application:

# a) make the list of mother data sets - each of these mimics a dataset
#   reported by a chemist or lab analyst
N_data <- 1
D_list <- list()
for (i in 1:N_data) D_list[[i]] <- simFREML_knownPopVar()

# b) add ML ests to each set
ml_est_list <- lapply(D_list, mmEst_FREML_knownPopVar)



## c) get list of list of PBS datasets
## NOTE: CAN BE PARALLELIZED for coverage study
pbs_data_list <- lapply(ml_est_list, pbs_FREML_knownPopVar)

## MUST BE PARALLELIZED for coverage study
Sys.time()
pbs_param_est_list <- lapply(pbs_data_list, sampDist_study_FREML_knownPopVar)
Sys.time()

## the following 3 databases should be saved together
ml_est_list
pbs_data_list
pbs_param_est_list
```

APPENDIX 5 - EFREML R FUNCTIONS AND EXAMPLE USAGE

This appendix contains code used to produce results in Chapter 4. Program was developed using R version 4.3.2 [9].

```
## ========================================================================
#
#   Authors: Aloka Dayarathne and Stephen J. Walsh
#
#   January 2024, Utah State University,
#                 Department of Mathematics and Statistics
#
#
#   README: ---------------------------------------------------------------
#   Contents: this script contains the full set of functions and an
#             updated workflow for implementing studies under the Extended-
#             FREML model where population variances are unknown and
#             reported with corresponding degrees-of-freedom.
#
#
#   Functions:
#   1.) simFREMLext           --- simulate data from FREML model
#   2.) nLogLik_FREML_knownPopVar        --- EXTFREML model negative
#                                         log-likelihood
#   3.) mmEst_FREML_extended_A2_forPBS   --- optimization wrapper to perform ML
#                                         estimation given data
#   4.) mmEst_FREML_extended_A2_forPBS   --- a simplified ML estimation routine
#                                         for use inside parametric
#                                         bootstrapping
#   5.) pbs_FREML_extended_A2            --- the parametric bootstrap procedure,
#                                         output is a list of potential
#                                         (bootstrapped)
#                                         datasets that are supported by the
#                                         initial data sample
#   6.) sampDist_study_FREML_extended_A2 --- build sampling distributions given
#                                         a PBS dataset list
## ========================================================================

rm(list = ls())

## 1.0 Simulate data from ExtendedFREML model with known parameters ------------
## See Manuscript for definitions
simFREMLext <- function(n = 10, a = 0, b = 1, u_min = 1, u_max = 100,
                        rsd_x = 0.05, rsd_x_r = 0.02, rsd_y = 0.15,
                        rsd_y_r = 0.03, dfx = 5, dfy = 5) {
  dfxi <- rep(dfx, n)
  dfyi <- rep(dfy, n)
```

```
  # chemist will space these regularly
  u_vc <- seq(u_min, u_max, length = n)

  # make true v's
  v_vc <- a + b * u_vc

  sigma_rsd_x <- runif(n, min = rsd_x - rsd_x_r, max = rsd_x + rsd_x_r)
  sigma_x <- abs(u_vc * sigma_rsd_x)
  sxi <- sqrt(sigma_x^2 * rchisq(n = n, df = dfxi) / dfxi)

  x_vc <- u_vc + rnorm(n, mean = 0, sd = sigma_x)

  # make y's
  sigma_rsd_y <- runif(n, min = rsd_y - rsd_y_r, max = rsd_y + rsd_y_r)
  sigma_y <- abs(v_vc * sigma_rsd_y)

  y_vc <- v_vc + rnorm(n, mean = 0, sigma_y)
  syi <- sqrt(sigma_y^2 * rchisq(n = n, df = dfyi) / dfyi)

  ## make dataframe, note change of name of samp SDs
  D <- data.frame(
    x = x_vc, sd_x_samp = sxi, dfxi = dfxi,
    y = y_vc, sd_y_samp = syi, dfyi = dfyi
  )

  true_params <- list(
    a = a, b = b, u = u_vc, var_pop.x = sigma_x^2,
    var_pop_y = sigma_y^2
  )
  result <- list(data = D, params = true_params)

  return(result)
}

## === Extended FREML FUNCTIONS ================================================

# evaluates the EXT FREML negative log-likeihood given parameters and data
nLogLik_FREML_knownPopVar <- function(
    theta, # parameter vector: (a,b,u_1,...,u_n)
    y, # measurements on Y
    x, # measurements on X
    sigma_x, # pop SD of Y
    sigma_y # popSD of X
    ) {
  # separate parameters
  a <- theta[1]
  b <- theta[2]
  u <- theta[3:length(theta)]
  # Compute SSD_x
  x_err <- sum((x - u)^2 / sigma_x^2)
  # compute SSD_y
  y_err <- sum((y - (a + b * u))^2 / sigma_y^2)
  # return negative log likelihood
  res <- 0.5 * (x_err + y_err)
```

```
  return(res)
}



# apply ML estimation to a data set
mmEst_FREML_extended_A2 <- function(Dlist, allResult = FALSE) {
  D <- Dlist$data
  x <- D$x
  y <- D$y
  sd_x_mle <- D$sd_x_samp
  sd_y_mle <- D$sd_y_samp
  n <- length(x)

  # original FREML procedure
  theta_0 <- c(lm(y ~ x)$coefficients, x)

  mle <- optim( # nlm is a function for Non-Linear Minimization
    fn        = nLogLik_FREML_knownPopVar,
    par       = theta_0,
    x         = x,
    y         = y,
    sigma_x   = sd_x_mle,
    sigma_y   = sd_y_mle,
    method    = "Nelder-Mead",
    control   = list(maxit = 50000) # , factr = 1E-10, pgtol = 1E-30, trace = 0)
  )
  # grab estimates
  estimates <- mle$par

  if (allResult) {
    nlmOut <- mle
  } else {
    (
      nlmOut <- NULL
    )
  }

  ans <- list(
    "data" = D,
    "true_params" = Dlist$params,
    "estimates" = list(
      "coefficients" = estimates[1:2],
      "u_i"          = estimates[3:(n + 2)]
    ),
    "optim Output" = nlmOut
  )

  return(ans) # Return the Results
}


mmEst_FREML_extended_A2_forPBS <- function(D, allResult = F) {
  x <- D$x
```

```
  y <- D$y
  sd_x_samp <- D$sd_x_samp_s
  sd_y_samp <- D$sd_y_samp_s
  n <- length(x)
  # apply estimation under known-pop-var freml to initiate this optim search
  # original FREML procedure
  theta_0 <- c(lm(y ~ x)$coefficients, x)

  mle <- optim( # nlm is a function for Non-Linear Minimization
    fn        = nLogLik_FREML_knownPopVar,
    par       = theta_0,
    x         = x,
    y         = y,
    sigma_x   = sd_x_samp,
    sigma_y   = sd_y_samp,
    method    = "Nelder-Mead",
    control   = list(maxit = 50000) # , factr = 1E-10, pgtol = 1E-30, trace = 0)
  )
  # grab estimates
  estimates <- mle$par
  conv <- mle$convergence

  # organize output
  if (conv == 0) {
    res <- as.data.frame(t(estimates))
  } else {
    res <- as.data.frame(t(rep(NA, length(estimates))))
  }

  cnames <- c("a_hat", "b_hat", paste0("u", 1:n))
  names(res) <- cnames
  return(res)
}


# ------------------------------------------------------------------------------

## Parametric Bootstrap ----------------------------------------------------
#   given ML parameter estimates, use FREML model to make M new potential
#   datasets supported by the mother data
## make PBS datasets given param estimates ---------------------------------
pbs_FREML_extended_A2 <- function(p_list, M = 5000) {
  # get data
  data_t <- p_list$data
  sd_x_samp <- data_t$sd_x_samp
  sd_y_samp <- data_t$sd_y_samp
  dfxi <- data_t$dfxi
  dfyi <- data_t$dfyi


  # first, pick off ML ests of params
  mest_t <- p_list$estimates
  a_hat <- mest_t$coefficients[1]
  b_hat <- mest_t$coefficients[2]
```

```
  ui_hat <- mest_t$u_i
  n <- length(ui_hat)

  # PBS data list
  pbs_d_list <- list()
  for (i in 1:M) {
    v_s <- a_hat + b_hat * ui_hat

    # bootstrap a new data sample variance from observed sample
    #  variance acting in place of \sigma^2 as MLE
    sd_x_samp_s <- sqrt(rchisq(n, df = dfxi) * (sd_x_samp^2) / dfxi)
    sd_y_samp_s <- sqrt(rchisq(n, df = dfyi) * (sd_y_samp^2) / dfyi)

    # propogate uncertainty in knowledge of true \sigma_{xi}^2 and
    #  true \sigma_{yi}^2 into boostrapped data (x, y)
    sd_x_pop_s <- sqrt(dfxi * (sd_x_samp^2) / rchisq(n, df = dfxi))
    sd_y_pop_s <- sqrt(dfyi * (sd_y_samp^2) / rchisq(n, df = dfyi))

    # sample new X and Y
    x_s <- ui_hat + rnorm(n, 0, sd_x_pop_s)
    y_s <- v_s + rnorm(n, 0, sd_y_pop_s)

    data_pbs <- data.frame(
      x = x_s, sd_x_samp_s = sd_x_samp_s,
      y = y_s, sd_y_samp_s = sd_y_samp_s
    )
    pbs_d_list[[i]] <- data_pbs
  }
  return(pbs_d_list)
}

# this function just makes the PBS param sampling distribution list to dataframe
sampDist_study_FREML_extended_A2 <- function(Dl) {
  # this is to make sampling distributions for
  #  one mother data set
  rt <- lapply(Dl, mmEst_FREML_extended_A2_forPBS)
  res <- do.call(rbind, rt)
  return(res)
}


#-------------------------------------------------------------------------------

## -----------------------------------------------------------------------------
## Example Application:

# a) make the list of mother data sets - each of these mimics a dataset
#   reported by a chemist or lab analyst
N_data <- 1
D_list <- list()
for (i in 1:N_data) D_list[[i]] <- simFREMLext()

# b) add ML ests to each set
ml_est_list <- lapply(D_list, mmEst_FREML_extended_A2)
```

```
## c) get list of list of PBS datasets
## NOTE: CAN BE PARALLELIZED for coverage study
pbs_data_list <- lapply(ml_est_list, pbs_FREML_extended_A2)

## MUST BE PARALLELIZED for coverage study
Sys.time()
pbs_param_est_list <- lapply(pbs_data_list, sampDist_study_FREML_extended_A2)
Sys.time()

## the following 3 databases should be saved together
ml_est_list
pbs_data_list
pbs_param_est_list
```

APPENDIX 6 - FREML FUNCTIONS APPLIED TO DATA FROM RIPLEY AND THOMPSON (1987).

This appendix contains code used to reproduce analysis performed in Ripley and Thompson paper in Chapter 3. The following code is a usage of the codes in Appendix 4. Program was developed using R version 4.3.2 [9].

```
## Example data set found from Ripley and Thompson(1987) paper

x <- c(
  8.71, 7.01, 3.28, 5.60, 1.55, 1.75, 0.73, 3.66, 0.9, 9.39, 4.39, 3.69, 0.34,
  1.94, 2.07, 1.38, 1.81, 1.27, 0.82, 1.88, 5.66, 0, 0, 0.4, 0, 1.98, 10.21,
  4.64, 5.66, 19.25
)
sx <- c(
  1.92, 1.56, 0.76, 1.26, 0.39, 0.43, 0.22, 0.84, 0.25, 2.07, 1, 0.84, 0.13,
  0.47, 0.5, 0.36, 0.45, 0.33, 0.23, 0.46, 1.27, 0.06, 0.06, 0.15, 0.06, 0.48,
  2.24, 1.05, 1.27, 4.18
)
y <- c(
  7.35, 7.92, 3.4, 5.44, 2.07, 2.29, 0.66, 3.43, 1.25, 6.58, 3.31, 2.72, 2.32,
  1.5, 3.5, 1.17, 2.31, 1.88, 0.44, 1.37, 7.04, 0, 0.49, 1.29, 0.37, 2.16,
  12.53, 3.9, 4.66, 15.86
)
sy <- c(
  2.07, 2.23, 0.96, 1.53, 0.59, 0.65, 0.19, 0.97, 0.36, 1.85, 0.93, 0.77, 0.66,
  0.43, 0.99, 0.33, 0.66, 0.54, 0.13, 0.4, 1.98, 0.01, 0.15, 0.37, 0.12, 0.62,
  3.51, 1.1, 1.31, 4.45
)

RT_data <- list(data = data.frame(x, sd_x_pop = sx, y, sd_y_pop = sy))

## a) MLE estimates for original data set
ml_est <- mmEst_FREML_knownPopVar(RT_data)

## b) get list of PBS datasets
pbs_data_list <- pbs_FREML_knownPopVar(ml_est)

## c) apply MLE estimation for each data set
Sys.time()
pbs_param_est <- lapply(pbs_data_list, mmEst_FREML_knownPopVar_forPBS)
Sys.time()

## Uncertainties estimated
pbs_param_est_final <- do.call(rbind, pbs_param_est)
cov(pbs.param.est_final[,1:2])
```