University of Wollongong
Research Online

University of Wollongong Thesis Collection

University of Wollongong Thesis Collections

2011

Analysis of probabilistically linked data

Klairung Samart
University of Wollongong

Recommended Citation

Samart, Klairung, Analysis of probabilistically linked data, Doctor of Philosophy thesis, School of Mathematics and Applied Statistics, University of Wollongong, 2011. http://ro.uow.edu.au/theses/3513

Research Online is the open access institutional repository for the University of Wollongong. For further information contact Manager Repository Services: morgan@uow.edu.au.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Analysis of Probabilistically Linked Data

A thesis submitted in fulfilment of the

requirements for the award of the degree

Doctor of Philosophy

from

University of Wollongong

by

Klairung Samart

BSc (Hons), GradDip, MStat (University of Newcastle)

School of Mathematics and Applied Statistics

2011

CERTIFICATION

I, Klairung Samart, declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Mathematics and Applied Statistics, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Klairung Samart

12 December, 2011

Acknowledgements

This thesis would not have been possible without the help and support of many people. I would firstly like to express the deepest appreciation to my supervisor, Professor Ray Chambers, whose encouragement, guidance, great patience and support from the initial to the final level enabled me to develop an understanding of the subject. Without his guidance and persistent help this thesis would not have been possible. One simply could not wish for a better or friendlier supervisor.

I would like to acknowledge Professor David Steel, Professor Matt Wand, Associate Professor Robert Clark and Dr Pam Davy for their generosity to have me in their classes. I have gained a lot of knowledge from the courses which substantially contributed to the development of this work.

I would also like to thank Dr Gunky Kim for his help, comments and advice throughout my thesis with his patience and knowledge. Especially, when I first arrived and had trouble understanding the previous work.

I am also grateful to all my fellow PhD students, particularly Preeya, Amirah, Diane and Bothaina for sharing the experiences. Also, thanks to my dear friend, Somphong for his advice in LaTex and programming.

This research has been supported by a Prince of Songkla University Scholarship and a University Postgraduate Award (UPA) from the University of Wollongong. I am grateful for their confidence in me. Especially, thanks to the Department of Mathematics and Statistics, Prince of Songkla University for giving me this opportunity and patiently waiting for me.

Lastly, I would like to thank my family; father, mother and brother, for all their love, encouragement, and always believing in me. Most importantly, I would like to show my gratitude to my beloved husband Warakorn, for his unfailing love, patience, and encouragement throughout my journey away from home. Without his encouragement and understanding it would have been impossible for me to finish this work.

Abstract

Probabilistic matching of records from different data sets is often used to create linked data sets for use in research in health, epidemiology, economics, demography and sociology. Clearly, this type of matching can lead to linkage errors, which in turn can lead to bias and increased variability when standard statistical estimation techniques are used with the linked data. Recently, an inferential framework for statistical modelling using probabilistically linked data has been defined, which has then been used to develop modified estimation methods for regression models based on the assumption that the correctly linked records are mutually uncorrelated. In real life, however, measurements are usually made on clusters of correlated statistical units, such as people in a family, patients in a hospital or students in a school, and when analyzing such data, linear mixed models are often used.

In this thesis we show how this inferential framework can be used to develop unbiased regression parameter estimates when fitting a linear mixed model to probabilistically linked data. Furthermore, since estimation of variance components is also an important objective when fitting a mixed model, we develop appropriate modifications to standard methods of variance components estimation in order to account for linkage error. In particular, we focus on three widely used methods of variance components estimation: analysis of variance (ANOVA), maximum likelihood (ML) and restricted maximum likelihood (REML). A simulation study investigates the bias and variability of parameter estimates obtained by methods developed in this work. Simulation results indicate that all methods developed here perform reasonably well.

An application to longitudinal modeling is further investigated. In this situation, we focus on fitting linear mixed models to linked longitudinal registers. That is, more than two registers are linked and linkage errors occur across the entire registers. The results from a simulation study illustrate the performance of this approach, and show that although there is improved efficiency compared to the naive method which ignores the linkage errors, there are some issues that still need further investigation and improvement.

Contents

1 Introduction **1**

2 Literature Review **6**

2.1 Introduction 6

2.2 Methods for Linking Data 6

2.2.1 Techniques Used in Record Linkage 7

2.2.2 The Practical Steps of Record Linkage 7

2.2.3 The Fellegi-Sunter Model of Record Linkage 8

2.2.4 Parameter Estimation Methodology 11

2.3 Measurement Error 12

2.3.1 Ordinary Least Squares and Measurement Error 13

2.3.2 Measurement Variance Known 15

2.3.3 Ratio of Measurement Variances Known 17

2.3.4 Instrumental Variable Estimation 18

2.3.5 Linkage Error Is a Different Type of Measurement Error . . . 20

2.4 Research on the Impact of Linkage Errors 21

2.5 Longitudinal Linked Data 28

3 Linear Mixed Model Using Linked Data **31**

3.1 Linear Mixed Models 34

3.2 Bias-Corrected WLS Inference 35

3.3 Efficient Linear Estimation Using Linked Data 39

3.4 A Fixed Population Approach 41

3.5 Variance Estimation When Linkage Probabilities Are Estimated . . . 43

4 Estimation of Variance Components 48

4.1 Analysis of Variance (ANOVA) 48

4.2 Pseudo-Maximum Likelihood (Pseudo-ML) 53

4.2.1 Point Estimation 53

4.2.2 Confidence Intervals 57

4.3 Pseudo-Restricted Maximum Likelihood (Pseudo-REML) 59

4.4 Comparison Between ANOVA, ML and REML Estimation 63

5 Application to Longitudinal Modeling 64

5.1 Background and Assumptions 65

5.2 Weighted Least Squares Inference 69

5.3 Estimation of Variance Components 72

5.3.1 Analysis of Variance (ANOVA) 73

5.3.2 Pseudo-Maximum Likelihood (Pseudo-ML) 75

5.3.3 Pseudo-Restricted Maximum Likelihood (Pseudo-REML) . . . 77

6 Simulation Analysis 78

6.1 Simulation of Linear Mixed Model for Cross-Sectional Linked Data 78

6.2 Simulation of Linear Mixed Model for Longitudinal Linked Data 83

6.3 Sensitivity of Correct Linkage Probabilities 89

7 Conclusions and Further Research 93

7.1 Summary 93

7.2 Further Research 96

A Approximating the K matrix 99

B ANOVA Estimation **108**

B.1 Variance Components Estimation for Perfect Linkage 108

B.2 Variance Components Estimation for Imperfect Linkage 112

C Optimal Estimating Function **116**

D Simulation Study Code **121**

D.1 Simulation of Linear Mixed Model with Cross-Sectional Linked Data 121

D.1.1 Simulation of Known Lambda Case 121

D.1.2 Simulation of Estimated Lambda Case 145

D.2 Simulation of Linear Mixed Model with

Longitudinal Linked Data 152

D.2.1 Simulation of Known Lambda Case 152

D.2.2 Simulation of Estimated Lambda Case 171

Bibliography **179**

List of Figures

5.1 Spinal bone mineral data versus age for female subjects. 65

6.1 Boxplots of percentage relative errors of coefficient parameters generated by different estimators in linear mixed model simulations. Note that N is the Naïve estimator while RE is the REML estimator. . . . 82

6.2 Boxplots of percentage relative errors of variance components parameters generated by different estimators in linear mixed model simulations. Note that N is the Naïve estimator while RE is the REML estimator. 83

6.3 Boxplots of percentage relative errors of coefficient parameters generated by different estimators in linear mixed model simulations from longitudinal registers 86

6.4 Boxplots of percentage relative errors of variance components parameters generated by different estimators in linear mixed model simulations from longitudinal registers 87

List of Tables

3.1 Specification of $\hat{\mathbf{D}}_{qr}$ and $\partial_\theta \hat{\mathbf{U}}_{qr}$ in (3.25) for the linear case. 47

5.1 Illustration of the true linked longitudinal data. 66

6.1 Simulation results for estimators of the regression coefficients of the linear mixed model 79

6.2 Simulation results for estimators of the variance components of the linear mixed model 80

6.3 Actual coverages of nominal 95% confidence intervals for the parameters of the linear mixed model 81

6.4 Simulation results for estimators of the regression coefficients of the linear mixed model for longitudinal registers 84

6.5 Simulation results for estimators of the variance components of the linear mixed model for longitudinal registers 85

6.6 Actual coverages of nominal 95% confidence intervals for the parameters of the linear mixed model for longitudinal registers 85

6.7 Bias and standard error of standard error estimators of the regression coefficients estimators when linkage probabilities were estimated from longitudinal registers 88

6.8 Bias and standard error of standard error estimators of the variance components estimators when linkage probabilities were estimated from longitudinal registers 89

6.9 Simulation results for estimators of the regression coefficients under true value of $\lambda = 0.75$ 90

6.10 Simulation results for estimators of the variance components under true value of $\lambda = 0.75$ 91

6.11 Actual coverages of nominal 95% confidence intervals for the regression coefficients estimators of the linear mixed model under true value of $\lambda = 0.75$ 92

6.12 Actual coverages of nominal 95% confidence intervals for the variance components estimators of the linear mixed model under true value of $\lambda = 0.75$ 92

Chapter 1

Introduction

There has been a vast growth in information technology over the last three decades. Databases containing huge amounts of information in health related research and many other scientific areas have been created. Nevertheless, to investigate relationships between variables on different databases, one needs record linkage between these databases, particularly probability linkage which has been developed by Fellegi and Sunter [17].

According to Gill *et al.* [22], record linkage is the process of bringing together related records, or abstracts of records, compiled at different times and in different places. When it is required to link such records, there should be a unique personal identifier to link the records. Unfortunately, such a unique identifier does not generally exist [7]. Therefore, to identify different records relating to the same individual, Gill *et al.* [22] suggest that one can use other methods such as the use of surnames, forenames, and dates of birth. The three main steps in linking records are: blocking of records that have a potential relationship, matching to determine if records within a block are likely to be related and linking matched records so they can be analyzed as information for the one individual [22, 33].

Linked data sets are particular useful for research in health, epidemiology, economics, demography, sociology and many other scientific areas [7, 33]. For example, the data obtained from linking physician billing claims held by the Ontario Health Insurance Plan with data for consenting Ontario respondents to the 1994/95 Cana-

dian National Population Health Survey were used for an analysis of the association between utilization and costs of physician services and incidence of self-reported chronic conditions for residents of Ontario province in Canada [34]. In Australia, the use of record linkage has been summarized by Sibthorpe *et al.* [61]. There are some examples of linkage systems used for special research purposes such as the Maternal and Child Health Linked Database used in studies of perinatal and paediatric outcomes [63], the Roadwatch Road Injury Research Database [18], and a pilot study of linkage of Health Insurance Commission Medicare records [42].

Data linkage is not only used in the health related areas. Chesher and Nesheim [9] described in a review commissioned by the UK Department for Trade and Industry that data linkage has been extensively used in economic research, particularly in the United States. The US Census Bureau has been working on linking business data sets focusing on firms for almost 40 years. More recently, a project linking firms and workers has also been started. In Australia, the Census Data Enhancement Initiative of the Australian Bureau of Statistics (ABS) aims at creating a Statistical Longitudinal Census Dataset that combines census data from the same individuals from different censuses to build a research resource for longitudinal analysis of the Australian population. In the UK, record linkage has also been used by the Interdepartmental Migration and Population Task Force to improve migration and population statistics in the UK by linking administrative, health register, school enrollment and university student data with incoming passenger survey and labour force survey data to create an integrated longitudinal data set for in-depth analysis of the UK migrant experience [7].

In terms of businesses, record linkage is particularly useful to remove duplicate entries from mailing lists which in turn will reduce both printing and mailing costs, and otherwise will operate more efficiently [29]. Moreover, businesses might also use record linkage to improve the functionality of their databases. For example, in the corporate combination between Bank of America and Countrywide Mortgage, it was widely conjectured that a major factor driving the deal was Bank of America's

desire to obtain e-mail addresses of its retail customers from Countrywide Mortgage's database [29].

In order to link data sets from different sources, a probabilistic matching of records from one data set to another is often used. This is the most sophisticated technique in record linkage [32] which can be done by using the matching variables present in both data sets to maximize the probability that the values of the variables making up the linked record are the correct measurements for the population unit corresponding to that record [7]. Clearly, this type of record matching can lead to *linkage errors*. The definition and importance of linkage errors are discussed by Chambers [7] who points out that

> *"Linkage errors are the errors caused by incorrectly linking different population units as well as the errors caused by not linking the same population units in the data sets that are linked. These errors are a particular type of measurement error, and can lead to biased inference unless appropriate steps are taken to control and/or adjust for this bias."*

Unfortunately, these errors are typically ignored when analysis is undertaken. Although there has been a number of statistical methods for linking data sets, there has been little methodological research carried out on the impact of linkage errors on analysis of linked data. This can be seen in Neter *et al.* [48], and Lahiri and Larsen [38]. A methodological framework for analysis of linked data was developed in Chambers [7]. In his work, appropriate modifications to standard statistical analysis methods were used to ensure that they remain unbiased when applied to probabilistically linked data. The framework was based on modeling the relationship between the probabilistically linked data and the true data that would be obtained if error free linkage were possible. A simple linear regression model was fitted to linked data from two registers that each cover the same population. However, this development was based on a number of strong assumptions about the linkage process that will typically be violated. One of these assumptions was that all measurements are independent. Obviously, this assumption is unrealistic since measurements are

usually made on clustered of correlated statistical units, such as people in a family, patients in a hospital or students in a school, and when analyzing such data, linear mixed models are often used.

Given the importance of linked data and limitations of the existing statistical analysis methods, the focus of this thesis is on developing methodologies for efficient fitting of linear mixed models to probability-linked data. The statistical inference will be made under two scenarios. The first is where linkage probabilities are known. The second is where they are estimated. Furthermore, since estimation of variance components is also an important objective when fitting a mixed model, appropriate modifications to standard methods of variance components estimation are developed in order to account for linkage error. Lastly, we consider the longitudinal case, where registers are linked across time, and develop methods for fitting linear mixed models to these data that allow for linkage errors.

A review of the literature in Chapter 2 includes a description of a methodological framework that has been used to provide appropriate modifications to standard statistical analysis methods used with probabilistically linked data. In particular, we focus on the case where a linear regression model is applied to data that have been obtained by linking two registers that each cover the same population. This discussion leads to fitting linear mixed models using linked data proposed in Chapter 3. In this chapter, a variety of bias adjusted estimators of the slope and intercept parameters are obtained.

When fitting linear mixed models, the variance components are used to obtain efficient estimates of the regression parameters. In reality, these variance components are unknown and need to be estimated. This leads to Chapter 4 where variance components estimation is developed in order to account for linkage error. In particular, we focus on three widely used methods of variance components estimation: Analysis of Variance (ANOVA), maximum likelihood (ML) and restricted maximum likelihood (REML).

In Chapter 5, an application of these methods to longitudinal modeling is illus-

trated. In this chapter we focus on a situation where registers are linked over time to create longitudinal data. Then simulation results comparing all estimators are described in Chapter 6. These results are for two scenarios: known linkage probabilities and unknown linkage probabilities. They illustrate the relative biases and relative root mean squared errors of these estimators, as well as the actual coverages of the nominal 95% confidence intervals for the model parameters generated by their associated estimators of variance. Lastly, Chapter 7 contains the conclusions for the research described in the preceding chapters, as well as a discussion of outstanding issues that need further research.

Chapter 2

Literature Review

2.1 Introduction

As mentioned in the previous chapter, linkage errors are a particular type of measurement error, and can lead to biased inference and therefore affect the power to detect important changes and relationships. Although there have been well established statistical methods for linking data sets, most of them have mainly focused on the confidentiality issues that arise as a consequence of linkage [7]. There are only few studies that focus on the impact of linkage errors on analysis of linked data such as Neter *et al.* [48], Scheuren and Winkler [58], Lahiri and Larsen [38], and Chambers [7]. In this chapter, methods for linking data, the characteristics of measurement error, its effect on statistical analysis, and measurement error models are described, and research on the impact of linkage errors is reviewed.

2.2 Methods for Linking Data

In the following, we review the techniques used and the practical steps required for record linking as described in Holman [32]. Also, the widely used model for record linkage described in Fellegi and Sunter [17] is reviewed.

2.2.1 Techniques Used in Record Linkage

In general, there are two techniques that are employed in the matching phase of record linkage:

- **Unique Matching or Deterministic Matching:** Unique identifiers such as an insurance number which match exactly are used for linkage. The files of records are sorted into the same order, and two records that have identical unique identifiers are matched.

- **Probability Matching:** This technique is one in which decision rules for record linkage are built into a software package based on the probability of two records being from different people having the same identifier. These probabilities are aggregated into a score and checked against a threshold to determine whether a match is made.

In the Scandinavian countries such as Sweden, Norway, Denmark and so on where unique identifiers for all citizens and residents always exist, record linkage is based on deterministic method most of the time and more than 99% of true matches could be identified [40]. On the other hand, this process may only identify 80-85% of the true matches in some countries where unique identifiers do not always necessarily exist such as Australia, Canada or United States [32, 40]. In theses countries, record linkage much rely on probabilistic method which typically identifies 95-99% of true matches, while 1-2% of matches are false positives.

In Scotland, England, and Japan where good data quality and some personal identifiers are available, record linkage can be conducted using probabilistic and deterministic methods with high degree of success and accuracy [40].

2.2.2 The Practical Steps of Record Linkage

Before commencing the practical steps of record linkage, data preparation is necessary [32]. This involves searching for and correcting invalid values in the data fields;

frequency listing and cross-checking; recoding and transforming data into a standard format; and production of phonetic codes from names using name compression algorithms. That is, subjects with the same name pronunciations such as "Jeff" and "Geoff" are matched together. Then record linkage proceeds as follows:

1. **Blocking:** This step is to increase the efficiency of searching for matches. For unique matching, blocking simply involves sorting the files into the same order of the unique identifier. For probability matching, use is typically made of multiple blocking (multiple passes), such as by distinct values derived from a name compression algorithm and by date of birth. The blocking strategies for each pass should be independent to the extent possible. The fields that are best suited for blocking are those that best discriminate between records that definitely come from different individuals.

2. **Matching:** In this step, potentially linkable pairs of records are systematically compared against all other candidate records that are in the same block to determine whether or not they relate to the same person. In the matching process, some information such as gender, marital status or date of birth is provided in each matching field to determine the status of the pair being examined.

3. **Linking:** Matched records are brought together and assembled in such a way that they can be analyzed as a composite record for one individual. This step includes validation checks, error detection and correction. It also includes the development and use of systems for the storage and retrieval of links for use in future data extraction.

2.2.3 The Fellegi-Sunter Model of Record Linkage

Record linkage as a way of creating data sets for analysis of relationships is now widespread [17, 49, 50, 62, 64]. Newcombe and Kennedy [49] and Newcombe *et al.* [50] describe many ways of record linkage without training data [29]. Their ideas

were formalised by Fellegi and Sunter (FS) [17] in form of a mathematical model. That is, suppose there are two populations A and B whose elements are matched i.e.

$$A \times B = \{(a, b); a \in A, b \in B\}$$

is the union of two disjoint sets

$$M = \{(a, b); a = b, a \in A, b \in B\}$$

and

$$U = \{(a, b); a \neq b, a \in A, b \in B\}$$

which are the sets of true matches and true nonmatches respectively.

The first step to link the records of the two files is the comparison of records. The set of outcomes that can result as a consequence of comparing two records is formally defined as the *comparison vector* γ and the set of all possible realizations of γ or *comparison space* is denoted by Γ. For instance, the result of comparing two records can be statements such as "name is the same", "name disagrees", "name missing on one record", etc. Once γ is observed, we have to decide whether (a, b) is a matched pair $(a, b) \in M$ (denoted by A_1) or (a, b) is an unmatched pair $(a, b) \in U$ (denoted by A_3). However, in some cases it is difficult to make decisions so that we allow a third decision as a possible match (denoted by A_2) and hold for clerical review.

The procedure described above leads to a linkage rule L which can be formally defined as a mapping from Γ onto a set of random decision functions $D = \{d(\gamma)\}$ where

$$d(\gamma) = \{\Pr(A_1 \mid \gamma), \Pr(A_2 \mid \gamma), \Pr(A_3 \mid \gamma)\};\ \gamma \in \Gamma$$

and

$$\sum_{i=1}^{3} \Pr(A_i \mid \gamma) = 1.$$

We see that there are two types of error associated with a linkage rule. The first

occurs when the decision A_1 is made where the members of the comparison pair are in fact unmatched. The probability of this error is defined as

$$\Pr(A_1 \mid U) = \sum_{\gamma \in \Gamma} u(\gamma) \cdot \Pr(A_1 \mid \gamma) \tag{2.1}$$

where $u(\gamma) = \Pr\{\gamma \mid (a, b) \in U\}$.

The second occurs when the decision A_3 is made where the members of the comparison pair are in fact matched. The probability of this error is defined as

$$\Pr(A_3 \mid M) = \sum_{\gamma \in \Gamma} m(\gamma) \cdot \Pr(A_3 \mid \gamma) \tag{2.2}$$

where $m(\gamma) = \Pr\{\gamma \mid (a, b) \in M\}$. Note that if one can estimate the probability of the second error defined in (2.2), one can also obtain the probability of correct linkage from $\Pr(A_3^c \mid M) = 1 - \Pr(A_3 \mid M)$ which is used as a key parameter in the methodological framework of Chambers [7] to analyze the linked data.

FS then considered ratios of probabilities of the form

$$R = m(\gamma)/u(\gamma) \tag{2.3}$$

and the decision rule of FS is summarized by Herzog *et al.* [29] as:

If $R > T_\mu$, then designate pair as a match.

If $T_\lambda \leq R \leq T_\mu$, then designate pair as a possible match and hold for clerical review.

If $R < T_\lambda$, then designate pair as a nonmatch.

The cutoff thresholds T_μ and T_λ are defined by a priori error bounds on false matches and false nonmatches i.e.

$$T_\mu = \frac{m(\gamma_n)}{u(\gamma_n)}$$
$$T_\lambda = \frac{m(\gamma_{n'})}{u(\gamma_{n'})}$$

where $1 < n < n' - 1 < N_\Gamma$ and N_Γ is the number of points in Γ. We see that this rule is reasonable. If γ consists primarily of agreements, then it is intuitive that γ is more likely to occur among matches than nonmatches and ratio (2.3) will be large. On the other hand, if γ consists primarily of disagreements, then the ratio (2.3) will be small [29].

The optimality of this rule was first proved by FS. In general, they proved that their theorem holds for any γ over the set of pairs in the product space $\mathbf{A} \times \mathbf{B}$ from two files. Furthermore, they observed that the quality of the results from the rule depends on the accuracy of the estimates of $m(\gamma)$ and $u(\gamma)$.

In most situations of record linkage, however, typographical error can result in poor matching performances [29]. For example, "Abell" instead "Abel" due to the duplication of the `L` key can result in incorrect linkage since they can refer to different individuals. The FS model has therefore been extended to allow string comparisons. The idea of this method is that we need to have a function that represents approximate agreement. Let's say 1 represents agreement and numbers between 0 and 1 represent partial agreement. Consequently, the likelihood ratios (2.3) needs to be adjusted according to the partial agreement. Winkler [71] showed that a variant of the string comparator introduced by Jaro [35] dramatically improves matching efficacy in comparison with situations when string comparators are not used. The empirical study in Herzog *et al.* [29] also demonstrated that a dramatic improvement in matches can occur when string comparators are first used.

2.2.4 Parameter Estimation Methodology

A method for computing the probabilities in ratio (2.3) was first given by FS. The computation can be simplified by the conditional independence assumption (CI) i.e.

$$m(\gamma) = m_1(\gamma^1) \cdot m_2(\gamma^2) \cdots m_K(\gamma^K)$$

and

$$u(\gamma) = u_1(\gamma^1) \cdot u_2(\gamma^2) \cdots u_K(\gamma^K)$$

where K is number of fields.

In general, however, with a two state comparison agree/disagree, one can obtain such estimates via the EM algorithm. Since the EM is a local maximization algorithm, the starting probabilities may need to be chosen with care based on experience with similar types of files. If the EM algorithm is applied with care, then it will generally yield good parameter estimates [29]. Winkler [70, 72] showed that the EM algorithm yielded optimal parameters in the sense of effective local maxima of the likelihood. It also has robustness and convergence properties under condition (CI).

2.3 Measurement Error

Measurement error is defined to be the difference between the value of a characteristic provided by the respondent and the true (unknown) value of that characteristic [4, 26]. A substantial literature exists on measurement error in sample surveys. See Biemer *et al.* [4] and Lyberg *et al.* [39] for reviews of important measurement error issues.

Measurement error has serious effects on inference i.e. it can give rise to both bias and variance in a survey estimate. Palca [52] explains how measurement error may be important in the interpretation of large scale health studies. Peychev *et al.* [53] report that abortion prevalence has been found to suffer from measurement error and when this error is linked with unit nonresponse it becomes a particularly critical problem for both researchers interested in studying abortion rates and survey research practitioners. Chua and Fuller [10] have also shown that measurement error can cause biases of 100% in estimates of gross change for labor status. Biemer *et al.* [4] illustrate how the measurement error affects a multivariate problem. They concluded that a variable whose theoretical coefficient is zero often shows a significant

coefficient in a regression containing independent variables measured with error.

Measurement error is practically unavoidable, no data generating process can ever be measured without error [9]. Thus, it is essential to understand the major causes of measurement errors in order to control them. Chesher [8] stated that it is necessary to understand the distortions influenced by measurement error in order to understand the effect of measurement error on the information produced by statistical procedures.

Several general approaches for studying measurement error have been suggested in the literature. One approach compares the survey responses with potentially more accurate data from another source. The most common approach of assessing measurement error is the use of a "record check study" [4] which is usually assumed that one of the sources is the "golden standard" for the other(s). Such a study generally assumes that information contained in the records is without error. Record check studies are usually employed to estimated measurement errors that are conceptualized as biases. They are often used to compare measurement procedures to see which will match the record data most closely. That procedure is judged to have the lowest measurement bias.

Various statistical techniques have been used to correct for the bias due to the effect of measurement error. Fuller [21] has described a number of methods for correcting measurement error bias that are in practical use. Some of these methods will be reviewed in the following subsections.

2.3.1 Ordinary Least Squares and Measurement Error

Consider the classical linear regression model with one independent variable

$$Y_t = \beta_0 + \beta_1 x_t + e_t; \quad t = 1, ..., n, \tag{2.4}$$

where x_t are assumed to be independent drawings from a $N(\mu_x, \sigma_x^2)$ distribution and the e_t are independent $N(0, \sigma_e^2)$ random variables. It is well known that the least

squares estimator

$$\hat{\beta}_1 = \left\{ \sum_{t=1}^{n} (x_t - \bar{x})^2 \right\}^{-1} \sum_{t=1}^{n} (x_t - \bar{x})(Y_t - \bar{Y})$$

is unbiased for β_1 and has the smallest variance of unbiased linear estimators. However, in some situations one can not observe x_t directly. Instead, one observes the sum

$$X_t = x_t + u_t \tag{2.5}$$

where u_t is a $(0, \sigma_u^2)$ random variable. That is, u_t has mean 0 and variance σ_u^2. An example of this situation has been given by Fuller [21] in considering the relationship between the yield of corn and available nitrogen in the soil. Suppose that (2.4) is appropriate to model the relationship between yield and nitrogen. To estimate the available nitrogen in the soil, we need to sample the soil of the experimental plot and perform a laboratory analysis on the selected sample. Therefore, we do not observe x_t but observe an estimate of x_t due to a result of the sampling and laboratory analysis. That is, X_t represents the observed nitrogen where u_t is the measurement error introduced by sampling and laboratory analysis.

To investigate the effect of measurement error on the least squares coefficient in the model (2.4) and (2.5), x_t are assumed to be random variables and the regression coefficient computed from the observed variables becomes

$$\hat{\beta}_{1Ob} = \left\{ \sum_{t=1}^{n} (X_t - \bar{X})^2 \right\}^{-1} \sum_{t=1}^{n} (X_t - \bar{X})(Y_t - \bar{Y})$$

where the vector $(Y_t, X_t)^\top$ is distributed as a bivariate normal vector with mean vector

$$E\{(Y, X)\} = (\mu_Y, \mu_X) = (\beta_0 + \beta_1 \mu_x, \mu_x)$$

and covariance matrix

$$\begin{bmatrix} \sigma_Y^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_X^2 \end{bmatrix} = \begin{bmatrix} \beta_1^2\sigma_x^2 + \sigma_e^2 & \beta_1\sigma_x^2 \\ \beta_1\sigma_x^2 & \sigma_x^2 + \sigma_u^2 \end{bmatrix}. \tag{2.6}$$

Thus,

$$E(\hat{\beta}_{1Ob}) = (\sigma_X^2)^{-1}\sigma_{XY} = \beta_1(\sigma_x^2 + \sigma_u^2)^{-1}\sigma_x^2. \tag{2.7}$$

It can be seen that the least squares regression coefficient is biased toward zero under the assumption that the measurement error in X_t is independent of the true values, x_t, and of the errors, e_t. In other words, the regression coefficient has been attenuated by the measurement error.

It has been shown in (2.7) that the expected value of the least squares estimator $\hat{\beta}_{1Ob}$ is the true β_1 multiplied by the *reliability ratio* $\kappa_{xx} = (\sigma_X^2)^{-1}\sigma_x^2$. Fuller [21] suggested that one method of correcting for this bias is that if the ratio κ_{xx} is known, an unbiased estimator of β_1 is given by

$$\hat{\beta}_1 = \kappa_{xx}^{-1}\hat{\beta}_{1Ob}.$$

2.3.2 Measurement Variance Known

Recall that we assume

$$Y_t = \beta_0 + \beta_1 x_t + e_t, \quad X_t = x_t + u_t, \tag{2.8}$$

$$(x_t, e_t, u_t)^\top \sim N\{(\mu_x, 0, 0)^\top, \mathrm{diag}(\sigma_x^2, \sigma_e^2, \sigma_u^2)\}.$$

The population moments of (Y_t, X_t) satisfy

$$(\sigma_Y^2, \sigma_{XY}, \sigma_X^2) = (\beta_1^2\sigma_x^2 + \sigma_e^2, \beta_1\sigma_x^2, \sigma_x^2 + \sigma_u^2)$$

and

$$(\mu_Y, \mu_X) = (\beta_0 + \beta_1\mu_x, \mu_x). \tag{2.9}$$

If it is assumed that the variance of the measurement error, σ_u^2, is known and the sample mean and sample covariances of (Y_t, X_t) are determined, another method to create an unbiased estimator proposed by Fuller [21] is to replace the unknown population moments with their sample estimators to obtain a system of equations in the unknown parameters. That is, we replace the left side of (2.9) with $s_Y^2, s_{XY}, s_X^2, \bar{Y}$ and $\bar{X}$, respectively. Solving, we have

$$\begin{aligned} \hat{\beta}_1 &= (s_X^2 - \sigma_u^2)^{-1} s_{XY}, \\ (\hat{\sigma}_x^2, \hat{\sigma}_e^2) &= (s_X^2 - \sigma_u^2, s_Y^2 - \hat{\beta}_1 s_{XY}), \\ (\hat{\mu}_x, \hat{\beta}_0) &= (\bar{X}, \bar{Y} - \hat{\beta}_1 \bar{X}). \end{aligned} \tag{2.10}$$

Note that $\hat{\beta}_1$ is a ratio of two random variables and such ratios are typically biased estimators of the ratio of the expectations. Thus, Fuller [21] modified the estimator (2.10) to produce an estimator that is nearly unbiased for β_1. This estimator is given by

$$\tilde{\beta}_1 = \{\hat{H}_x^2 + \alpha(n-1)^{-1}\sigma_u^2\}^{-1} s_{XY},$$

where $\alpha > 0$ is a fixed number to be determined,

$$\hat{H}_x^2 = \begin{cases} s_X^2 - \sigma_u^2 & \text{if } \hat{\lambda} \geq 1 + (n-1)^{-1} \\ s_X^2 - \{\hat{\lambda} - (n-1)^{-1}\}\sigma_u^2 & \text{if } \hat{\lambda} < 1 + (n-1)^{-1}, \end{cases}$$

and $\hat{\lambda}$ is the root of

$$| \, \mathbf{s}_Z^2 - \lambda \, \mathrm{diag}(0, \sigma_u^2) \, | = 0$$

where $\mathbf{s}_Z^2 = (n-1)^{-1} \sum_{t=1}^{n} (\mathbf{Z}_t - \bar{\mathbf{Z}})^\top (\mathbf{Z}_t - \bar{\mathbf{Z}})$ and $\mathbf{Z}_t = (Y_t, X_t)$.

The generalization of model (2.8) to a model containing a vector of x variables is

$$Y_t = \beta_0 + \mathbf{x}_t \boldsymbol{\beta}_1 + e_t, \quad \mathbf{X}_t = \mathbf{x}_t + \mathbf{u}_t, \tag{2.11}$$

where

$$\begin{bmatrix} \mathbf{x}_t^\top \\ e_t \\ \mathbf{u}_t^\top \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu}_x^\top \\ 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_e^2 & \boldsymbol{\Sigma}_{eu} \\ \mathbf{0} & \boldsymbol{\Sigma}_{ue} & \boldsymbol{\Sigma}_{uu} \end{bmatrix} \right).$$

Assume that $\boldsymbol{\Sigma}_{uu}$ and $\boldsymbol{\Sigma}_{ue}$ are known, the maximum likelihood estimators adjusted for degrees of freedom are

$$\begin{aligned} \hat{\boldsymbol{\beta}}_1 &= (s_X^2 - \boldsymbol{\Sigma}_{uu})^{-1}(s_{XY} - \boldsymbol{\Sigma}_{ue}), \\ (\hat{\boldsymbol{\mu}}_x, \hat{\beta}_0) &= (\bar{\mathbf{X}}, \bar{Y} - \bar{\mathbf{X}}\hat{\boldsymbol{\beta}}_1), \\ \sigma_e^2 &= s_Y^2 - 2s_{YX}\hat{\boldsymbol{\beta}}_1 + \hat{\boldsymbol{\beta}}_1^T s_X^2 \hat{\boldsymbol{\beta}}_1 + 2\boldsymbol{\Sigma}_{eu}\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_1^T \boldsymbol{\Sigma}_{uu}\boldsymbol{\beta}_1. \end{aligned}$$

and $\hat{\boldsymbol{\Sigma}}_{xx} = \mathbf{s}_X^2 - \boldsymbol{\Sigma}_{uu}$.

2.3.3 Ratio of Measurement Variances Known

Given information on the relative magnitude of the error variances associated with the two variables in the model, we rewrite model (2.4) as

$$\begin{aligned} y_t &= \beta_0 + \beta_1 x_t, \\ (Y_t, X_t) &= (y_t, x_t) + (e_t, u_t), \end{aligned} \tag{2.12}$$

where (Y_t, X_t) is observed, y_t is the true value of the dependent variable, x_t is the true value of the independent variable, and (e_t, u_t) is the vector of measurement error. If the ratio

$$\delta = (\sigma_u^2)^{-1}\sigma_e^2$$

is known, using the method of moment yields the estimator

$$\hat{\beta}_1 = (\hat{\sigma}_x^2)^{-1} s_{XY} \tag{2.13}$$

where

$$\hat{\sigma}_x^2 = (2\delta)^{-1}\left[\{(s_Y^2 - \delta s_X^2)^2 + 4\delta s_{XY}^2\}^{1/2} - (s_Y^2 - \delta s_X^2)\right].$$

The extension of the model (2.12) to a model with a vector of explanatory variables is

$$\begin{aligned} y_t &= \mathbf{x}_t\boldsymbol{\beta}, \\ (Y_t, \mathbf{X}_t) &= (y_t, \mathbf{x}_t) + (e_t, \mathbf{u}_t) \end{aligned} \tag{2.14}$$

for $t = 1, ..., n$, where $\{\mathbf{x}_t\}$ is a sequence of fixed $k-$dimensional row vectors. We can also write the defining equations (2.14) as

$$\mathbf{Z}_t = \mathbf{z}_t + \boldsymbol{\varepsilon}_t,$$

where $\mathbf{Z}_t = (Y_t, \mathbf{X}_t)$, $\mathbf{z}_t = (y_t, \mathbf{x}_t)$, and $\boldsymbol{\varepsilon}_t = (e_t, \mathbf{u}_t)^\top$ is the vector of measurement errors. Following Theorem 2.3.1 of Fuller [21], let $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\varepsilon\varepsilon})$, where $\boldsymbol{\Sigma}_{\varepsilon\varepsilon} = \boldsymbol{\Upsilon}_{\varepsilon\varepsilon}\sigma^2$ and $\boldsymbol{\Upsilon}_{\varepsilon\varepsilon}$ is known. Then the maximum likelihood estimator of $\boldsymbol{\beta}$ and σ^2 are

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{S}_X^2 - \hat{\lambda}\boldsymbol{\Upsilon}_{uu})^{-1}(\mathbf{S}_{XY} - \hat{\lambda}\boldsymbol{\Upsilon}_{ue}), \\ \hat{\sigma}^2 &= (k+1)^{-1}\hat{\lambda}, \end{aligned}$$

where $\mathbf{S}_Z^2 = n^{-1}\sum_{t=1}^n \mathbf{Z}_t^\top \mathbf{Z}_t$ and $\hat{\lambda}$ is the smallest root of

$$|\ \mathbf{S}_Z^2 - \lambda\boldsymbol{\Upsilon}_{\varepsilon\varepsilon}\ | = 0.$$

2.3.4 Instrumental Variable Estimation

In this subsection the use of a different type of auxiliary information is considered. Following model (2.4) and (2.5), suppose we have a third variable available, denoted by W_t which is correlated with x_t and both u_t and e_t are independent of W_t. For example, we might conduct an experiment where X_t is the observed nitrogen in the

leaves of the plant and Y_t is the dry weight of the plant. We would expect the true nitrogen in the leaves, x_t, to be correlated with nitrogen fertilizer, W_t.

Fuller [21] suggested that it would be convenient to have parametric expression for the fact that x_t and W_t are related. By using the parameters of the population regression of x_t on W_t to quantify the relationship, let

$$\pi_{22} = \left[E\left\{ \sum_{t=1}^{n} (W_t - \bar{W})^2 \right\} \right]^{-1} E\left\{ \sum_{t=1}^{n} (W_t - \bar{W}) x_t \right\}, \tag{2.15}$$

$$\pi_{12} = E(\bar{x} - \pi_{22}\bar{W}). \tag{2.16}$$

It is assumed that

$$\begin{bmatrix} x_t \\ e_t \\ u_t \\ W_t \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_x \\ 0 \\ 0 \\ \mu_W \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \sigma_{xe} & \sigma_{xu} & \sigma_{xW} \\ \sigma_{xe} & \sigma_e^2 & \sigma_{eu} & 0 \\ \sigma_{xu} & \sigma_{eu} & \sigma_u^2 & 0 \\ \sigma_{xW} & 0 & 0 & \sigma_W^2 \end{bmatrix} \right),$$

where $\mu_x = \pi_{12} + \pi_{22}\mu_W$ and $\sigma_{xW} = \pi_{22}\sigma_W^2 \neq 0$.

Note that the ratio of covariances

$$\sigma_{XW}^{-1}\sigma_{YW} = (\pi_{22}\sigma_W^2)^{-1}\beta_1\pi_{22}\sigma_W^2 = \beta_1.$$

It follows that we can estimate β_1 and β_0 by

$$\hat{\beta}_1 = s_{XW}^{-1} s_{YW}, \tag{2.17}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \tag{2.18}$$

where $(\bar{Y}, \bar{X}) = n^{-1} \sum_{t=1}^{n} (Y_t, X_t)$ and

$$(s_{YW}, s_{XW}) = (n-1)^{-1} \sum_{t=1}^{n} (Y_t - \bar{Y}, X_t - \bar{X})(W_t - \bar{W}).$$

Since the sample moments are consistent estimators of the population moments,

the estimators of β_1 and β_0 will be consistent under the model assumptions. The assumption that $\sigma_{xW} \neq 0$ is critical. If $\sigma_{xW} = 0$, the denominator of $\hat{\beta}_1$ defined in (2.17) is estimating zero and $\hat{\beta}_1$ is not a consistent estimator of β_1. The less restrictive assumptions can be found in Theorem 1.4.1 of Fuller [21].

The extension of this method to equations with several explanatory variables is

$$Y_t = \mathbf{x}_t\boldsymbol{\beta} + e_t, \quad \mathbf{X}_t = \mathbf{x}_t + \mathbf{u}_t \tag{2.19}$$

for $t = 1, .., n$, where $\mathbf{x}_t$ is a $k-$dimensional row vector of explanatory variables, $(Y_t, \mathbf{X}_t)$ is observed, and $\mathbf{u}_t$ is the vector of measurement errors. Let a $q-$dimensional vector of instrumental variables, denoted by $\mathbf{W}_t$, be available and assume that $n > q \geq k$. If n observations are available, we can write the model in matrix notation as

$$\mathbf{Y} = \mathbf{x}\boldsymbol{\beta} + \mathbf{e}, \quad \mathbf{X} = \mathbf{x} + \mathbf{u}$$

where Y is the $n-$dimensional column vector of observations on Y_t, and $\mathbf{X} = (\mathbf{X}_1^\top, ..., \mathbf{X}_n^\top)^\top$ is the $n \times k$ matrix of observations on $\mathbf{X}_t$.

If $q = k$, one can construct an estimator for $\boldsymbol{\beta}$ as

$$\hat{\boldsymbol{\beta}} = (\mathbf{W}^\top\mathbf{X})^{-1}\mathbf{W}^\top\mathbf{Y}$$

which is the matrix generalization of expression (2.17).

2.3.5 Linkage Error Is a Different Type of Measurement Error

Measurement error issues in linking data may arise for a number of reasons. Chesher and Nesheim [9] described that measurement error can result from incorrectly linked unit so that data from one unit is incorrectly associated with another unit. Measurement error caused by this issue is also called “linkage error”.

From what we developed so far in the previous subsections, we see that the measurement error models were based on equation (2.14). That is

$$Y_t = y_t + e_t \quad \text{and} \quad X_t = x_t + u_t$$

where y_t and x_t are the true values, e_t is regression error, and u_t is measurement error.

It is obvious that measurement error was assumed to occur on X_t. In data linking context, however, we observe x_t directly but measurement error occurs in the linked data Y_t i.e. $Y_t = y_t + e_t$ where y_t is the true value which can be completely different to Y_t, and e_t is linkage error. We see that e_t in a data linking context has a different meaning to the way it is used in the measurement error models considered in the previous subsections. There it represented regression error. Here it represents linkage error. As a result, these models can not be applied directly to linked data.

In the second half of this thesis we will consider the case where Y_t and X_t are observed without error and are linked with records for the same unit at different times and held in different registers to create longitudinal linked data. In particular, we consider the situation where there is linkage error associated with the value of the grouping variable held on each register, so that data from different units at different times are linked. In this situation, the measurement error models in Fuller [21] can not be applied directly since these models do not allow for measurement error in the grouping structure of the data.

2.4 Research on the Impact of Linkage Errors

The impact of linkage errors was first investigated by Neter *et al.* [48]. In their study, the implications of linkage errors were made for two simplified models in order to study the effect of such errors on the analysis of response errors. The first model assumes that each unit in the sample has the same probability of correct linkage. Also, if a linkage occurs the model assumes that each unit has an equal

probability of incorrect linkage. Whereas the second model assumes that linkages occur according to a random mechanism within subsets of the parent population and permits the probability of correct linkage to vary among subsets. Because of these two assumptions, the second model was considered to be more realistic and flexible. In addition, for the simplicity of the analysis, they also assumed that the linkage was complete i.e. there were no non-links in the sample.

In considering the first model, they defined Y as a response variable and a random variable Z_j for Y_j $(j = 1, ..., N)$ as follows:

$$Z_j = \begin{cases} Y_j \text{ with probability } p \\ \\ Y_k \text{ with probability } q \ (k \neq j) \end{cases} \tag{2.20}$$

where p is the probability of correct linkage and q is the probability of incorrect linkage. Thus, it follows that

$$p + (N - 1)q = 1 \tag{2.21}$$

where N is the sample size.

However, in reality the probability of a correct linkage may not be the same for the entire population, nor that linkage errors can occur throughout the population. Instead, such errors may occur within subsets of the population, such as persons with the same name and age, persons at the same address, or persons in a household. Moreover, the probability of correct linkage may differ subset to subset. These limitations lead to the second model which assumes that:

- The population is divided into K mutually exclusive and exhaustive subsets.
- Linkage errors occur within a subset.
- Within the ith subset, containing N_i elements, the probability of a correct linkage for any element is p_i, and the probability of an incorrect linkage for

any element is q_i. Thus,

$$p_i + (N_i - 1)q_i = 1 \tag{2.22}$$

One of their conclusions was that relatively small linkage error could lead to a substantial bias in estimating the relationship between response errors and true values.

Scheuren and Winkler [58] then investigated the effect of linkage errors on the bias of ordinary least squares estimators of regression coefficients in a standard regression model and proposed a method of adjusting for the bias. That is, for known linkage probabilities, they obtained their estimator of regression coefficients by adjusting the bias of the ordinary least squares estimator for the regression model with linkage errors. However, their estimator is not unbiased in general. The best performance of this estimator still produces a very small bias. Therefore, Lahiri and Larsen [38] proposed an alternative method for the bias correction which provides an unbiased estimator directly for a transformed regression model. In their simulations, they found that their unbiased method performed very well across a range of situations.

Chambers [7] has then made a significant contribution to this area. In gathering all of the previous studies, he described an inferential framework for statistical modeling using probabilistically linked data, which is then used to develop modified estimation methods for regression models. Here we summarize the linkage error model underpinning the development in Chambers [7].

We assume that there is a population of N units, indexed by $i = 1, ..., N$. For each unit in this population, there are the measurement values of a scalar random variable Y and a vector random variable $\mathbf{X}$. The aim is to model the relationship between Y and $\mathbf{X}$ in this population, and in particular to fit a linear regression model where the regression coefficients are unknown and need to be estimated. However, there is no single database that contains the values of Y and $\mathbf{X}$ from this population. Instead, one can obtain such values from two registers that separately contain the population values of Y and $\mathbf{X}$ and both registers refer to the same population and have no duplicates. That is, each register consists of N records.

In order to link records from the two registers, one needs a unique identifier in each unit in the population. However, in a number of countries such identifier does not exist. Instead, some form of probability-linking is used to link records from the two registers. We also assume that linkage is complete and one to one between the Y and $\mathbf{X}$-registers. Clearly, this type of linked data set can contain linkage errors, i.e. records where the values of Y and $\mathbf{X}$ actually come from different population units.

In addition, it is assumed that the linked records can be partitioned into Q distinct blocks such that there is no possibility that linked records in different blocks contain data for the same population unit. This assumption is based on the fact that probability linking algorithms will only attempt to link records that are similar in some sense. That is, there is a categorical population variable F with values that can be obtained from the information on either register and that different blocks correspond to different values of F. In other words, if a record on one register does not have the same value of F as the record on the other register, then two records cannot be the same unit in the population. The variable F is referred to as a blocking variable below, and it is assumed that all population units with the same value of F are in the same block. Furthermore, we also assume that F is measured without error on both the Y-register and the $\mathbf{X}$-register, which implies that linkage errors can only occur within the same block.

Without loss of generality, we assume that F takes the Q distinct values 1,...,Q and let block q correspond to the M_q population units with $F = q$ such that $N = \sum_q M_q$. Let i index records in the $\mathbf{X}$-register. Then y_i^* is used to denote the Y-value from block q on the Y-register that is matched to $\mathbf{X}_i$ in block q on the $\mathbf{X}$-register i.e. there are M_q linked data pairs $(y_i^*, \mathbf{X}_i)$ in block q. Thus, $\mathbf{y}_q^*$ is used to denote the vector of order M_q of the linked values y_i^* in block q and $\mathbf{X}_q$ as the matrix with rows defined by the values $\mathbf{X}_i$ in the same block. Lastly, we use $\mathbf{y}_q$ to denote the unknown vector of the true Y values in block q that are associated with $\mathbf{X}_q$.

Since the linkage is assumed to be complete and one to one between the Y and

X-registers, randomness in the outcome of the linkage process is modeled via the identity

$$\mathbf{y}_q^* = \mathbf{A}_q \mathbf{y}_q \tag{2.23}$$

where $\mathbf{A}_q = [a_{ij}^q]$ is an unknown random permutation matrix of order M_q i.e. the entries a_{ij}^q of $\mathbf{A}_q$ are either zero or one, with a value of one occurring just once in each row and column. In addition, due to the assumption that linkage errors can occur within blocks, then $\mathbf{A}_{q_1}$ and $\mathbf{A}_{q_2}$ are independently distributed when $q_1 \neq q_2$.

Furthermore, it is assumed that linkage is *non-informative* at each level of F so that the distribution of $\mathbf{A}_q$ is independent of $\mathbf{y}_q$ given $\mathbf{X}_q$. Let

$$E(\mathbf{A}_q \mid \mathbf{X}_q) = \mathbf{E}_q. \tag{2.24}$$

Neter *et al.* [48] characterized the distribution of linkage errors via an exchangeable linkage errors model, where for each value of q

$$\Pr(\text{correct linkage}) = \Pr(a_{ii}^q = 1) = \lambda_q \tag{2.25}$$

and, for $i \neq j$,

$$\Pr(\text{incorrect linkage}) = \Pr(a_{ij}^q = 1) = \gamma_q. \tag{2.26}$$

Given (2.25) and (2.26) hold, (2.24) is then of the form

$$\mathbf{E}_q = (\lambda_q - \gamma_q)\mathbf{I}_q + \gamma_q \mathbf{1}_q \mathbf{1}_q^\top \tag{2.27}$$

where $\mathbf{I}_q$ is the identity matrix of order M_q and $\mathbf{1}_q$ denotes a vector of ones of length M_q. Since $\mathbf{1}_q^\top \mathbf{A}_q = \mathbf{1}_q^\top$ and $\mathbf{A}_q \mathbf{1}_q = \mathbf{1}_q$ thus, $\mathbf{1}_q^\top \mathbf{E}_q = \mathbf{1}_q^\top$ and $\mathbf{E}_q \mathbf{1}_q = \mathbf{1}_q$. That is, (2.27) implies

$$\begin{aligned} \lambda_q + (M_q - 1)\gamma_q &= 1 \\ \gamma_q &= \frac{1 - \lambda_q}{M_q - 1}. \end{aligned} \tag{2.28}$$

Obviously, λ_q is the key parameter to completely specify the first order properties of the linkage mechanism under the model (2.27). All of these properties and models will be used throughout this thesis.

Linear regression model is then the focus of inference. That is, for the population values of Y and $\mathbf{X}$ in each block, we define

$$E_x(\mathbf{y}_q) = \mathbf{X}_q\boldsymbol{\beta} = \mathbf{f}_q \tag{2.29}$$

$$Var_x(\mathbf{y}_q) = \sigma^2\mathbf{I}_q \tag{2.30}$$

where a subscript of X denotes conditioning on the value $\mathbf{X}_q$ of the explanatory variables in block q and the regression parameter $\boldsymbol{\beta}$ in (2.29) is the target of inference. Then, four methods to obtain the estimators of regression coefficients have been proposed. Firstly, an estimator of regression coefficients is obtained by adjusting the bias of the ordinary least squares (OLS) estimator for the regression model with linkage errors. Secondly, following Lahiri and Larsen [38] yields the OLS estimator from directly modeling the regression relationship between the linked values $\mathbf{y}_q^*$ and the values in $\mathbf{X}_q$. Thirdly, the maximum likelihood estimator by using the Missing Information Principle (MIP) given the linked data is obtained. Lastly, the y_q known OLS estimator is constructed when perfectly linked data is assumed.

We now turn our attention to the second estimator. Note that the optimality of this estimator depends on the regression errors being homoscedastic. However, this condition generally does not hold since the variances of the regression errors defined by the linked data vary between blocks. Thus, this estimator has been developed using the solution of the *optimal estimating equation* to be the best linear unbiased estimator (BLUE) when heteroscedasticity of the variance is accounted for.

In doing so, we extend the ideas developed for linear regression analysis in the beginning to where the regression model of interest is fitted via the solution of an estimating equation. In particular, it is assumed that this model is characterized by

a $p-$dimensional parameter θ, which is then estimated by solving

$$\mathbf{H}(\theta) = \mathbf{0}$$

where $\mathbf{H}(\theta)$ is a $p-$dimensional unbiased estimating function for θ. In particular, when $\mathbf{H}(\theta)$ is of the form

$$\mathbf{H}(\theta) = \sum_q \mathbf{G}_q(\theta)\{\mathbf{y}_q - \mathbf{f}_q(\theta)\} \tag{2.31}$$

where $\mathbf{G}_q(\theta)$ is the $p \times M_q$ matrix with columns defined by the vectors $\mathbf{G}_i(\theta)$ which is a vector of order p and is a function of θ and $\mathbf{X}_i$ but not of y_i, and $\mathbf{f}_q(\theta)$ is the vector of order M_q defined by their corresponding values of $f_i(\theta)$.

Clearly, (2.31) defines an unbiased estimating function for θ_0 which is the true value of θ. However, instead of $\mathbf{y}_q$, we have access to $\mathbf{y}_q^*$. Let $\mathbf{H}^*(\theta)$ denote the value of (2.31) when $\mathbf{y}_q^*$ is used instead of $\mathbf{y}_q$. That is, the naive estimator $\hat{\theta}^*$ of θ_0 that assumes no linkage errors satisfies

$$\mathbf{H}^*(\hat{\theta}^*) = \sum_q \mathbf{G}_q(\hat{\theta}^*)\{\mathbf{y}_q^* - \mathbf{f}_q(\hat{\theta}^*)\} = \mathbf{0}. \tag{2.32}$$

We see that $E_X\{\mathbf{H}^*(\theta_0)\} \neq \mathbf{0}$, i.e. $\mathbf{H}^*(\theta)$ is biased if linkage is not perfect. So, the resulting estimator $\hat{\theta}^*$ is also biased. Given $\mathbf{E}_q$, this bias can be corrected by replacing the estimating function $\mathbf{H}^*(\theta_0)$ by its bias-corrected version

$$\mathbf{H}^*_{adj}(\theta) = \sum_q \mathbf{G}_q(\theta)\{\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q(\theta)\}. \tag{2.33}$$

As far as the estimating function (2.33) is concerned, Chambers [7] claimed that the optimal version of $\mathbf{G}_q(\theta)$ is

$$\mathbf{G}_q^{opt}(\theta) = \partial_\theta \left\{E_x(\mathbf{y}_q^*)\right\} Var_x^{-1}(\mathbf{y}_q^*) \tag{2.34}$$

and the solution of the optimal estimating equation leads to the BLUE estimator

as mentioned before. A proof for Theorem 2.1 is given in Appendix C.

Theorem 2.1 *If $Y_1, ..., Y_T$ are independent random variables with $E(Y_t) = \alpha_t(\theta)$, $Var(Y_t) = \sigma_t^2(\theta)$, the α_t's and σ_t^2's being specified differentiable functions, then $G^* = \sum_{t=1}^{T} \frac{\partial \alpha_t}{\partial \theta} \sigma_t^{-2}(\theta)\{y_t - \alpha_t(\theta)\}$ is an optimal estimating function in the class G_0.*

2.5 Longitudinal Linked Data

Longitudinal linked data are defined as microdata that contain observations from two or more related sampling frame, with measurements from multiple time periods for all units of observation [1]. A wide range of research efforts essentially carry out longitudinal linked data as they provide exceptionally rich source of information to address complex policy issues in key areas such as health care, education, and economic policy [65]. The use of longitudinal linked data has recently become a key research tool in projects commissioned by central and local government in many countries. A number of official statistical agencies have undertaken initiatives to develop longitudinal linked data sets, particularly in Europe and U.S. [65]. This is because in their perspective, longitudinal linked data have great potential to enhance existing official statistics, improve data accuracy and reduce data collection redundancies [41].

In the UK, longitudinal information has traditionally been derived from social surveys or from the Census Longitudinal Study (CLS). For example, Elias [16] reports the use of Higher Education Statistics Agency (HESA) data to create a longitudinal sample of students completing higher education and follow-up into the labor market. In a study of welfare dynamics among lone mothers, Noble *et al.* [51] linked together seven individual-level extracts of Housing Benefit/Council Tax Benefit (HB/CTB) over a three year period in one large local authority which allowed both longitudinal analysis of low income lone mothers.

In Australia, the Australian Bureau of Statistics (ABS) has proposed the Australian Longitudinal Learning Database (ALLD) project as a core enduring database

of education and socio-demographic statistical information [46]. An ALLD will link data on the pathways and outcomes of Australian students from early childhood education to schooling, post-school education and potentially labor force outcomes. Currently, these data are fragmented and there is only limited information on the educational pathways and outcomes for students. This is largely due to the data being collected from a variety of sources and there are currently no accessible database that integrate data sources across the different education sectors for statistical and research purposes.

The ALLD will be constructed using probabilistic linkage techniques. That is, variables such as age, sex, geographic location and other socio-demographic characteristics will be used to match records from one data set to those in another. Therefore, construction of the ALLD could commence immediately without the implementation of a unique student identifier for each Australian student.

An obvious example of creating a longitudinal database was provided by Statistic New Zealand [36]. The Linked Employer-Employee Data (LEED) has been developed by linking data sets held by the Inland Revenue Department (IRD) as administrative tax records with data held by Statistics New Zealand. This data set produces statistics on firm and employment such as job and worker flows, employment tenure, multiple job holding and business demography. In order to achieve such analysis, it is necessary to be able to link employers over time or longitudinally.

Typically, an employer reports their tax using the same unique identifier each month, *employer_ird_number*, so that in the first instance, employers were linked over time by such identifier. Therefore, it was assumed that all values of *employer_ird_number* were valid and correct. In other words, two records identified by the same *employer_ird_number* referred to the same employer. However, there were some cases that an employer cannot be linked over time by its *employer_ird_number*, so that tracking common employees and deterministic verification of potential links were undertaken.

Although several methods were used to maximize the efficiency of linking, in

reality however, two types of error can occur [36]:

- A false negative error occurs when two records that belong to the same entity are not linked.
- A false positive error occurs when two records that refer to different entities are linked in error.

We see that although statistical method for linking data sets longitudinally are well established, statistical research in this area mainly focused on linkage method. As a result, the last part of this thesis will focus on statistical inference of longitudinal linked data.

Chapter 3

Linear Mixed Model Using Linked Data

In Chambers [7], a linear model is fitted using linked data under the assumption that all measurements are independent. In this chapter we consider the situation where a linear mixed model is the focus of inference based on an assumption that complete linkage is conceptually possible i.e. the registers are all the same size and contain the population data for the same N units. We therefore introduce an auxiliary grouping variable Z which takes values $1, ..., G$, and let group g correspond to the N_g population units with $Z = g$ such that $M_q = \sum_g n_{qg}$ and $N_g = \sum_q n_{qg}$ where n_{qg} is the number of population units in block q and group g. We assume throughout this chapter that the values of Z in the linked data are correct, i.e. this variable is stored on the **X**-register.

To illustrate the feature of this data, suppose that the **X**-register is made up of the eight records

$$\mathbf{X} = \left[\begin{array}{cc} 1 & x_1 \\ \hline 1 & x_2 \\ \hline 1 & x_3 \\ \hline 1 & x_4 \\ \hline 1 & x_5 \\ \hline 1 & x_6 \\ \hline 1 & x_7 \\ \hline 1 & x_8 \end{array}\right] = \left[\begin{array}{c} \mathbf{X}_{11} \\ \hline \mathbf{X}_{12} \\ \hline \mathbf{X}_{21} \\ \hline \mathbf{X}_{22} \end{array}\right] = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$$

where $\mathbf{X}_{qg}$ denotes the matrix with rows defined by the values $\mathbf{X}_i$ in block q group g and $\mathbf{X}_q$ denotes the matrix with rows defined by the values $\mathbf{X}_i$ in block q.

Similarly, the $Y-$register is constructed as

$$Y = \left[\begin{array}{c} y_1 \\ \hline y_2 \\ \hline y_3 \\ \hline y_4 \\ \hline y_5 \\ \hline y_6 \\ \hline y_7 \\ \hline y_8 \end{array}\right] = \begin{bmatrix} \mathbf{y}_{11} \\ \mathbf{y}_{12} \\ \mathbf{y}_{21} \\ \mathbf{y}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$$

where $\mathbf{y}_{qg}$ denotes a vector of y_i in block q group g and $\mathbf{y}_q$ denotes a vector of y_i in block q.

After probability linking, we declare the links to be

$$Y^* = \left[\begin{array}{c} y_1 \\ y_3 \\ \hline y_2 \\ y_4 \\ \hline y_5 \\ y_6 \\ \hline y_7 \\ y_8 \end{array}\right] = \left[\begin{array}{c} \mathbf{y}^*_{11} \\ \mathbf{y}^*_{12} \\ \mathbf{y}^*_{21} \\ \mathbf{y}^*_{22} \end{array}\right] = \left[\begin{array}{c} \mathbf{y}^*_{1} \\ \mathbf{y}^*_{2} \end{array}\right]$$

where $\mathbf{y}^*_{qg}$ denotes a vector of y^*_i in block q group g and $\mathbf{y}^*_q$ denotes a vector of y^*_i in block q.

The matrix $\mathbf{A}$ is then given by

$$\mathbf{A} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{cc} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{array}\right].$$

The relationship between the linked values in Y^* and the true values in Y in this example is then characterized by

$$Y^* = \left[\begin{array}{c} \mathbf{y}^*_1 \\ \mathbf{y}^*_2 \end{array}\right] = \mathbf{A}Y = \left[\begin{array}{cc} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{array}\right] \left[\begin{array}{c} \mathbf{y}_1 \\ \mathbf{y}_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{A}_1\mathbf{y}_1 \\ \mathbf{A}_2\mathbf{y}_2 \end{array}\right].$$

3.1 Linear Mixed Models

From the illustration above we see that linkage errors occur within blocks but measurements from different blocks can be in the same group and therefore are not independent. The linear mixed model is then given by

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{3.1}$$

where $\mathbf{X}$ and $\mathbf{Z}$ are the design matrices corresponding to the fixed and random parts of the model respectively,

$$E\begin{bmatrix}\mathbf{u}\\ \mathbf{e}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} \quad \text{and} \quad \text{Cov}\begin{bmatrix}\mathbf{u}\\ \mathbf{e}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\zeta} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}\end{bmatrix}$$

where $\boldsymbol{\zeta} = \sigma_u^2\mathbf{I}_G$ and $\mathbf{R} = \sigma_e^2\mathbf{I}_N$. Thus, the variance-covariance matrix of $\mathbf{Y}$ is of the form

$$\mathbf{V} = \mathbf{Z}\boldsymbol{\zeta}\mathbf{Z}^\top + \mathbf{R} = \sigma_u^2\mathbf{Z}\mathbf{Z}^\top + \sigma_e^2\mathbf{I}_N = \left[\begin{array}{cccc|cccc} \sigma^2 & \sigma_u^2 & 0 & 0 & \sigma_u^2 & \sigma_u^2 & 0 & 0 \\ \sigma_u^2 & \sigma^2 & 0 & 0 & \sigma_u^2 & \sigma_u^2 & 0 & 0 \\ 0 & 0 & \sigma^2 & \sigma_u^2 & 0 & 0 & \sigma_u^2 & \sigma_u^2 \\ 0 & 0 & \sigma_u^2 & \sigma^2 & 0 & 0 & \sigma_u^2 & \sigma_u^2 \\ \hline \sigma_u^2 & \sigma_u^2 & 0 & 0 & \sigma^2 & \sigma_u^2 & 0 & 0 \\ \sigma_u^2 & \sigma_u^2 & 0 & 0 & \sigma_u^2 & \sigma^2 & 0 & 0 \\ 0 & 0 & \sigma_u^2 & \sigma_u^2 & 0 & 0 & \sigma^2 & \sigma_u^2 \\ 0 & 0 & \sigma_u^2 & \sigma_u^2 & 0 & 0 & \sigma_u^2 & \sigma^2 \end{array}\right]$$

where $\sigma^2 = \sigma_u^2 + \sigma_e^2$ and the population values of Y and $\mathbf{X}$ in each block satisfy

$$E_X(\mathbf{y}_q) = \mathbf{X}_q\boldsymbol{\beta} = \mathbf{f}_q \tag{3.2}$$

$$Var_X(\mathbf{y}_q) = \sigma_u^2\mathbf{Z}_q\mathbf{Z}_q^\top + \sigma_e^2\mathbf{I}_q \tag{3.3}$$

$$Cov_X(\mathbf{y}_q, \mathbf{y}_r) = \sigma_u^2\mathbf{Z}_q\mathbf{Z}_r^\top \tag{3.4}$$

where the subscript X denotes conditioning on the value $\mathbf{X_q}$, r is another block index, σ_u^2 is the between-group variance, and σ_e^2 is the within-group variance.

Note that apart from the regression parameter $\boldsymbol{\beta}$ in (3.2), which is the target of inference, variance components σ_u^2 and σ_e^2 in (3.3) are also included as unknown parameters which need to be estimated. Given the $\mathbf{y}_r$ and $\mathbf{X}_q$, the optimal estimator of $\boldsymbol{\beta}$ is then its Weighted Least Squares (WLS) estimator

$$\hat{\boldsymbol{\beta}} = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r \right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r \right) \tag{3.5}$$

where $\mathbf{W} = \mathbf{V}^{-1}$ and $\mathbf{W}_{qr}$ is the component of $\mathbf{W}$ corresponding to block $F = q$ and block $F = r$.

However, to be able to calculate (3.5), the linkage has to be perfect but this does not usually happen. Instead, what we can do is to substitute the linked data values $\mathbf{y}_r^*$ for $\mathbf{y}_r$, which leads to the naïve linked data WLS estimator

$$\hat{\boldsymbol{\beta}}^* = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r \right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right). \tag{3.6}$$

3.2 Bias-Corrected WLS Inference

Under the linkage error model (2.23), (3.6) can be rewritten in the form

$$\hat{\boldsymbol{\beta}}^* = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r \right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{A}_r \mathbf{y}_r \right).$$

Under non-informative linkage

$$E_X(\mathbf{A}_r \mathbf{y}_r) = E_X(\mathbf{A}_r) E_X(\mathbf{y}_r) = \mathbf{E}_r \mathbf{f}_r$$

thus

$$E_X(\hat{\boldsymbol{\beta}}^*) = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r\right) \boldsymbol{\beta} = \mathbf{D}\boldsymbol{\beta}. \tag{3.7}$$

Clearly, the naïve WLS estimator (3.6) based on the linked data set is biased. Given $\mathbf{E}_r$ and $\mathbf{W}_{qr}$ are known and the inverse of $\mathbf{D}$ in (3.7) exists, Chambers [7] suggests an unbiased estimator using a ratio-type correction for the bias in the naive estimator of $\boldsymbol{\beta}$,

$$\hat{\boldsymbol{\beta}}_R = \mathbf{D}^{-1}\hat{\boldsymbol{\beta}}^* = \left\{ \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r\right) \right\}^{-1} \hat{\boldsymbol{\beta}}^*.$$

Given that $\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r$ is of full rank, then

$$\hat{\boldsymbol{\beta}}_R = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^*\right). \tag{3.8}$$

By using the law of total variance, variance of $\mathbf{y}_q^*$ is given by

$$\begin{aligned}
Var_X(\mathbf{y}_q^*) &= E_X\left\{Var_{AX}(\mathbf{y}_q^*)\right\} + Var_X\left\{E_{AX}\left(\mathbf{y}_q^*\right)\right\} \\
&= E_X\left\{Var_{AX}(\mathbf{A}_q\mathbf{y}_q)\right\} + Var_X\left\{E_{AX}\left(\mathbf{A}_q\mathbf{y}_q\right)\right\} \\
&= E_X\left\{\mathbf{A}_q Var_{AX}(\mathbf{y}_q)\mathbf{A}_q^\top\right\} + Var_X\left\{\mathbf{A}_q E_{AX}\left(\mathbf{y}_q\right)\right\} \\
&= E_X\left\{\mathbf{A}_q(\sigma_u^2\mathbf{Z}_q\mathbf{Z}_q^\top + \sigma_e^2\mathbf{I}_q)\mathbf{A}_q^\top\right\} + Var_X\left(\mathbf{A}_q\mathbf{f}_q\right) \\
&= \sigma_u^2 E_X\left(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top\right) + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q \\
&= \sigma_u^2\ \mathbf{K}_q + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q \\
&= \boldsymbol{\Sigma}_q
\end{aligned} \tag{3.9}$$

where $\mathbf{V}_q$ was approximated by Chambers [7] as

$$\mathbf{V}_q \approx \text{diag}\left[(1-\lambda_q)\left\{\lambda_q(f_i - \bar{f}_q)^2 + \bar{f}_q^{\,(2)} - \bar{f}_q^2\right\}\right] \tag{3.10}$$

where $\mathbf{f}_q = (f_i)$ and $\bar{f}_q,\ \bar{f}_q{}^{(2)}$ denote the block q averages of the components of $\mathbf{f}_q$ and their squares respectively. A similar approximation of $\mathbf{K}_q$ is developed in Appendix A. Defining $\mathbf{K}_q = [k_{ij}]$, this approximation is given by

$$\mathbf{K}_q = \begin{cases} \lambda_q + \frac{(1-\lambda_q)}{M_q - 1}(G_q n_{qh} - 1), & \text{if } i = j \\ \left\{\lambda_q + (n_{qh} - 2)\frac{(1-\lambda_q)}{M_q-1}\right\}^2 & \\ +(n_{qh} - 1)\left\{\frac{(1-\lambda_q)}{M_q-1}\right\}^2 \{1 + (G_q - 1)n_{qh}\}, & \text{if } i \neq j; \\ & i, j \in g;\ g = h \\ (n_{qh} - 1)\frac{(1-\lambda_q)}{M_q-1}\left\{2\lambda_q + \frac{(1-\lambda_q)}{M_q-1}(G_q n_{qh} - 2)\right\}, & \text{if } i \neq j;\ i \in g \\ & j \in h;\ g \neq h. \end{cases} \tag{3.11}$$

where G_q is number of groups in block q and n_{qh} is number of population units in block q and group h. In terms of computational aspects of $\mathbf{K}_q$, it can be seen in Appendix D that an element k_{ij} is determined depending on whether i and j are in the same group. If they are, then we have to consider whether the element is on diagonal in that group.

The covariance between $\mathbf{y}_q^*$ and $\mathbf{y}_r^*$ is then

$$\begin{aligned} Cov_X(\mathbf{y}_q^*, \mathbf{y}_r^*) &= Cov_X(\mathbf{A}_q\mathbf{y}_q, \mathbf{A}_r\mathbf{y}_r) \\ &= E_X\left\{Cov_X(\mathbf{A}_q\mathbf{y}_q, \mathbf{A}_r\mathbf{y}_r \mid \mathbf{A}_q, \mathbf{A}_r)\right\} \\ &\quad + Cov_X\left\{E_X(\mathbf{A}_q\mathbf{y}_q \mid \mathbf{A}_q), E_X(\mathbf{A}_r\mathbf{y}_r \mid \mathbf{A}_r)\right\} \\ &= E_X\left\{\mathbf{A}_q Cov_X(\mathbf{y}_q, \mathbf{y}_r)\mathbf{A}_r^\top\right\} + Cov_X(\mathbf{A}_q\mathbf{f}_q, \mathbf{A}_r\mathbf{f}_r) \\ &= \sigma_u^2\left(\mathbf{E}_q\mathbf{Z}_q\mathbf{Z}_r^\top\mathbf{E}_r^\top\right) \end{aligned} \tag{3.12}$$

In order to obtain the variance of the estimator $\hat{\boldsymbol{\beta}}_R$, the *ultimate cluster variance estimation* will be employed since there is a correlation within a cluster and deriving

$Var_X(\hat{\boldsymbol{\beta}}_R)$ directly can be complicated. Thus, the variance of $\hat{\boldsymbol{\beta}}_R$ is given by

$$Var_X(\hat{\boldsymbol{\beta}}_R) = Var_X\left\{ \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r \right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right) \right\}$$

$$\begin{aligned} &= \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r \right)^{-1} Var_X \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right) \\ &\quad \left\{ \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{E}_r \mathbf{X}_r \right)^{-1} \right\}^\top. \end{aligned}$$

Note that

$$\begin{aligned} Var_X \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right) &= Var_X \left[\sum_g \left\{ \sum_q \sum_r (\mathbf{X}_q^g)^\top \mathbf{W}_{qr}^g \mathbf{y}_r^{g*} \right\} \right] \\ &= Var_X \left[\frac{1}{G} \sum_g \left\{ G \sum_q \sum_r (\mathbf{X}_q^g)^\top \mathbf{W}_{qr}^g \mathbf{y}_r^{g*} \right\} \right]. \end{aligned}$$

Since $G \sum_q \sum_r (\mathbf{X}_q^g)^\top \mathbf{W}_{qr}^g \mathbf{y}_r^{g*}$ are uncorrelated with one another, a variance estimator for $\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^*$ is of the form

$$\begin{aligned} \hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right) &= \frac{1}{G(G-1)} \sum_g \left\{ G \sum_q \sum_r (\mathbf{X}_q^g)^\top \mathbf{W}_{qr}^g \mathbf{y}_r^{g*} - \sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right\} \\ &\quad \left\{ G \sum_q \sum_r (\mathbf{X}_q^g)^\top \mathbf{W}_{qr}^g \mathbf{y}_r^{g*} - \sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r^* \right\}^\top. \end{aligned} \tag{3.13}$$

An estimator of $Var_X(\hat{\boldsymbol{\beta}}_R)$ is then

$$\begin{aligned} \hat{\mathbf{V}}_X(\hat{\boldsymbol{\beta}}_R) &= \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{E}_r \mathbf{X}_r \right)^{-1} \hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{y}_r^* \right) \\ &\quad \left\{ \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{E}_r \mathbf{X}_r \right)^{-1} \right\}^\top. \end{aligned} \tag{3.14}$$

3.3 Efficient Linear Estimation Using Linked Data

Apart from the approach in Section 3.2, we can fit a linear mixed model using the probability-linked data by directly modeling the regression relationship between the linked values $\mathbf{y}_q^*$ and the values in $\mathbf{X}_q$. Since $\mathbf{y}_q^* = \mathbf{A}_q\mathbf{y}_q$, and $\mathbf{A}_q$ and $\mathbf{y}_q$ are independently distributed given $\mathbf{X}_q$ it follows

$$E_X(\mathbf{y}_q^*) = E_X(\mathbf{A}_q)E_X(\mathbf{y}_q) = \mathbf{E}_q\mathbf{X}_q\boldsymbol{\beta} = \mathbf{H}_q\boldsymbol{\beta}. \tag{3.15}$$

We see that the $\mathbf{y}_q^*$ can also be modeled linearly, with regression coefficient $\boldsymbol{\beta}$ but with a modified set of explanatory variables $\mathbf{H}_q$ in block q. Following Lahiri and Larsen [38], an alternative estimator of $\boldsymbol{\beta}$ in this case is therefore

$$\begin{aligned}\hat{\boldsymbol{\beta}}_A &= \left(\sum_q\sum_r \mathbf{H}_q^\top\mathbf{W}_{qr}\mathbf{H}_r\right)^{-1}\left(\sum_q\sum_r \mathbf{H}_q^\top\mathbf{W}_{qr}\mathbf{y}_r^*\right)\\ &= \left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\mathbf{W}_{qr}\mathbf{E}_r\mathbf{X}_r\right)^{-1}\left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\mathbf{W}_{qr}\mathbf{y}_r^*\right).\end{aligned} \tag{3.16}$$

This estimator would be optimal if the regression errors under (3.15) were homoscedastic. Clearly, this generally does not hold, since (3.9) shows that the variances of the regression errors defined by the linked data vary between blocks. By obtaining the solution of the optimal estimating equation defined in Theorem (2.1), the best linear unbiased estimator (BLUE) for β given these data can be approximated by

$$\begin{aligned}\hat{\boldsymbol{\beta}}_C &= \left(\sum_q\sum_r \mathbf{H}_q^\top\boldsymbol{\Sigma}_{qr}\mathbf{H}_r\right)^{-1}\left(\sum_q\sum_r \mathbf{H}_q^\top\boldsymbol{\Sigma}_{qr}\mathbf{y}_r^*\right)\\ &= \left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_{qr}\mathbf{E}_r\mathbf{X}_r\right)^{-1}\left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_{qr}\mathbf{y}_r^*\right)\end{aligned} \tag{3.17}$$

where $\boldsymbol{\Sigma} = Var^{-1}(\mathbf{y}^*)$ and $\boldsymbol{\Sigma}_{qr}$ is the component of $\boldsymbol{\Sigma}$ corresponding to block $F = q$ and block $F = r$. To compute $\boldsymbol{\Sigma}$, it can be done by first combining variance - covari-

ance matrices of all blocks defined in (3.9) and (3.12) to construct $Var(\mathbf{y}^*)$. Then, one can simply use function `solve` in `R` to obtain $\mathbf{\Sigma} = Var^{-1}(\mathbf{y}^*)$. Nevertheless, in practice, a wide range of linked data sets are relatively large and obtaining $\mathbf{\Sigma}$ directly can be difficult. Hence, one can avoid this potential problem by employing the following lemma [55].

Lemma 3.1 *When* $\mathbf{A}$, $\mathbf{D}$ *are symmetric matrices such that the inverses which occur in the expressions exist,*

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{F}\mathbf{E}^{-1}\mathbf{F}^\top & -\mathbf{F}\mathbf{E}^{-1} \\ -\mathbf{E}^{-1}\mathbf{F}^\top & \mathbf{E}^{-1} \end{bmatrix}$$

where $\mathbf{E} = \mathbf{D} - \mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B},\ \mathbf{F} = \mathbf{A}^{-1}\mathbf{B}$.

The estimators of the variances of (3.16) and (3.17) can be obtained using the ultimate cluster variance estimation developed in the previous section. That is,

$$\begin{aligned} \hat{\mathbf{V}}_X(\hat{\boldsymbol{\beta}}_A) &= \left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{E}_r\mathbf{X}_r\right)^{-1} \hat{V}_X\left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{y}_r^*\right) \\ &\quad \left\{\left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{E}_r\mathbf{X}_r\right)^{-1}\right\}^\top \end{aligned} \tag{3.18}$$

where

$$\begin{aligned} \hat{V}_X\left(\sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{y}_r^*\right) &= \frac{1}{G(G-1)} \\ &\quad \sum_g\left\{G\sum_q\sum_r (\mathbf{X}_q^g)^\top(\mathbf{E}_q^g)^\top\hat{\mathbf{W}}_{qr}^g\mathbf{y}_r^{g*} - \sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{y}_r^*\right\} \\ &\quad \left\{G\sum_q\sum_r (\mathbf{X}_q^g)^\top(\mathbf{E}_q^g)^\top\hat{\mathbf{W}}_{qr}^g\mathbf{y}_r^{g*} - \sum_q\sum_r \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\mathbf{W}}_{qr}\mathbf{y}_r^*\right\}^\top. \end{aligned}$$

Similarly,

$$\hat{\mathbf{V}}_X(\hat{\boldsymbol{\beta}}_C) = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{E}_r \mathbf{X}_r\right)^{-1} \hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{y}_r^*\right) \left\{ \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{E}_r \mathbf{X}_r\right)^{-1} \right\}^\top \tag{3.19}$$

where

$$\hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{y}_r^*\right) = \frac{1}{G(G-1)} \sum_g \left\{ G \sum_q \sum_r (\mathbf{X}_q^g)^\top (\mathbf{E}_q^g)^\top \hat{\boldsymbol{\Sigma}}_{qr}^g \mathbf{y}_r^{g*} - \sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{y}_r^* \right\} \left\{ G \sum_q \sum_r (\mathbf{X}_q^g)^\top (\mathbf{E}_q^g)^\top \hat{\boldsymbol{\Sigma}}_{qr}^g \mathbf{y}_r^{g*} - \sum_q \sum_r \mathbf{X}_q^\top \mathbf{E}_q^\top \hat{\boldsymbol{\Sigma}}_{qr} \mathbf{y}_r^* \right\}^\top .$$

3.4 A Fixed Population Approach

In the case that linkage is perfect, the efficient estimator is then the WLS estimator

$$B = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r\right). \tag{3.20}$$

In this situation, we look for a predictor $\hat{B}$ of B that satisfies

$$E_{YX}(\hat{B}) = B. \tag{3.21}$$

Note that we need to estimate (3.20) since $\mathbf{y}_r$ is unknown in practice.

In order to derive a predictor that satisfies (3.21), Chambers [7] has suggested the class of linear predictors of B that can be written in the form

$$\hat{B} = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{L}_r \mathbf{y}_r^*\right).$$

Following Kovacevic's suggestion (personal communication with Chambers, 2008) $\mathbf{L}_r = (\mathbf{E}_r^\top \mathbf{E}_r)^{-1}\mathbf{E}_r^\top$ leads to the predictor

$$\hat{\boldsymbol{\beta}}_B = \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{X}_r\right)^{-1} \left\{\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} (\mathbf{E}_r^\top \mathbf{E}_r)^{-1} \mathbf{E}_r^\top \mathbf{y}_r^*\right\} \tag{3.22}$$

which is unbiased since

$$E_{YX}(\hat{\boldsymbol{\beta}}_B) = \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{X}_r\right)^{-1} \left(\sum_q \sum_r \mathbf{X}_q^\top \mathbf{W}_{qr} \mathbf{y}_r\right).$$

That is, (3.21) holds.

By using the ultimate cluster variance estimation, the variance estimator of (3.22) is

$$\begin{aligned} \hat{\mathbf{V}}_X(\hat{\boldsymbol{\beta}}_B) \;=\; & \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{X}_r\right)^{-1} \hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} (\mathbf{E}_r^\top \mathbf{E}_r)^{-1} \mathbf{E}_r^\top \mathbf{y}_r^*\right) \\ & \left\{\left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} \mathbf{X}_r\right)^{-1}\right\}^\top . \end{aligned} \tag{3.23}$$

where

$$\begin{aligned} \hat{V}_X \left(\sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} (\mathbf{E}_r^\top \mathbf{E}_r)^{-1} \mathbf{E}_r^\top \mathbf{y}_r^*\right) \;=\; & \frac{1}{G(G-1)} \\ & \sum_g \left[G \sum_q \sum_r (\mathbf{X}_q^g)^\top \hat{\mathbf{W}}_{qr}^g \{(\mathbf{E}_r^g)^\top \mathbf{E}_r^g\}^{-1} (\mathbf{E}_r^g)^\top \mathbf{y}_r^{g*} \right. \\ & \left. - \sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} (\mathbf{E}_r^\top \mathbf{E}_r)^{-1} \mathbf{E}_r^\top \mathbf{y}_r^*\right] \\ & \left[G \sum_q \sum_r (\mathbf{X}_q^g)^\top \hat{\mathbf{W}}_{qr}^g \{(\mathbf{E}_r^g)^\top \mathbf{E}_r^g\}^{-1} (\mathbf{E}_r^g)^\top \mathbf{y}_r^{g*} \right. \\ & \left. - \sum_q \sum_r \mathbf{X}_q^\top \hat{\mathbf{W}}_{qr} (\mathbf{E}_r^\top \mathbf{E}_r)^{-1} \mathbf{E}_r^\top \mathbf{y}_r^*\right]^\top . \end{aligned}$$

3.5 Variance Estimation When Linkage Probabilities Are Estimated

The work so far is based on the assumption that the matrix of expected values $\mathbf{E}_q$ for the stochastic linkage matrix $\mathbf{A}_q$ is known. That is, we assume that the probabilities of correct linkage λ_q are known under the the exchangeable errors model (2.27). In real life, however, this is hardly possible, and these probabilities will have to be estimated in some way. Clearly, this estimation can cause extra uncertainty which needs to be accounted for when the variance estimation for the estimators is carried out.

As in Chambers [7], the regression model can be fitted via the solution of an estimating equation. That is, it can be estimated by solving

$$\mathbf{H}(\theta) = \mathbf{0}$$

where $\mathbf{H}(\theta)$ is a $p-$dimensional unbiased estimating function for the estimators θ. Let λ denote the vector defined by the block-specific values of λ_r, the estimating function by its bias-corrected version is then given by

$$\mathbf{H}^*(\theta,\lambda) = \sum_q \sum_r \mathbf{D}_{qr}(\theta)\{\mathbf{y}_r^* - \mathbf{E}_r(\lambda)\mathbf{f}_r(\theta)\} = \sum_q \sum_r \mathbf{U}_{qr}(\theta,\lambda_r)$$

which is a function of both θ and λ. By developing a first order Taylor series approximation, we obtain

$$\begin{aligned}
\mathbf{0} = \mathbf{H}^*(\hat{\theta},\hat{\lambda}) &\approx \mathbf{H}^*(\theta_0,\lambda_0) + \partial_\theta \mathbf{H}^*(\theta_0,\lambda_0)(\hat{\theta}-\theta_0) + \partial_\lambda \mathbf{H}^*(\theta_0,\lambda_0)(\hat{\lambda}-\lambda_0) \\
\hat{\theta}-\theta_0 &= -(\partial_\theta \mathbf{H}_0^*)^{-1}\left\{\mathbf{H}_0^* + \partial_\lambda \mathbf{H}_0^*(\hat{\lambda}-\lambda_0)\right\} \\
\hat{\theta} &= \theta_0 - (\partial_\theta \mathbf{H}_0^*)^{-1}\left\{\mathbf{H}_0^* + \partial_\lambda \mathbf{H}_0^*(\hat{\lambda}-\lambda_0)\right\}
\end{aligned}$$

where $\mathbf{H}_0^* = \mathbf{H}^*(\theta_0,\lambda_0)$. Note that θ_0 and λ_0 denote the true values of these parameters with $\hat{\theta}$ and $\hat{\lambda}$ denote their corresponding estimators.

Then, we can approximate the variance of $\hat{\theta}$ by

$$\begin{aligned} Var_X(\hat{\theta}) &\approx (\partial_\theta \mathbf{H}_0^*)^{-1} Var_X \left\{ \mathbf{H}_0^* + \partial_\lambda \mathbf{H}_0^* (\hat{\lambda} - \lambda_0) \right\} \left\{ (\partial_\theta \mathbf{H}_0^*)^{-1} \right\}^\top \\ &= (\partial_\theta \mathbf{H}_0^*)^{-1} \left\{ Var_X \left(\mathbf{H}_0^* \right) + \left(\partial_\lambda \mathbf{H}_0^* \right) Var_X(\hat{\lambda}) \left(\partial_\lambda \mathbf{H}_0^* \right)^\top \right\} \left\{ (\partial_\theta \mathbf{H}_0^*)^{-1} \right\}^\top . \end{aligned}$$

Note that $\left(\partial_\lambda \mathbf{H}_0^*\right) Var_X(\hat{\lambda}) \left(\partial_\lambda \mathbf{H}_0^*\right)^\top$ is an extra uncertainty arisen from estimating λ. If data are perfectly linked, this term will become zero.

Once again, there is a correlation within a cluster and deriving $Var_X\left(\mathbf{H}_0^*\right)$ directly can be complicated, therefore $Var_X\left(\mathbf{H}_0^*\right)$ can be obtained via the ultimate cluster variance estimator which does not require an estimate of the intracluster correlation i.e.

$$\begin{aligned} Var_X\left(\mathbf{H}_0^*\right) &= Var_X \left[\sum_q \sum_r \mathbf{D}_{qr}(\theta_0) \{ \mathbf{y}_r^* - \mathbf{E}_r(\lambda_0) \mathbf{f}_r(\theta_0) \} \right] \\ &= Var_X \left[\sum_g \left\{ \sum_q \sum_r \mathbf{D}_{qr}^g(\theta_0) \mathbf{y}_r^{g*} \right\} \right] \\ &= Var_X(\mathbf{H}_{0C}^*) \end{aligned}$$

where

$$\mathbf{H}_{0C}^* = \sum_g \left\{ \sum_q \sum_r \mathbf{D}_{qr}^g(\theta_0) \mathbf{y}_r^{g*} \right\} = \frac{1}{G} \sum_g \mathbf{H}_{0Cg}^*$$

and

$$\mathbf{H}_{0Cg}^* = G \sum_q \sum_r \mathbf{D}_{qr}^g(\theta_0) \mathbf{y}_r^{g*}.$$

Note that the $\mathbf{H}_{0Cg}^*$ are uncorrelated with one another. Therefore, a variance esti-

mator for $\mathbf{H}^*_{0C}$ can be written in the form

$$\begin{aligned}\hat{V}_X\left(\mathbf{H}^*_{0C}\right) &= \frac{1}{G^2}\hat{V}_X\left(\sum_g \mathbf{H}^*_{0Cg}\right)\\ &= \frac{1}{G^2}\sum_g \hat{V}_X\left(\mathbf{H}^*_{0Cg}\right)\\ &= \frac{1}{G(G-1)}\sum_g (\mathbf{H}^*_{0Cg}-\mathbf{H}^*_{0C})(\mathbf{H}^*_{0Cg}-\mathbf{H}^*_{0C})^\top.\end{aligned}$$

Thus, we have

$$\begin{aligned}Var_X(\hat{\theta}) &= \left(\sum_q\sum_r \partial_\theta \mathbf{U}_{0qr}\right)^{-1}\left\{\hat{V}_X\left(\mathbf{H}^*_{0C}\right)+\sum_q\sum_r\left(\partial_{\lambda_r}\mathbf{U}_{0qr}\right)Var_X(\hat{\lambda}_r)\left(\partial_{\lambda_r}\mathbf{U}_{0qr}\right)^\top\right\}\\ &\quad\left\{\left(\sum_q\sum_r \partial_\theta \mathbf{U}_{0qr}\right)^{-1}\right\}^\top\end{aligned}$$

where $\mathbf{U}_{0qr} = \mathbf{U}_{qr}(\theta_0, \lambda_{0r})$. Also, we assume that the distribution of $\hat{\lambda}$ is approximately independent of the distribution of $\mathbf{H}^*_0$. Furthermore, we need an expression for

$$\partial_{\lambda_r}\mathbf{U}_{0qr} = \partial_{\lambda_r}\mathbf{D}_{0qr}\{\mathbf{y}^*_r - \mathbf{E}_r(\lambda_r)\mathbf{f}_{0r}\} = -\mathbf{D}_{0qr}\partial_{\lambda_r}\{\mathbf{E}_r(\lambda_r)\}\mathbf{f}_{0r}$$

where $\mathbf{f}_{0r} = \mathbf{f}_r(\theta_0)$.

Under the exchangeable linkage errors model (2.27)

$$\begin{aligned}\mathbf{E}_r(\lambda_r) &= \left\{\lambda_r - \frac{(1-\lambda_r)}{M_r-1}\right\}\mathbf{I}_r + \frac{(1-\lambda_r)}{M_r-1}\mathbf{1}_r\mathbf{1}_r^\top\\ &= (M_r-1)^{-1}\{(\lambda_r M_r - 1)\mathbf{I}_r + (1-\lambda_r)\mathbf{1}_r\mathbf{1}_r^\top\}\end{aligned}$$

so

$$\partial_{\lambda_r}\{\mathbf{E}_r(\lambda_r)\} = (M_r-1)^{-1}(M_r\mathbf{I}_r - \mathbf{1}_r\mathbf{1}_r^\top)$$

and hence

$$\partial_{\lambda_r}\mathbf{U}_{0qr} = -(M_r-1)^{-1}\mathbf{D}_{0qr}(M_r\mathbf{I}_r - \mathbf{1}_r\mathbf{1}_r^\top)\mathbf{f}_{0r}.$$

That is, we have

$$Var_X(\hat{\theta}) \approx \left(\sum_q \sum_r \partial_\theta \mathbf{U}_{0qr}\right)^{-1} \left\{\hat{V}_X\left(\mathbf{H}^*_{0C}\right) + \sum_q \sum_r \mathbf{D}_{0qr}\Delta_{0r}\mathbf{D}^\top_{0qr}\right\} \left\{\left(\sum_q \sum_r \partial_\theta \mathbf{U}_{0qr}\right)^{-1}\right\}^\top \tag{3.24}$$

where

$$\begin{aligned}\Delta_{0r} &= (M_r - 1)^{-2} Var_X(\hat{\lambda}_r)(M_r\mathbf{I}_r - \mathbf{1}_r\mathbf{1}_r^\top)\mathbf{f}_{0r}\mathbf{f}_{0r}^\top(M_r\mathbf{I}_r - \mathbf{1}_r\mathbf{1}_r^\top) \\ &= M_r^2(M_r - 1)^{-2} Var_X(\hat{\lambda}_r)(\mathbf{f}_{0r} - \mathbf{1}_r\bar{f}_{0r})(\mathbf{f}_{0r} - \mathbf{1}_r\bar{f}_{0r})^\top.\end{aligned}$$

Note that, in order to calculate (3.24) we need to obtain $Var_X(\hat{\lambda}_r)$. If the estimates of the probabilities of correct linkage λ_r are obtained by checking a random audit sample of linked records in each block i.e. the number of correct linkages follows the binomial distribution, then $Var_X(\hat{\lambda}_r) = m_r^{-1}\lambda_{0r}(1 - \lambda_{0r})$. Thus, the estimator of (3.24) can be obtained by plugging in estimates for unknown quantities. That is, the estimator of $Var_X(\hat{\theta})$ is

$$\hat{V}_X(\hat{\theta}) = \left(\sum_q \sum_r \partial_\theta \hat{\mathbf{U}}_{qr}\right)^{-1} \left\{\hat{V}_X\left(\hat{\mathbf{H}}^*_C\right) + \sum_q \sum_r \hat{\mathbf{D}}_{qr}\hat{\Delta}_r\hat{\mathbf{D}}^\top_{qr}\right\} \left\{\left(\sum_q \sum_r \partial_\theta \hat{\mathbf{U}}_{qr}\right)^{-1}\right\}^\top \tag{3.25}$$

The specification of the components of (3.25) for the special case of linear regression and linkage bias corrected estimators described so far is shown in Table 3.1.

Table 3.1: Specification of $\hat{\mathbf{D}}_{qr}$ and $\partial_\theta \hat{\mathbf{U}}_{qr}$ in (3.25) for the linear case.

Estimator	$\hat{\mathbf{D}}_{qr}$	$\partial_\theta \hat{\mathbf{U}}_{qr}$
(3.8)	$(\mathbf{X}_q^g)^\top \hat{\mathbf{W}}_{qr}^g$	$-\hat{\mathbf{D}}_{qr}\hat{\mathbf{E}}_r\mathbf{X}_r$
(3.16)	$(\mathbf{X}_q^g)^\top (\hat{\mathbf{E}}_q^g)^\top \hat{\mathbf{W}}_{qr}^g$	$-\hat{\mathbf{D}}_{qr}\hat{\mathbf{E}}_r\mathbf{X}_r$
(3.17)	$(\mathbf{X}_q^g)^\top (\hat{\mathbf{E}}_q^g)^\top \hat{\Sigma}_{qr}^g$	$-\hat{\mathbf{D}}_{qr}\hat{\mathbf{E}}_r\mathbf{X}_r$
(3.22)	$(\mathbf{X}_q^g)^\top \hat{\mathbf{W}}_{qr}^g \{(\hat{\mathbf{E}}_r^g)^\top \hat{\mathbf{E}}_r^g\}^{-1} (\hat{\mathbf{E}}_r^g)^\top$	$-\hat{\mathbf{D}}_{qr}\hat{\mathbf{E}}_r\mathbf{X}_r$

Chapter 4

Estimation of Variance Components

When fitting a mixed model, estimation of variance components is also an important objective. In this chapter we develop appropriate modifications to three standard methods of variance components estimation in order to account for linkage error. These are the method of moments, typically referred to as the analysis of variance (ANOVA) method, the maximum likelihood (ML) method, and the restricted maximum likelihood (REML) method. The details of each method are described in the following sections.

4.1 Analysis of Variance (ANOVA)

Historically, ANOVA is the starting point for estimation of variance components [60]. The method is based on deriving the expected values of the between groups sum of squares (SSA) and the within groups sum of squares (SSE) under the mixed model of interest. One then equates observed and expected values and solves for the estimated values of the variance components. If it was perfect linked data, one would have

$$\hat{\sigma}_e^2 = \frac{\text{SSE}}{N-G} = \text{MSE} \quad \text{and} \quad \hat{\sigma}_u^2 = \frac{\text{SSA} - \hat{\sigma}_e^2(G-1)}{\left(N - \frac{\sum_g N_g^2}{N}\right)}.$$

However, given the linked data, the two sums of squares that are the basis of ANOVA for the linked data are

$$\text{SSA} = \mathbf{y}^{*\top}\mathbf{B}\,\mathbf{y}^*, \qquad ;\mathbf{B} = \{b_{ij}\} = \begin{cases} \frac{1}{N_g} - \frac{1}{N}, & \text{if } i,j \in g \\ -\frac{1}{N}, & \text{if } i \in g,\ j \in h,\ g \neq h \end{cases}$$

$$\text{SSE} = \mathbf{y}^{*\top}\mathbf{C}\,\mathbf{y}^*, \qquad ;\mathbf{C} = \{c_{ij}\} = \begin{cases} 1 - \frac{1}{N_g}, & \text{if } i,j \in g,\ i = j \\ -\frac{1}{N_g}, & \text{if } i,j \in g,\ i \neq j \\ 0, & \text{otherwise.} \end{cases}$$

We then need to obtain $\text{E}_X(\text{SSA})$ and $\text{E}_X(\text{SSE})$ and equate them to the observed values mentioned above. That is, we have

$$\begin{aligned} E_X(\text{SSA}) &= E_X\left(\sum_q \sum_r \mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^*\right) \\ &= \sum_q E_X\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qq}\mathbf{y}_q^*\right) + \sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^*\right) \\ &= \sigma_u^2\left\{\sum_q \text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r\neq q} \text{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\right)\right\} + \sigma_e^2 \sum_q \text{tr}\left(\mathbf{B}_{qq}\right) \\ &\quad + \sum_q \left\{\text{tr}\left(\mathbf{B}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qq}\mathbf{E}_q\mathbf{f}_q\right\} + \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r \\ &= \sigma_u^2 a + \sigma_e^2 b + m_0 \end{aligned}$$

and

$$\begin{aligned} E_X(\text{SSE}) &= \sigma_u^2 \left\{ \sum_q \text{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r \neq q} \text{tr}\left(\mathbf{E}_r \mathbf{Z}_r \mathbf{Z}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qr}\right) \right\} + \sigma_e^2 \sum_q \text{tr}\left(\mathbf{C}_{qq}\right) \\ &\quad + \sum_q \left\{ \text{tr}\left(\mathbf{C}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qq} \mathbf{E}_q \mathbf{f}_q \right\} + \sum_q \sum_{r \neq q} \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qr} \mathbf{E}_r \mathbf{f}_r \\ &= \sigma_u^2 c + \sigma_e^2 d + n_0. \end{aligned}$$

The details of method of deriving the expected values of the two sums of squares are illustrated in Appendix B which yields the variance components estimators

$$\hat{\sigma}_e^2 = \frac{mc - na}{bc - da} \tag{4.1}$$

and

$$\hat{\sigma}_u^2 = \frac{m - \hat{\sigma}_e^2 b}{a} \tag{4.2}$$

where

$$\begin{aligned} a &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r \neq q} \text{tr}\left(\mathbf{E}_r \mathbf{Z}_r \mathbf{Z}_q^\top \mathbf{E}_q^\top \mathbf{B}_{qr}\right) \\ b &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\right) \\ c &= \sum_q \text{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r \neq q} \text{tr}\left(\mathbf{E}_r \mathbf{Z}_r \mathbf{Z}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qr}\right) \\ d &= \sum_q \text{tr}\left(\mathbf{C}_{qq}\right) \\ m &= \text{SSA} - \sum_q \left\{ \text{tr}\left(\mathbf{B}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{B}_{qq} \mathbf{E}_q \mathbf{f}_q \right\} - \sum_q \sum_{r \neq q} \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{B}_{qr} \mathbf{E}_r \mathbf{f}_r \\ n &= \text{SSE} - \sum_q \left\{ \text{tr}\left(\mathbf{C}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qq} \mathbf{E}_q \mathbf{f}_q \right\} - \sum_q \sum_{r \neq q} \mathbf{f}_q^\top \mathbf{E}_q^\top \mathbf{C}_{qr} \mathbf{E}_r \mathbf{f}_r. \end{aligned}$$

Note that $c = 0$ if linkage is perfect, i.e. $\hat{\sigma}_e^2 = \dfrac{n}{d}$ and $\hat{\sigma}_u^2 = \dfrac{m - \hat{\sigma}_e^2 b}{a}$

where

$$\begin{aligned} a &= N - \frac{\sum_g N_g^2}{N} \\ b &= G - 1 \text{ is degrees of freedom of SSA} \\ d &= N - G \text{ is degrees of freedom of SSE} \\ m &= \text{SSA} - \sum_q \sum_r \mathbf{f}_q^\top \mathbf{B}_{qr} \mathbf{f}_r \\ n &= \text{SSE} - \sum_q \sum_r \mathbf{f}_q^\top \mathbf{C}_{qr} \mathbf{f}_r. \end{aligned}$$

To obtain the variance of $\hat{\sigma}_e^2$ and $\hat{\sigma}_u^2$, following (4.1) and Appendix B

$$\begin{aligned} \hat{\sigma}_e^2 &= \frac{mc - na}{bc - da} \\ &= \frac{(\text{SSA} - m_0)c - (\text{SSE} - n_0)a}{bc - da}. \end{aligned}$$

Thus,

$$\begin{aligned} Var(\hat{\sigma}_e^2) &= Var\left[\frac{(\text{SSA})c - (\text{SSE})a}{bc - da}\right] \\ &= \frac{1}{(bc - da)^2} Var[(\text{SSA})c - (\text{SSE})a] \\ &= \frac{1}{(bc - da)^2} Var(\mathbf{y^*}^\top \mathbf{L}\, \mathbf{y^*}) \quad \text{where } \mathbf{L} = \mathbf{B}c - \mathbf{C}a. \end{aligned}$$

Since $\mathbf{y}^* \sim N(\mathbf{Ef}, \mathbf{\Sigma})$, by using Theorem S4 of Searle *et al.* [60] we have

$$Var(\hat{\sigma}_e^2) = \frac{1}{(bc - da)^2} \left[2\text{tr}(\mathbf{L\Sigma L\Sigma}) + 4\mathbf{f}^\top \mathbf{E}^\top \mathbf{L\Sigma LEf}\right]. \tag{4.3}$$

An estimator of (4.3) is then

$$\hat{V}(\hat{\sigma}_e^2) = \frac{1}{(bc-da)^2}\left[2\text{tr}(\mathbf{L}\hat{\mathbf{\Sigma}}\mathbf{L}\hat{\mathbf{\Sigma}}) + 4\mathbf{f}^\top\mathbf{E}^\top\mathbf{L}\hat{\mathbf{\Sigma}}\mathbf{L}\mathbf{E}\mathbf{f}\right]. \tag{4.4}$$

Again, following (4.2) and Appendix B

$$\begin{aligned}\hat{\sigma}_u^2 &= \frac{m-\hat{\sigma}_e^2 b}{a} \\ &= \frac{\text{SSA}-m_0-\hat{\sigma}_e^2 b}{a}.\end{aligned}$$

Thus,

$$\begin{aligned}Var(\hat{\sigma}_u^2) &= Var\left(\frac{\text{SSA}-\hat{\sigma}_e^2 b}{a}\right) \\ &= \frac{1}{a^2}Var(\text{SSA}-\hat{\sigma}_e^2 b) \\ &= \frac{1}{a^2}\left[Var(\text{SSA}) - b^2 Var(\hat{\sigma}_e^2)\right] \\ &= \frac{1}{a^2}\left[2\text{tr}(\mathbf{B}\mathbf{\Sigma}\mathbf{B}\mathbf{\Sigma}) + 4\mathbf{f}^\top\mathbf{E}^\top\mathbf{B}\mathbf{\Sigma}\mathbf{B}\mathbf{E}\mathbf{f} - b^2 Var(\hat{\sigma}_e^2)\right]. \qquad (4.5)\end{aligned}$$

An estimator of (4.5) is then given by

$$\hat{V}(\hat{\sigma}_u^2) = \frac{1}{a^2}\left[2\text{tr}(\mathbf{B}\hat{\mathbf{\Sigma}}\mathbf{B}\hat{\mathbf{\Sigma}}) + 4\mathbf{f}^\top\mathbf{E}^\top\mathbf{B}\hat{\mathbf{\Sigma}}\mathbf{B}\mathbf{E}\mathbf{f} - b^2\hat{V}(\hat{\sigma}_e^2)\right]. \tag{4.6}$$

Note that although ANOVA estimators have merit in the unbiasedness property, the ANOVA estimates can be negative. Consequently, it is usually better to use a method of estimation that explicitly excludes the possibility of negative estimates. Such methods are maximum likelihood (ML) and restricted maximum likelihood (REML).

4.2 Pseudo-Maximum Likelihood (Pseudo-ML)

One of the most used and well-respected statistical methods used for fitting a statistical model to data, and providing estimates for the model's parameters is *maximum likelihood estimation*. It has a variety of optimality properties as described in many mathematical statistics books (see for example, [6, 31]). Here we use this method as an alternative approach to constructing efficient estimators of $\boldsymbol{\beta}, \sigma_u^2$ and σ_e^2 given the linked data.

Unlike the ANOVA method of estimation, a basic requirement of ML estimation is that the probability distribution of the data is known. We follow the usual convention of assuming multivariate normality. However, in general, there are no analytical expressions for the variance component estimators obtained by using ML. This has to be done numerically. In this section, we will use the method of scoring as an algorithm to obtain the estimators.

4.2.1 Point Estimation

To find the maximum likelihood estimate of $\boldsymbol{\beta}$ we first assume that σ_u^2 and σ_e^2 are known. Later, once the maximum likelihood estimate of $\boldsymbol{\beta}$ is obtained, we will use this estimate to estimate σ_u^2 and σ_e^2. Given that $\mathbf{y}^* \sim N(\mathbf{Ef}, \boldsymbol{\Sigma})$, y_i^* are not independent, the likelihood function is given by

$$L = (2\pi)^{-N/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{y}^* - \mathbf{Ef})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef})\right\} \tag{4.7}$$

and the log-likelihood function is denoted by l:

$$l = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{y}^* - \mathbf{Ef})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}). \tag{4.8}$$

In what follows, we assume that $\boldsymbol{\Sigma}$ is fixed. Differentiating (4.8) with respect to $\boldsymbol{\beta}$ then yields

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = \boldsymbol{l}_{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{E}^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}). \tag{4.9}$$

Similarly, differentiating (4.8) with respect to σ_u^2 leads to

$$\frac{\partial l}{\partial \sigma_u^2} = \boldsymbol{l}_{\sigma_u^2} = -\frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) + \frac{1}{2}(\mathbf{y}^* - \mathbf{Ef})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}) \tag{4.10}$$

where $\boldsymbol{\Sigma}_u = \partial\boldsymbol{\Sigma}/\partial\sigma_u^2$. Finally, differentiating (4.8) with respect to σ_e^2 gives

$$\frac{\partial l}{\partial \sigma_e^2} = \boldsymbol{l}_{\sigma_e^2} = -\frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}^{-1}\right) + \frac{1}{2}(\mathbf{y}^* - \mathbf{Ef})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}). \tag{4.11}$$

The pseudo-ML estimators for $\boldsymbol{\beta}, \sigma_u^2$ and σ_e^2 are defined by setting the derivatives (4.9), (4.10) and (4.11) to zero and solving for these parameters. Note that we refer to the resulting estimators as pseudo-ML because their estimating functions, which are defined by these derivatives, are based on the assumption that $\boldsymbol{\Sigma}$ is a known matrix. However, in reality this matrix is a function of $\boldsymbol{\beta}, \sigma_u^2$ and σ_e^2, and so analytic solutions to these estimating equations do not exist. We therefore now describe how the method of scoring [60] can be used to solve them.

Let $\boldsymbol{\theta}$ denote the vector of parameters to be estimated, i.e., $\boldsymbol{\theta}^\top = \begin{bmatrix}\boldsymbol{\beta}^\top & \sigma_u^2 & \sigma_e^2\end{bmatrix}$. The method of scoring thus uses an iteration scheme defined by

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + \{\bar{\mathbf{I}}(\boldsymbol{\theta}^{(m)})\}^{-1}\left.\frac{\partial l}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{(m)}},$$

where $\bar{\mathbf{I}}(\boldsymbol{\theta}^{(m)})$ is the expected information matrix calculated at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(m)}$.
We now develop the expression for $\bar{\mathbf{I}}(\boldsymbol{\theta})$. From (4.9),

$$\begin{aligned}
\frac{\partial^2 l}{\partial \boldsymbol{\beta}\, \partial \boldsymbol{\beta}^\top} = \boldsymbol{l}_{\boldsymbol{\beta}\boldsymbol{\beta}} &= -\mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\mathbf{EX} \\
\frac{\partial^2 l}{\partial \sigma_u^2 \partial \boldsymbol{\beta}^\top} = \boldsymbol{l}_{\sigma_u^2\boldsymbol{\beta}} &= -\mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}) \\
\frac{\partial^2 l}{\partial \sigma_e^2 \partial \boldsymbol{\beta}^\top} = \boldsymbol{l}_{\sigma_e^2\boldsymbol{\beta}} &= -\mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}).
\end{aligned}$$

Note that from (3.9) and (3.12), $Var_X(\mathbf{y}_q^*) = \sigma_u^2\mathbf{K}_q + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q$ and $Cov_X(\mathbf{y}_q^*, \mathbf{y}_r^*) =$

$\sigma_u^2(\mathbf{E}_q\mathbf{Z}_q\mathbf{Z}_r^\top\mathbf{E}_r^\top)$. Suppose there were only two blocks, $\boldsymbol{\Sigma}_u$ would be

$$\boldsymbol{\Sigma}_u = \begin{bmatrix} \mathbf{K}_q & \mathbf{E}_q\mathbf{Z}_q\mathbf{Z}_r^\top\mathbf{E}_r^\top \\ \mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top & \mathbf{K}_r \end{bmatrix}.$$

Furthermore, from (4.10) and (4.11) we have

$$\frac{\partial^2 l}{\partial\sigma_u^2\,\partial\sigma_u^2} = \boldsymbol{l}_{\sigma_u^2\sigma_u^2} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) - (\mathbf{y}^* - \mathbf{Ef})^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef})$$

$$\frac{\partial^2 l}{\partial\sigma_e^2\,\partial\sigma_u^2} = \boldsymbol{l}_{\sigma_e^2\sigma_u^2} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) - (\mathbf{y}^* - \mathbf{Ef})^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef})$$

and

$$\frac{\partial^2 l}{\partial\sigma_e^2\,\partial\sigma_e^2} = \boldsymbol{l}_{\sigma_e^2\sigma_e^2} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right) - (\mathbf{y}^* - \mathbf{Ef})^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{Ef}).$$

The expected information matrix is obtained by taking the expected values of the derivatives obtained above, noting that $E(\mathbf{y}^*) = \mathbf{Ef}$ and hence $E(\mathbf{y}^* - \mathbf{Ef}) = \mathbf{0}$. Also, $E(\mathbf{y}^* - \mathbf{Ef})^\top\mathbf{C}(\mathbf{y}^* - \mathbf{Ef}) = \mathrm{tr}(\mathbf{C}\boldsymbol{\Sigma})$ for non-stochastic $\mathbf{C}$. It follows that

$$\begin{aligned}
-E(\boldsymbol{l}_{\boldsymbol{\beta}\boldsymbol{\beta}}) &= \mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\mathbf{EX} \\
-E(\boldsymbol{l}_{\sigma_u^2\boldsymbol{\beta}}) &= \mathbf{0} \\
-E(\boldsymbol{l}_{\sigma_e^2\boldsymbol{\beta}}) &= \mathbf{0} \\
-E(\boldsymbol{l}_{\sigma_u^2\sigma_u^2}) &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) + \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) \\
-E(\boldsymbol{l}_{\sigma_e^2\sigma_u^2}) &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) + \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right)
\end{aligned}$$

and

$$\begin{aligned} -E(\boldsymbol{l}_{\sigma_e^2\sigma_e^2}) &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right) + \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right). \end{aligned}$$

Therefore, the expected information matrix is then given by

$$\begin{aligned} \bar{\mathbf{I}}\begin{bmatrix}\boldsymbol{\beta}\\ \sigma_u^2\\ \sigma_e^2\end{bmatrix} &= -E\begin{bmatrix}\boldsymbol{l}_{\boldsymbol{\beta}\boldsymbol{\beta}} & \boldsymbol{l}_{\sigma_u^2\boldsymbol{\beta}} & \boldsymbol{l}_{\sigma_e^2\boldsymbol{\beta}}\\ \boldsymbol{l}_{\boldsymbol{\beta}\sigma_u^2} & \boldsymbol{l}_{\sigma_u^2\sigma_u^2} & \boldsymbol{l}_{\sigma_e^2\sigma_u^2}\\ \boldsymbol{l}_{\boldsymbol{\beta}\sigma_e^2} & \boldsymbol{l}_{\sigma_u^2\sigma_e^2} & \boldsymbol{l}_{\sigma_e^2\sigma_e^2}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\mathbf{E}\mathbf{X} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) & \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right)\\ \mathbf{0} & \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) & \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right)\end{bmatrix}. \end{aligned} \tag{4.12}$$

Asymptotically,

$$\begin{aligned} Var\begin{bmatrix}\hat{\boldsymbol{\beta}}\\ \hat{\sigma}_u^2\\ \hat{\sigma}_e^2\end{bmatrix} &\simeq \left(\bar{\mathbf{I}}\begin{bmatrix}\boldsymbol{\beta}\\ \sigma_u^2\\ \sigma_e^2\end{bmatrix}\right)^{-1} \\ &= \begin{bmatrix}Var(\hat{\boldsymbol{\beta}}) & \mathbf{0}\\ \mathbf{0} & Var(\hat{\boldsymbol{\sigma}}^2)\end{bmatrix} \end{aligned} \tag{4.13}$$

where

$$Var(\hat{\boldsymbol{\beta}}) \simeq (\mathbf{X}^\top\mathbf{E}^\top\boldsymbol{\Sigma}^{-1}\mathbf{E}\mathbf{X})^{-1} \tag{4.14}$$

and

$$Var(\hat{\boldsymbol{\sigma}}^2) \simeq 2\begin{bmatrix}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) & \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right)\\ \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\right) & \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\right)\end{bmatrix}^{-1}. \tag{4.15}$$

Estimators of (4.14) and (4.15) are then

$$\hat{V}(\hat{\boldsymbol{\beta}}) \simeq (\mathbf{X}^\top \mathbf{E}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{E}\mathbf{X})^{-1} \tag{4.16}$$

and

$$\hat{V}(\hat{\boldsymbol{\sigma}}^2) \simeq 2 \begin{bmatrix} \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}_u\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}_u\right) & \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}_u\right) \\ \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}_u\right) & \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\Sigma}}^{-1}\right) \end{bmatrix}^{-1}. \tag{4.17}$$

Even if we ignore the fact that $\boldsymbol{\Sigma}$ is a function of $\boldsymbol{\beta}$, the simulation results, although not reported in Chapter 6, showed that there is not much difference between pseudo-ML estimates and ML estimates using `optim` function in `R`. Moreover, by using pseudo-ML method we can obtain the standard errors of the estimators and therefore the confidence intervals of the estimators. On the other hand, `optim` function cannot provide the standard errors of the estimators and therefore the confidence intervals of the estimators. It is also time-consuming compared to the pseudo-ML method due to numerically intensive computation.

4.2.2 Confidence Intervals

Following the previous subsection, we see that the approximate variance-covariance matrix for the ML estimates is given by the inverse of the information matrix corresponding to the log-likelihood function $l = l(\boldsymbol{\beta}, \sigma_u^2, \sigma_e^2)$. Since

$$E(\boldsymbol{l}_{\sigma_u^2\boldsymbol{\beta}}) = 0 \quad \text{and} \quad E(\boldsymbol{l}_{\sigma_e^2\boldsymbol{\beta}}) = 0,$$

the information matrix corresponding to a linear mixed model is block diagonal and, therefore, the ML estimates of the fixed effects $\boldsymbol{\beta}$ are asymptotically uncorrelated with the ML estimates of σ_u^2 and σ_e^2.

Following Pinheiro and Bates [54], we assume that variance components follow log-normal distribution. The approximate distributions of the ML estimates in a

linear mixed model in this situation are

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &\sim N\left\{\boldsymbol{\beta}, (\mathbf{X}^\top \mathbf{E}^\top \boldsymbol{\Sigma}^{-1} \mathbf{E}\mathbf{X})^{-1}\right\}, \\
\begin{bmatrix} \log \hat{\sigma}_u \\ \log \hat{\sigma}_e \end{bmatrix} &\sim N\left(\begin{bmatrix} \log \sigma_u \\ \log \sigma_e \end{bmatrix}, \mathbf{I}^{-1}(\sigma_u, \sigma_e) \right), \qquad (4.18) \\
\mathbf{I}(\sigma_u, \sigma_e) &= -\begin{bmatrix} \dfrac{\partial^2 l}{\partial^2 \log \sigma_u} & \dfrac{\partial^2 l}{\partial \log \sigma_e \, \partial \log \sigma_u} \\ \dfrac{\partial^2 l}{\partial \log \sigma_u \, \partial \log \sigma_e} & \dfrac{\partial^2 l}{\partial^2 \log \sigma_e} \end{bmatrix}
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial^2 l}{\partial^2 \log \sigma_u} &= -2\sigma_u^2 \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u) + 2(\sigma_u^2)^2 \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u) \\
&\quad + 2\sigma_u^2 (\mathbf{y}^* - \mathbf{E}\mathbf{f})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{E}\mathbf{f}) \\
&\quad - 4(\sigma_u^2)^2 (\mathbf{y}^* - \mathbf{E}\mathbf{f})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{E}\mathbf{f}) \\
\frac{\partial^2 l}{\partial \log \sigma_e \, \partial \log \sigma_u} &= 2\sigma_u^2\sigma_e^2 \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u) - 4\sigma_u^2\sigma_e^2 (\mathbf{y}^* - \mathbf{E}\mathbf{f})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_u\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{E}\mathbf{f}) \\
\frac{\partial^2 l}{\partial \log \sigma_e \, \partial \log \sigma_e} &= -2\sigma_e^2 \mathrm{tr}(\boldsymbol{\Sigma}^{-1}) + 2(\sigma_e^2)^2 \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}) \\
&\quad + 2\sigma_e^2 (\mathbf{y}^* - \mathbf{E}\mathbf{f})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{E}\mathbf{f}) \\
&\quad - 4(\sigma_e^2)^2 (\mathbf{y}^* - \mathbf{E}\mathbf{f})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^* - \mathbf{E}\mathbf{f})
\end{aligned}$$

Note that we use $\log \sigma_u$ and $\log \sigma_e$ in place of σ_u^2 and σ_e^2 to give unrestricted parameterizations for which the normal approximation tends to be more accurate [54].

Approximate confidence intervals on the fixed effects and the variance-covariance components are obtained using the approximation distributions described above. An

approximate confidence interval of level $1-\alpha$ for $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} \pm z_{1-\frac{\alpha}{2}} se(\hat{\boldsymbol{\beta}})$$

where $z_{1-\frac{\alpha}{2}}$ denotes the $1-\alpha/2$ quantile of the standard normal distribution and $se(\hat{\boldsymbol{\beta}})$ is evaluated at the estimated value of σ_u^2 and σ_e^2.

Confidence intervals on the between-group and within-group standard deviations σ_u and σ_e are obtained from the approximation distribution in (4.18). Letting $[\mathbf{I}^{-1}]_{\sigma_{uu}}$ denote the first diagonal element of the inverse empirical information matrix defined in (4.18), an approximate confidence interval of level $1-\alpha$ for σ_u is

$$\left[\hat{\sigma}_u \exp\left(-z_{1-\frac{\alpha}{2}}\sqrt{[\mathbf{I}^{-1}]_{\sigma_{uu}}}\right), \hat{\sigma}_u \exp\left(z_{1-\frac{\alpha}{2}}\sqrt{[\mathbf{I}^{-1}]_{\sigma_{uu}}}\right)\right].$$

Similarly, letting $[\mathbf{I}^{-1}]_{\sigma_{ee}}$ denote the last diagonal element of the inverse empirical information matrix defined in (4.18), an approximate confidence interval of level $1-\alpha$ for σ_e is

$$\left[\hat{\sigma}_e \exp\left(-z_{1-\frac{\alpha}{2}}\sqrt{[\mathbf{I}^{-1}]_{\sigma_{ee}}}\right), \hat{\sigma}_e \exp\left(z_{1-\frac{\alpha}{2}}\sqrt{[\mathbf{I}^{-1}]_{\sigma_{ee}}}\right)\right].$$

4.3 Pseudo-Restricted Maximum Likelihood (Pseudo-REML)

One criticism of the ML method is that in estimating variance components it takes no account of the degrees of freedom that are involved in estimating fixed effects [60]. Also, the variance component estimators obtained by solving the likelihood equations are generally biased, unlike the ANOVA estimators [27, 60].

The first criticism above is overcome by using restricted maximum likelihood (REML) [60]. The main idea behind REML estimation is to separate that part of the data used for estimation of random effects from that used for estimation of fixed effects. Rather than using $\mathbf{y}^*$ directly, REML uses ML estimating equations based

on a modified response variable defined by a linear combination $\mathbf{s}^\top \mathbf{y}^*$ of elements of $\mathbf{y}^*$, chosen in such a way that the distribution of this combination does not depend on the fixed effects in the model. That is, the vectors $\mathbf{s}^\top$ are chosen so that $E(\mathbf{s}^\top \mathbf{y}^*) = \mathbf{s}^\top \mathbf{E}\mathbf{X}\boldsymbol{\beta} = \mathbf{0}$, i.e.

$$\mathbf{s}^\top \mathbf{E}\mathbf{X} = \mathbf{0}. \tag{4.19}$$

Note that strict application of the REML approach requires that the distribution of $\mathbf{s}^\top \mathbf{y}^*$ is independent of $\boldsymbol{\beta}$. However, when we use linked data, the variance of $\mathbf{y}^*$ is implicitly a function of this parameter. Consequently, we refer to this method as "pseudo-REML" since it is based on application of standard REML arguments, ignoring the fact that the variance still depends on the fixed effects in the model.

When $\mathbf{E}\mathbf{X}$ of order $N \times p$ has rank r, there are only $N-r$ linearly independent vectors $\mathbf{s}^\top$ satisfying (4.19) [60]. Using a set of such $N-r$ linearly independent vectors $\mathbf{s}^\top$ as rows of $\mathbf{S}^\top$, we then have $\mathbf{S}^\top \mathbf{y}^*$ where $\mathbf{S}^\top$ is a $(N-r) \times N$ matrix whose rows are any $N-r$ linearly independent rows of the matrix $\mathbf{I} - \mathbf{E}\mathbf{X}\{(\mathbf{E}\mathbf{X})^\top (\mathbf{E}\mathbf{X})\}^{-1} (\mathbf{E}\mathbf{X})^\top$.

With $\mathbf{y}^* \sim N(\mathbf{E}\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$ we have, for $\mathbf{S}^\top \mathbf{E}\mathbf{X} = \mathbf{0}$

$$\mathbf{S}^\top \mathbf{y}^* \sim N(\mathbf{0}, \mathbf{S}^\top \boldsymbol{\Sigma} \mathbf{S}).$$

Let l_R be the log likelihood for the variance effects defined by $\mathbf{S}^\top \mathbf{y}^*$. That is

$$\begin{aligned} l_R &= -\frac{1}{2}(N-r)\log(2\pi) - \frac{1}{2}\log|\mathbf{S}^\top \boldsymbol{\Sigma} \mathbf{S}| - \frac{1}{2}\mathbf{y}^{*\top} \mathbf{S}(\mathbf{S}^\top \boldsymbol{\Sigma} \mathbf{S})^{-1} \mathbf{S}^\top \mathbf{y}^* \\ &= -\frac{1}{2}(N-r)\log(2\pi) - \frac{1}{2}\log|\mathbf{S}^\top \boldsymbol{\Sigma} \mathbf{S}| - \frac{1}{2}\mathbf{y}^{*\top} \mathbf{M} \mathbf{y}^* \end{aligned}$$

where $\mathbf{M} = \mathbf{S}(\mathbf{S}^\top \boldsymbol{\Sigma} \mathbf{S})^{-1} \mathbf{S}^\top = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} \mathbf{E}\mathbf{X} \left\{ (\mathbf{E}\mathbf{X})^\top \boldsymbol{\Sigma}^{-1} \mathbf{E}\mathbf{X} \right\}^{-1} (\mathbf{E}\mathbf{X})^\top \boldsymbol{\Sigma}^{-1}$.

The REML estimating function for $\boldsymbol{\beta}$ is unchanged from the corresponding ML estimating function (4.9). However the ML estimating functions (4.10) and (4.11) for the variance components σ_u^2 and σ_e^2 are now replaced by alternative REML

estimating functions obtained by differentiating l_R. In this context, we note that

$$\begin{aligned}
\frac{\partial \mathbf{M}}{\partial \sigma_u^2} &= \frac{\partial}{\partial \sigma_u^2}\mathbf{S}(\mathbf{S}^\top\boldsymbol{\Sigma}\mathbf{S})^{-1}\mathbf{S}^\top \\
&= -\mathbf{S}(\mathbf{S}^\top\boldsymbol{\Sigma}\mathbf{S})^{-1}\mathbf{S}^\top\boldsymbol{\Sigma}_u\mathbf{S}(\mathbf{S}^\top\boldsymbol{\Sigma}\mathbf{S})^{-1}\mathbf{S}^\top = -\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M} \\
\frac{\partial \mathbf{M}}{\partial \sigma_e^2} &= -\mathbf{M}\mathbf{M}.
\end{aligned}$$

The REML estimating equations are defined by setting (4.9) and the REML estimating functions for σ_u^2 and σ_e^2 to zero. As before, we use the method of scoring to solve these equations. In order to define the expected information matrix in this case, we need the first and second derivatives of l_R :

$$\begin{aligned}
\frac{\partial l_R}{\partial \sigma_u^2} = \boldsymbol{l}_{\sigma_u^2} &= -\frac{1}{2}\mathrm{tr}\left\{(\mathbf{S}^\top\boldsymbol{\Sigma}\mathbf{S})^{-1}\mathbf{S}^\top\boldsymbol{\Sigma}_u\mathbf{S}\right\} + \frac{1}{2}\mathbf{y}^{*\top}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{y}^* \\
&= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\right) + \frac{1}{2}\mathbf{y}^{*\top}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{y}^* \\
\frac{\partial^2 l_R}{\partial \sigma_u^2\,\partial \sigma_u^2} = \boldsymbol{l}_{\sigma_u^2\sigma_u^2} &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\right) - \mathbf{y}^{*\top}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{y}^* \\
\frac{\partial^2 l_R}{\partial \sigma_e^2\,\partial \sigma_u^2} = \boldsymbol{l}_{\sigma_e^2\sigma_u^2} &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\right) - \mathbf{y}^{*\top}\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{y}^* \\
\frac{\partial l_R}{\partial \sigma_e^2} = \boldsymbol{l}_{\sigma_e^2} &= -\frac{1}{2}\mathrm{tr}\left\{(\mathbf{S}^\top\boldsymbol{\Sigma}\mathbf{S})^{-1}\mathbf{S}^\top\mathbf{S}\right\} + \frac{1}{2}\mathbf{y}^{*\top}\mathbf{M}\mathbf{M}\mathbf{y}^* \\
&= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\right) + \frac{1}{2}\mathbf{y}^{*\top}\mathbf{M}\mathbf{M}\mathbf{y}^* \\
\frac{\partial^2 l_R}{\partial \sigma_e^2\,\partial \sigma_e^2} = \boldsymbol{l}_{\sigma_e^2\sigma_e^2} &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\right) - \mathbf{y}^{*\top}\mathbf{M}\mathbf{M}\mathbf{M}\mathbf{y}^*.
\end{aligned}$$

Then, we take expected values of second derivatives of l_R.

$$\begin{aligned} -E(\boldsymbol{l}_{\sigma_u^2\sigma_u^2}) &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\right) + \mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}\right) \\ &\quad + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{E}^\top\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{E}\mathbf{X}\boldsymbol{\beta} \\ &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\right) + \mathrm{tr}\left(\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}\mathbf{M}\right) + \mathbf{0} \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\right) \end{aligned}$$

where the second equality follows because $\mathbf{MEX} = \mathbf{0}$ and the third equality follows because $\mathbf{M}\boldsymbol{\Sigma}\mathbf{M} = \mathbf{M}$.

$$\begin{aligned} -E(\boldsymbol{l}_{\sigma_e^2\sigma_u^2}) &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\right) + \mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}\right) + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{E}^\top\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\mathbf{E}\mathbf{X}\boldsymbol{\beta} \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\right) \end{aligned}$$

and

$$\begin{aligned} -E(\boldsymbol{l}_{\sigma_e^2\sigma_e^2}) &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\right) + \mathrm{tr}\left(\mathbf{M}\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}\right) + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{E}^\top\mathbf{M}\mathbf{M}\mathbf{M}\mathbf{E}\mathbf{X}\boldsymbol{\beta} \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}\right). \end{aligned}$$

It follows that the component $\bar{\mathbf{I}}(\sigma_u^2, \sigma_e^2)$ of the observed information matrix is then given by

$$\therefore \quad \mathbf{I}\begin{bmatrix}\sigma_u^2 \\ \sigma_e^2\end{bmatrix} = -E\begin{bmatrix}\boldsymbol{l}_{\sigma_u^2\sigma_u^2} & \boldsymbol{l}_{\sigma_u^2\sigma_e^2} \\ \boldsymbol{l}_{\sigma_e^2\sigma_u^2} & \boldsymbol{l}_{\sigma_e^2\sigma_e^2}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\mathrm{tr}\left(\mathbf{M}\boldsymbol{\Sigma}_u\mathbf{M}\boldsymbol{\Sigma}_u\right) & \mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\right) \\ \mathrm{tr}\left(\mathbf{M}\mathbf{M}\boldsymbol{\Sigma}_u\right) & \mathrm{tr}\left(\mathbf{M}\mathbf{M}\right)\end{bmatrix}. \quad (4.20)$$

Similar to the ML approach, we have in the limit

$$Var\begin{bmatrix}\sigma_u^2\\ \sigma_e^2\end{bmatrix} \simeq 2\begin{bmatrix}\mathrm{tr}\,(\mathbf{M\Sigma}_u\mathbf{M\Sigma}_u) & \mathrm{tr}\,(\mathbf{MM\Sigma}_u)\\ \mathrm{tr}\,(\mathbf{MM\Sigma}_u) & \mathrm{tr}\,(\mathbf{MM})\end{bmatrix}^{-1} \tag{4.21}$$

exactly the same as (4.15) for ML, except for ML there is $\mathbf{\Sigma}^{-1}$ where here we have $\mathbf{M}$.

Approximate confidence intervals for REML estimates can be obtained using the formulations developed in Section 4.2, with the obvious modifications.

4.4 Comparison Between ANOVA, ML and REML Estimation

As it is known that ML and REML are based on the likelihood principle which leads to useful properties such as consistency, asymptotic normality, and efficiency [60, 66]. Therefore, it would be reasonable if each of ML and REML were to be preferred over any ANOVA method. In contrast, the only merit property of ANOVA estimators is unbiasedness.

To compare between ML and REML, it is hard to decide which method is the better one. Both methods have the same merits of being based on the maximum likelihood principle, and they have the same demerit of computational requirements. ML provides estimators of fixed effects, whereas REML does not. But ML estimators are generally biased, as we will see in Chapter 6, whereas REML estimators are not. Also, with balanced data REML solutions are identical to ANOVA estimators which have optimal minimum variance properties. As a result, REML would be more preferable compared to ML.

Chapter 5

Application to Longitudinal Modeling

In the previous chapters, the development of methods has been focusing on linked cross-sectional registers. In particular, when there are only two registers and measurements are clustered into groups such as people in a family, patients in a hospital or students in a school. However, many administrative data sources are longitudinal registers i.e. a single measurement is collected repeatedly over time on each subject in the study. Figure 5.1 is one example, which displays a longitudinal data set on bone mineral acquisition in young females [3, 68]. The data consist of spinal bone mineral density (SBMD) measurements on each of 230 female subjects aged between 8 and 27. Each subject is measured between one and four times. Clearly, measurements on the same subject are correlated with one another.

This chapter focuses on fitting linear mixed models to linked longitudinal registers. In particular, we initially use three registers for the regression analysis as it is the simplest case of this type of data. However, in the next section we will see clearly that we can easily deal with any number of data sets.

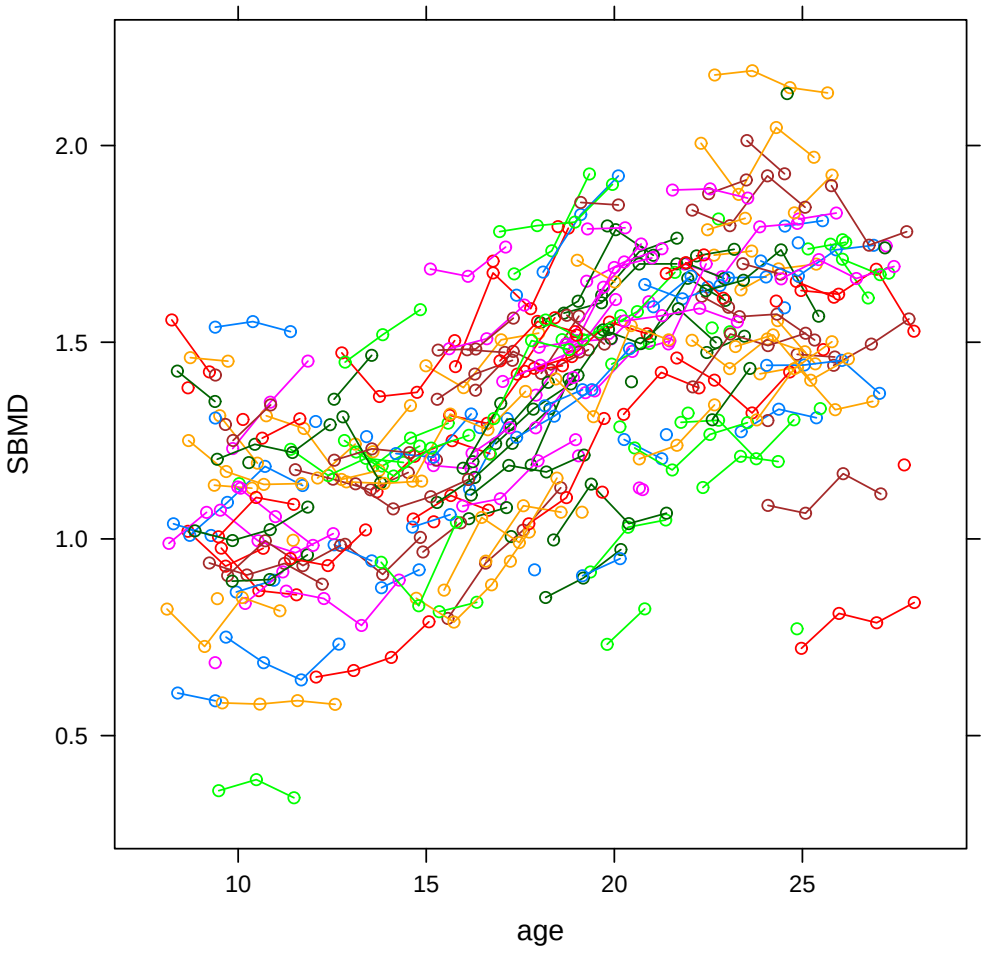

Figure 5.1: Spinal bone mineral data versus age for female subjects.

5.1 Background and Assumptions

Suppose that we have three registers created separately at different times. That is, we have X_1, Y_1 collected at time 1, X_2, Y_2 collected at time 2, and X_3, Y_3 collected at time 3. When these three data sets are linked with no unique identifier, there could be some linkage errors occurred both on X and Y at time 2 and time 3. To illustrate, suppose that in one block each register is made up of the three records, as in Table 5.1

Table 5.1: Illustration of the true linked longitudinal data.

Subject	Register 1	Register 2	Register 3
1	x_{11} y_{11}	x_{12} y_{12}	x_{13} y_{13}
2	x_{21} y_{21}	x_{22} y_{22}	x_{23} y_{23}
3	x_{31} y_{31}	x_{32} y_{32}	x_{33} y_{33}

where x_{ij} and y_{ij} denote values of X and Y of subject or individual i at time j in block q. Let T_i denote the within-subject sample size: it is the number of observations on subject i. We have a sample of M_q subjects in each block with i indexing subjects and running from 1 up to M_q.

The total number of observations is

$$N = \sum_{q=1}^{Q} \sum_{i=1}^{M_q} T_i.$$

We have two sample sizes associated with this type of data. The total number of observations or measurements is N and the total number of subjects n where $n = \sum_q M_q$. Dropping block notation for the time being, the entire set of observations in Table 5.1 is defined as

$$\mathbf{Y} = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ \hline y_{12} \\ y_{22} \\ y_{32} \\ \hline y_{13} \\ y_{23} \\ y_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} \quad \text{and} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{21} \\ 1 & x_{31} \\ \hline 1 & x_{12} \\ 1 & x_{22} \\ 1 & x_{32} \\ \hline 1 & x_{13} \\ 1 & x_{23} \\ 1 & x_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{x}_1 \\ \mathbf{1} & \mathbf{x}_2 \\ \mathbf{1} & \mathbf{x}_3 \end{bmatrix}.$$

The model equation (3.1) for the observations in Table 5.1 are

$$\begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ \hline y_{12} \\ y_{22} \\ y_{32} \\ \hline y_{13} \\ y_{23} \\ y_{33} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{21} \\ 1 & x_{31} \\ \hline 1 & x_{12} \\ 1 & x_{22} \\ 1 & x_{32} \\ \hline 1 & x_{13} \\ 1 & x_{23} \\ 1 & x_{33} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \\ \hline e_{12} \\ e_{22} \\ e_{32} \\ \hline e_{13} \\ e_{23} \\ e_{33} \end{bmatrix}.$$

Here register 1 is regarded as a bench mark data set and register 2 and 3 are linked with register 1 i.e. there could be some mislinks between register 1 and 2 or register 1 and 3. That is, we could have

$$\mathbf{Y}^* = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ \hline y_{22} \\ y_{12} \\ y_{32} \\ \hline y_{13} \\ y_{33} \\ y_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2^* \\ \mathbf{y}_3^* \end{bmatrix}.$$

The relationship between the linked values in $\mathbf{Y}^*$ and the true values in $\mathbf{Y}$ in this example is then characterized by

$$\mathbf{Y}^* = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2^* \\ \mathbf{y}_3^* \end{bmatrix} = \mathbf{AY} = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{A}_2\mathbf{y}_2 \\ \mathbf{A}_3\mathbf{y}_3 \end{bmatrix}.$$

At this stage we assume that the three subjects are in block q and $\mathbf{A}_{2q}$ and $\mathbf{A}_{3q}$ are independently distributed where $\mathbf{A}_{tq}$ are permutation matrices of order M_{tq}. By applying the non-informative linkage assumption as in (2.24) and (2.27), let

$$E(\mathbf{A}_{tq} \mid \mathbf{X}_{tq}) = \mathbf{E}_{tq} \tag{5.1}$$

where

$$\mathbf{E}_{tq} = (\lambda_{tq} - \gamma_{tq})\mathbf{I}_{tq} + \gamma_{tq}\mathbf{1}_{tq}\mathbf{1}_{tq}^{\top} \tag{5.2}$$

and

$$\begin{aligned}\lambda_{tq} &= \Pr(\text{correct linkage between } \mathbf{X}_q \text{ and } \mathbf{Y}_q \text{ at time 1 and } \mathbf{X}_q \text{ and } \mathbf{Y}_q \text{ at time } t)\\ \gamma_{tq} &= \Pr(\text{incorrect linkage between } \mathbf{X}_q \text{ and } \mathbf{Y}_q \text{ at time 1 and } \mathbf{X}_q \text{ and } \mathbf{Y}_q \text{ at time } t).\end{aligned}$$

We see that in each block we need to specify λ_{tq} in order to completely specify the first order properties of the linkage mechanism under the model (5.2). Obviously, number of λ_{tq} in each block is number of data sets minus one.

5.2 Weighted Least Squares Inference

In this section we focus on fitting linear mixed models to the linked longitudinal data. Let, $\mathbf{Y}^*$, the $N \times 1$ vector of observations, be taken to have model equation

$$\mathbf{Y}^* = \mathbf{X}^*\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{5.3}$$

where $\mathbf{Y}^* = \mathbf{A}\mathbf{Y}$ and $\mathbf{X}^* = (\mathbf{1}, \mathbf{A}\mathbf{X})$. Suppose that the population values of $\mathbf{Y}$ and $\mathbf{X}$ in each block satisfy equation (3.2) and (3.3). Then, the WLS estimator in this situation is

$$\hat{\boldsymbol{\beta}}^* = \left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{y}_q^*\right) \tag{5.4}$$

where $\boldsymbol{\Sigma}_q$ is defined similarly to (3.9) i.e.

$$\begin{aligned}\boldsymbol{\Sigma}_q = Var_X(\mathbf{y}_q^*) &= E_X\left\{Var_{AX}(\mathbf{y}_q^*)\right\} + Var_X\left\{E_{AX}\left(\mathbf{y}_q^*\right)\right\}\\ &= E_X\left\{\mathbf{A}_q Var_{AX}(\mathbf{y}_q)\mathbf{A}_q^\top\right\} + Var_X(\mathbf{A}_q\mathbf{f}_q)\\ &= E_X\left\{\mathbf{A}_q(\sigma_u^2\mathbf{Z}_q\mathbf{Z}_q^\top + \sigma_e^2\mathbf{I}_q)\mathbf{A}_q^\top\right\} + \mathbf{V}_q\\ &= \sigma_u^2 E_X(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top) + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q\\ &= \sigma_u^2\mathbf{K}_q^L + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q.\end{aligned} \tag{5.5}$$

where $\mathbf{K}_q^L$ denotes $E_X(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top)$ for this situation. Note that for balanced data, from the previous example,

$$\mathbf{Z}_q = \begin{bmatrix} \mathbf{Z}_{1q} \\ \mathbf{Z}_{2q} \\ \mathbf{Z}_{3q} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{1q} \\ \mathbf{I}_{2q} \\ \mathbf{I}_{3q} \end{bmatrix}$$

so

$$\mathbf{A}_q\mathbf{Z}_q = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{1q} \\ \mathbf{I}_{2q} \\ \mathbf{I}_{3q} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{1q} \\ \mathbf{A}_{2q} \\ \mathbf{A}_{3q} \end{bmatrix}.$$

Thus,

$$\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top = \begin{bmatrix} \mathbf{I}_{1q} & \mathbf{A}_{2q}^\top & \mathbf{A}_{3q}^\top \\ \mathbf{A}_{2q} & \mathbf{A}_{2q}\mathbf{A}_{2q}^\top & \mathbf{A}_{2q}\mathbf{A}_{3q}^\top \\ \mathbf{A}_{3q} & \mathbf{A}_{3q}\mathbf{A}_{2q}^\top & \mathbf{A}_{3q}\mathbf{A}_{3q}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{1q} & \mathbf{A}_{2q}^\top & \mathbf{A}_{3q}^\top \\ \mathbf{A}_{2q} & \mathbf{I}_{2q} & \mathbf{A}_{2q}\mathbf{A}_{3q}^\top \\ \mathbf{A}_{3q} & \mathbf{A}_{3q}\mathbf{A}_{2q}^\top & \mathbf{I}_{3q} \end{bmatrix}$$

since $\mathbf{A}_{tq}\mathbf{A}_{tq}^\top = \mathbf{I}_{tq}$. By assuming independence between $\mathbf{A}_{2q}$ and $\mathbf{A}_{3q}$, it follows that

$$\mathbf{K}_q^L = E_X(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top) = \begin{bmatrix} \mathbf{I}_{1q} & \mathbf{E}_{2q} & \mathbf{E}_{3q} \\ \mathbf{E}_{2q} & \mathbf{I}_{2q} & \mathbf{E}_{2q}\mathbf{E}_{3q} \\ \mathbf{E}_{3q} & \mathbf{E}_{3q}\mathbf{E}_{2q} & \mathbf{I}_{3q} \end{bmatrix}. \tag{5.6}$$

Then, under non-informative linkage

$$E_X(\mathbf{y}_q^*) = E_X(\mathbf{A}_q\mathbf{y}_q) = \mathbf{E}_q\mathbf{X}_q\boldsymbol{\beta}.$$

Thus, as in (3.7)

$$\begin{aligned} E_X(\hat{\boldsymbol{\beta}}^*) &= E_X\left[E_{AX}\left\{\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{y}_q^*\right)\right\}\right] \\ &= E_X\left\{\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{X}_q\right)\right\}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}. \end{aligned} \tag{5.7}$$

Clearly, the WLS estimator (5.4) based on the linked data set is unbiased. Therefore, there is no need to correct bias for $\boldsymbol{\beta}$.

The variance of (5.4) is obtained by using an iterated expectation argument as follows.

$$Var_X(\hat{\boldsymbol{\beta}}^*) = E_X\left\{Var_{AX}(\hat{\boldsymbol{\beta}}^*)\right\} + Var_X\left\{E_{AX}(\hat{\boldsymbol{\beta}}^*)\right\}.$$

Note that $E_{AX}(\hat{\boldsymbol{\beta}}^*) = \boldsymbol{\beta}$, therefore $Var_X\left\{E_{AX}(\hat{\boldsymbol{\beta}}^*)\right\} = 0$. That is,

$$Var_X(\hat{\boldsymbol{\beta}}^*) = E_X\left\{Var_{AX}(\hat{\boldsymbol{\beta}}^*)\right\}$$

where

$$\begin{aligned} Var_{AX}(\hat{\boldsymbol{\beta}}^*) &= \left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1} Var_{AX}\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{y}_q^*\right)\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1} \\ &= \left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\left\{\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q Var_{AX}(\mathbf{y}_q)\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right\} \\ &\quad \left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1} \end{aligned}$$

$$
\begin{aligned}
&= \left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\left\{\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q(\sigma_u^2\mathbf{Z}_q\mathbf{Z}_q^\top+\sigma_e^2\mathbf{I}_q)\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right\}\\
&\quad\left(\sum_q \mathbf{X}_q^{*\top}\boldsymbol{\Sigma}_q^{-1}\mathbf{X}_q^*\right)^{-1}\\
&= \left(\sum_q \mathbf{X}_q^\top\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{X}_q\right)^{-1}\left\{\sigma_u^2\sum_q \mathbf{X}_q^\top\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{X}_q\right.\\
&\quad\left.+\sigma_e^2\sum_q \mathbf{X}_q^\top\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{X}_q\right\}\left(\sum_q \mathbf{X}_q^\top\mathbf{A}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{A}_q\mathbf{X}_q\right)^{-1}.
\end{aligned}
$$

We see that taking expected values of the expression above is not tractable. Thus, obtaining a first order approximation gives

$$
\begin{aligned}
Var_X(\hat{\boldsymbol{\beta}}^*) &= E_X\left\{Var_{AX}(\hat{\boldsymbol{\beta}}^*)\right\}\\
&\approx \left(\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right)^{-1}\left\{\sigma_u^2\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right.\\
&\quad\left.+\sigma_e^2\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right\}\left(\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right)^{-1}.
\end{aligned}
$$

An estimator of $Var_X(\hat{\boldsymbol{\beta}}^*)$ is then

$$
\begin{aligned}
\hat{V}_X(\hat{\boldsymbol{\beta}}^*) &= \left(\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\boldsymbol{\Sigma}}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right)^{-1}\left\{\hat{\sigma}_u^2\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\boldsymbol{\Sigma}}_q^{-1}\mathbf{K}_q^L\hat{\boldsymbol{\Sigma}}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right.\\
&\quad\left.+\hat{\sigma}_e^2\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\boldsymbol{\Sigma}}_q^{-1}\hat{\boldsymbol{\Sigma}}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right\}\left(\sum_q \mathbf{X}_q^\top\mathbf{E}_q^\top\hat{\boldsymbol{\Sigma}}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right)^{-1}. \quad (5.8)
\end{aligned}
$$

5.3 Estimation of Variance Components

As discussed earlier, variance components estimation is also an important objective when fitting linear mixed models. The three methods of estimation are still described here. Most of the algebra are nearly the same to the previous chapter except the definition of $\mathbf{y}_q^*$ and therefore $\boldsymbol{\Sigma}_q^{-1}$.

5.3.1 Analysis of Variance (ANOVA)

Following Section 4.1 we have

$$\begin{aligned}
\text{SSA} &= \mathbf{y}^{*\top}\mathbf{B}\mathbf{y}^{*} \\
&= \sum_q \sum_r \mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^{*}
\end{aligned}$$

$$\begin{aligned}
E_X(\text{SSA}) &= \sum_q E_X\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qq}\mathbf{y}_q^{*}\right) + \sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^{*}\right) \\
&= \sum_q \text{tr}\left\{\mathbf{B}_{qq}Var_X(\mathbf{y}_q^{*})\right\} + \sum_q \mathbf{f}_q^{\top}\mathbf{E}_q\mathbf{B}_{qq}\mathbf{E}_q\mathbf{f}_q + \sum_q \sum_{r\neq q} \mathbf{f}_q^{\top}\mathbf{E}_q\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r \\
&= \sigma_u^2 \sum_q \text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q^{L}\right) + \sigma_e^2 \sum_q \text{tr}\left(\mathbf{B}_{qq}\right) + \sum_q \text{tr}(\mathbf{B}_{qq}\mathbf{V}_q) + \sum_q \sum_r \mathbf{f}_q^{\top}\mathbf{E}_q\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r \\
&= \sigma_u^2 a + \sigma_e^2 b + m_0
\end{aligned}$$

where

$$\begin{aligned}
a &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q^{L}\right) \\
b &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\right) \\
m_0 &= \sum_q \text{tr}(\mathbf{B}_{qq}\mathbf{V}_q) + \sum_q \sum_r \mathbf{f}_q^{\top}\mathbf{E}_q\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r.
\end{aligned}$$

Note that

$$\begin{aligned}
\text{SSE} &= \mathbf{y}^{*\top}\mathbf{C}\mathbf{y}^{*} \\
&= \sum_q \mathbf{y}_q^{*\top}\mathbf{C}_{qq}\mathbf{y}_q^{*} \quad \text{since } \mathbf{C}_{qr} = \mathbf{0}
\end{aligned}$$

$$\begin{aligned}\therefore\ E_X(\text{SSE}) &= \sigma_u^2 \sum_q \text{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q^L\right) + \sigma_e^2 \sum_q \text{tr}\left(\mathbf{C}_{qq}\right) + \sum_q \text{tr}(\mathbf{C}_{qq}\mathbf{V}_q) + \sum_q \mathbf{f}_q^\top \mathbf{E}_q \mathbf{C}_{qq} \mathbf{E}_q \mathbf{f}_q \\ &= \sigma_u^2 c + \sigma_e^2 d + n_0\end{aligned}$$

where

$$\begin{aligned}c &= \sum_q \text{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q^L\right) \\ d &= \sum_q \text{tr}\left(\mathbf{C}_{qq}\right) \\ n_0 &= \sum_q \left\{\text{tr}(\mathbf{C}_{qq}\mathbf{V}_q) + \mathbf{f}_q^\top \mathbf{E}_q \mathbf{C}_{qq} \mathbf{E}_q \mathbf{f}_q\right\}.\end{aligned}$$

Once again, by solving two linear equations,

$$\hat{\sigma}_u^2 a + \hat{\sigma}_e^2 b = m \quad ; m = \text{SSA} - m_0$$

and

$$\hat{\sigma}_u^2 c + \hat{\sigma}_e^2 d = n \quad ; n = \text{SSE} - n_0,$$

the estimators are given by

$$\hat{\sigma}_e^2 = \frac{mc - na}{bc - da} \tag{5.9}$$

and

$$\hat{\sigma}_u^2 = \frac{m - \hat{\sigma}_e^2 b}{a} \tag{5.10}$$

where

$$\begin{aligned}a &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q^L\right) \\ b &= \sum_q \text{tr}\left(\mathbf{B}_{qq}\right)\end{aligned}$$

$$
\begin{aligned}
c &= \sum_q \operatorname{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q^L\right) \\
d &= \sum_q \operatorname{tr}\left(\mathbf{C}_{qq}\right) \\
m &= \sum_q \sum_r \left(\mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^* - \mathbf{f}_q^\top \mathbf{E}_q \mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r\right) - \sum_q \operatorname{tr}(\mathbf{B}_{qq}\mathbf{V}_q) \\
n &= \sum_q \left\{\mathbf{y}_q^{*\top}\mathbf{C}_{qq}\mathbf{y}_q^* - \operatorname{tr}(\mathbf{C}_{qq}\mathbf{V}_q) - \mathbf{f}_q^\top\mathbf{E}_q\mathbf{C}_{qq}\mathbf{E}_q\mathbf{f}_q\right\}.
\end{aligned}
$$

Then, we follow Section 4.1 to obtain the variance of $\hat{\sigma}_e^2$ and $\hat{\sigma}_u^2$.

5.3.2 Pseudo-Maximum Likelihood (Pseudo-ML)

Since there are M_i repeated measures on the same individual, we cannot assume that these repeated measures are independent. However, the vectors of measurements in each block are assumed to be independent of one another. Hence, the log-likelihood function, l, can be expressed as

$$
l = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_q \log|\mathbf{\Sigma}_q| - \frac{1}{2}\left\{\sum_q (\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q)^\top \mathbf{\Sigma}_q^{-1}(\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q)\right\} \tag{5.11}
$$

Then, to maximize l, we differentiate (5.11), first with respect to $\boldsymbol{\beta}$ which yields

$$
\frac{\partial l}{\partial \boldsymbol{\beta}} = \boldsymbol{l}_{\boldsymbol{\beta}} = \sum_q \mathbf{X}_q^\top \mathbf{E}_q \mathbf{\Sigma}_q^{-1}(\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q). \tag{5.12}
$$

Again, we ignore the fact that $\mathbf{\Sigma}_q$ is a function of $\boldsymbol{\beta}$. Second, differentiating (5.11) with respect to σ_u^2 gives

$$
\frac{\partial l}{\partial \sigma_u^2} = \boldsymbol{l}_{\sigma_u^2} = -\frac{1}{2}\sum_q \left\{\operatorname{tr}\left(\mathbf{\Sigma}_q^{-1}\mathbf{K}_q^L\right) - (\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q)^\top \mathbf{\Sigma}_q^{-1}\mathbf{K}_q^L\mathbf{\Sigma}_q^{-1}(\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q)\right\}. \tag{5.13}
$$

Finally, differentiating (5.11) with respect to σ_e^2 gives

$$\frac{\partial l}{\partial \sigma_e^2} = \boldsymbol{l}_{\sigma_e^2} = -\frac{1}{2}\sum_q \left\{ \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\right) - (\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q)^\top \boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}(\mathbf{y}_q^* - \mathbf{E}_q\mathbf{f}_q) \right\}. \tag{5.14}$$

Then, we proceed with the same procedure as in Section 4.2 to obtain the information matrix. Thus, we finally obtain

$$\mathbf{I}\begin{bmatrix} \boldsymbol{\beta} \\ \sigma_u^2 \\ \sigma_e^2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2\sum_q \mathbf{X}_q^\top \mathbf{E}_q \boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) \\ \mathbf{0} & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\right) \end{bmatrix}.$$

Therefore, asymptotically,

$$\begin{aligned} Var\begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\sigma}_u^2 \\ \hat{\sigma}_e^2 \end{bmatrix} &\simeq \left(\mathbf{I}\begin{bmatrix} \boldsymbol{\beta} \\ \sigma_u^2 \\ \sigma_e^2 \end{bmatrix}\right)^{-1} \\ &= \begin{bmatrix} Var(\hat{\boldsymbol{\beta}}) & \mathbf{0} \\ \mathbf{0} & Var(\hat{\boldsymbol{\sigma}}^2) \end{bmatrix} \end{aligned}$$

where

$$Var(\hat{\boldsymbol{\beta}}) \simeq \left(\sum_q \mathbf{X}_q^\top \mathbf{E}_q \boldsymbol{\Sigma}_q^{-1}\mathbf{E}_q\mathbf{X}_q\right)^{-1} \tag{5.15}$$

and

$$Var(\hat{\boldsymbol{\sigma}}^2) \simeq 2\begin{bmatrix} \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) \\ \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q^{-1}\right) \end{bmatrix}^{-1}. \tag{5.16}$$

5.3.3 Pseudo-Restricted Maximum Likelihood (Pseudo-REML)

For pseudo-REML in this situation, we follow Section 4.3. That is, we have

$$\begin{aligned}\mathbf{I}\begin{bmatrix}\sigma_u^2\\ \sigma_e^2\end{bmatrix} &= -E\begin{bmatrix}\boldsymbol{l}_{\sigma_u^2\sigma_u^2} & \boldsymbol{l}_{\sigma_u^2\sigma_e^2}\\ \boldsymbol{l}_{\sigma_e^2\sigma_u^2} & \boldsymbol{l}_{\sigma_e^2\sigma_e^2}\end{bmatrix}\\ &= \frac{1}{2}\begin{bmatrix}\sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{K}_q^L\mathbf{M}_q\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\mathbf{K}_q^L\right)\\ \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\right)\end{bmatrix}\end{aligned} \tag{5.17}$$

and asymptotic variance-covariance matrix of the two estimators is given by

$$Var\begin{bmatrix}\hat{\sigma}_u^2\\ \hat{\sigma}_e^2\end{bmatrix} \simeq 2\begin{bmatrix}\sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{K}_q^L\mathbf{M}_q\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\mathbf{K}_q^L\right)\\ \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\mathbf{K}_q^L\right) & \sum_q \mathrm{tr}\left(\mathbf{M}_q\mathbf{M}_q\right)\end{bmatrix}^{-1}. \tag{5.18}$$

Chapter 6

Simulation Analysis

In previous chapters we have developed various bias adjusted estimators of the coefficient parameters and variance components in the linear mixed model. In this chapter we provide simulation results that allow one to compare the different estimators for fixed effects and variance components proposed in previous chapters under repeated application of probabilistic linkage based on the exchangeable linkage error model defined by equations (2.23) - (2.27).

6.1 Simulation of Linear Mixed Model for Cross-Sectional Linked Data

For each simulation, data were generated for a population of size $N = 800$ consisting of four blocks, each of size 200. Each block was made up of 50 groups with four subjects in each group. Values of X were independently drawn from the uniform distribution over [0,1] with corresponding values of Y given by

$$y_{ig} = 2 + 4x_{ig} + u_g + e_{ig} \tag{6.1}$$

where e_{ig} were independently drawn from the $N(0, 3)$ distribution and u_g were independently drawn from the $N(0, 1)$ distribution. The true data pairs (y_{ig}, x_{ig}) were randomly allocated to blocks and groups. Next, linked data pairs (y^*_{ig}, x_{ig}) were

generated by using the exchangeable linkage errors model defined by (2.23) - (2.27) with correct linkage probabilities $\lambda_1 = 1,\ \lambda_2 = 0.95,\ \lambda_3 = 0.85$ and $\lambda_4 = 0.75$. That is, all links for block 1 were assumed to be correct, while those for blocks 2, 3 and 4 were assumed to have some errors. We present simulation results for two scenarios. The first corresponds to known correct linkage probabilities. The second set of results were obtained using estimated linkage probabilities. These were calculated by taking random samples of $m_q = 25$ linked pairs from each of block 2, 3 and 4 and checking to see how many of these sampled links were correct. Following Chambers [7], the estimate of λ_q was then calculated as

$$\hat{\lambda}_q = \min\left\{m_q^{-1}(m_q - 0.5), \max(M_q^{-1}, l_q)\right\}$$

where l_q is the proportion of correctly linked pairs identified in the audit sample in block q.

Table 6.1: Simulation results for estimators of the regression coefficients of the linear mixed model

Estimator	Relative Bias		Relative RMSE	
	Intercept	Slope	Intercept	Slope
Scenario 1: Linkage Probabilities Correctly Specified				
TR	0.43	-0.29	17.53	18.50
Naïve	11.78	-11.64	24.21	30.05
R	0.70	-0.56	18.83	21.42
A	0.77	-0.62	18.75	21.23
C	0.82	-0.67	18.71	21.19
B	0.65	-0.50	19.06	21.90
MLE	0.70	-0.55	18.71	21.21
REML	0.70	-0.55	18.71	21.21
Scenario 2: Linkage Probabilities Estimated From Audit Sample				
TR	0.18	-0.03	18.50	19.13
Naïve	11.42	-11.26	24.83	30.03
R	-0.03	0.18	20.98	23.58
A	0.33	-0.18	20.78	23.19
C	0.42	-0.28	20.73	23.11
B	-0.46	0.60	21.44	24.46
MLE	0.30	-0.15	20.74	23.10
REML	0.30	-0.15	20.73	23.10

A total of 800 independent simulations were carried out. The `R` codes used to carry out these simulations are shown in Appendix D. Table 6.1 and Figure 6.1 show the empirical values of relative biases and relative root mean squared errors of the regression coefficient estimators described in Chapter 3, whereas Table 6.2 and Figure 6.2 show the empirical values of relative biases and relative root mean squared errors of the different variance components estimators described in Chapter 4. The WLS estimator TR based on perfectly linked data and the naive WLS estimator based on the actual linked data were obtained using the default settings of the `lme` function in the R software package. Note that variance components estimators obtained using the ANOVA method are functions of $\boldsymbol{\beta}$. The estimator R, A, C and B shown in Table 6.2 are the ANOVA estimators obtained using those coefficient estimators corresponding to equations (3.8), (3.16), (3.17) and (3.22), respectively. The actual coverages of the nominal 95% confidence intervals for all the model parameters are shown in Table 6.3.

Table 6.2: Simulation results for estimators of the variance components of the linear mixed model

Estimator	Relative Bias		Relative RMSE	
	Between-Group	Within-Group	Between-Group	Within-Group
Scenario 1: Linkage Probabilities Correctly Specified				
TR	0.79	-0.28	30.46	15.00
Naïve	-20.43	5.18	34.00	22.31
R	1.23	-0.42	36.24	16.77
A	1.24	-0.39	36.24	16.71
C	1.24	-0.38	36.24	16.72
B	1.23	-0.44	36.24	16.92
MLE	-2.80	-0.29	33.40	16.43
REML	0.95	-0.24	33.93	16.44
Scenario 2: Linkage Probabilities Estimated From Audit Sample				
TR	1.25	0.19	32.06	15.92
Naïve	-20.59	5.71	34.83	24.43
R	1.33	-0.07	38.38	18.89
A	1.33	0.04	38.38	18.71
C	1.33	0.07	38.38	18.70
B	1.33	-0.21	38.38	19.25
MLE	-3.31	0.24	34.95	18.24
REML	0.42	0.29	35.46	18.28

The results set out in Table 1 show that the naive WLS estimator that assumes the data are perfectly linked is clearly biased. Since linkage error is a particular type of measurement error, this bias attenuates the estimate of the slope parameter and exaggerates that of the intercept. On the other hand, all four of the adjusted estimators correct this bias, with the REML estimator the most efficient. The results are unchanged under Scenario 2 where linkage probabilities were estimated by taking small audit samples.

The results displayed in Table 6.2 show that the naive variance components estimators that treat the linkage as perfect are also biased. As expected, the estimator obtained using the ML approach is slightly biased. All of the remaining adjusted estimators are essentially unbiased, with REML being the most efficient. Again, the results under Scenario 2 are in the same direction of those under Scenario 1.

Table 6.3: Actual coverages of nominal 95% confidence intervals for the parameters of the linear mixed model

Estimator	Coverage			
	Intercept	Slope	Between-Group	Within-Group
Scenario 1: Linkage Probabilities Correctly Specified				
TR	96.6	94.1	97.4	96.0
Naïve	84.6	77.1	98.4	85.0
R	96.6	94.9	93.4	97.4
A	96.4	95.2	93.4	97.5
C	96.4	95.4	93.4	97.4
B	96.5	94.5	93.4	96.8
MLE	96.2	95.5	98.0	95.9
REML	96.2	95.5	97.6	95.9
Scenario 2: Linkage Probabilities Estimated From Audit Sample				
TR	94.0	95.0	96.2	93.9
Naïve	85.1	77.2	98.1	78.9
R	94.6	94.9	91.6	94.8
A	94.8	94.8	91.6	94.6
C	94.9	94.4	91.6	94.5
B	94.5	95.0	91.6	94.6
MLE	94.1	94.5	97.6	92.4
REML	94.9	94.6	96.9	92.5

Finally, we note that the results displayed in Table 6.3 show that variance estimators that allow for the extra variability, $\sum_q \sum_r \hat{\mathbf{D}}_{qr} \hat{\Delta}_r \hat{\mathbf{D}}_{qr}^\top$ defined in equation (3.25), induced by estimation of the correct linkage probabilities lead to confidence intervals with good coverage properties. If this extra variability was ignored and not taken into account, confidence intervals would not give good coverage properties.

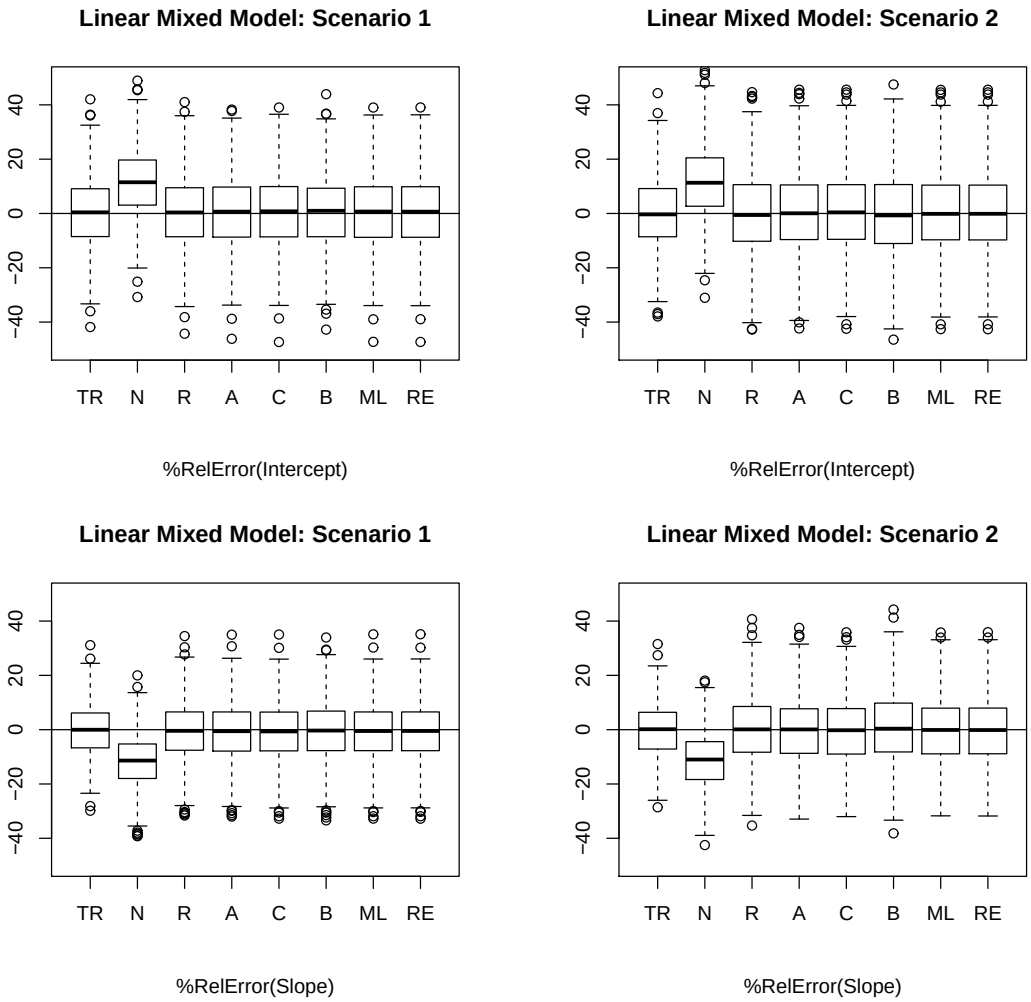

Figure 6.1: Boxplots of percentage relative errors of coefficient parameters generated by different estimators in linear mixed model simulations. Note that N is the Naïve estimator while RE is the REML estimator.

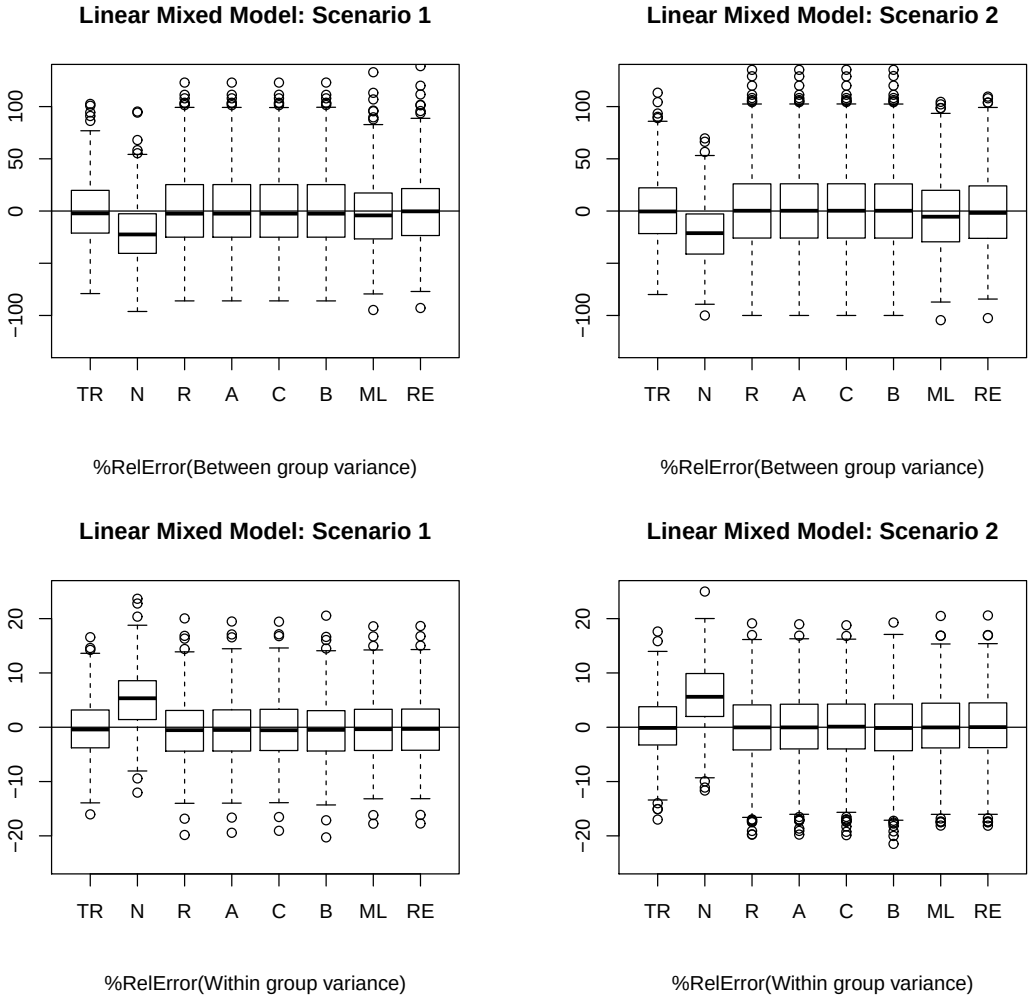

Figure 6.2: Boxplots of percentage relative errors of variance components parameters generated by different estimators in linear mixed model simulations. Note that N is the Naïve estimator while RE is the REML estimator.

6.2 Simulation of Linear Mixed Model for Longitudinal Linked Data

In this case, data were generated for 350 subjects in each register which consists of three blocks of size 130, 120 and 100 in block 1, 2 and 3 respectively, and there were four registers all together. Measurements in each register were supposed to be collected over time i.e. values of X at register 1, denoted by X_1, were independently drawn from the uniform distribution over [0,1]. Then, values of X were increased

sequentially by one in the next register with corresponding values of Y given by

$$y_{ij} = 2 + 4x_{ij} + u_i + e_{ij} \tag{6.2}$$

where e_{ij} were independently drawn from the $N(0,1)$ distribution and u_i were independently drawn from the $N(0,3)$ distribution. In this case we assume that data were linked from register 1 to other registers i.e. we have true data pairs (x_{i1}, x_{it}) and (y_{i1}, y_{it}). Linked data pairs (x_{i1}, x^*_{it}) and (y_{i1}, y^*_{it}) were then generated by using the exchangeable linkage errors model defined by (2.23) - (2.27) with correct linkage probabilities $\lambda_{tq} = 0.95, 0.9$ and 0.85 for block 1, 2 and 3 respectively. Note that register 1 was regarded as a benchmark data set. Here we present simulation results for two scenarios. The first set of results were obtained from known linkage probabilities. The second set of results were obtained from estimated linkage probabilities by taking random samples of $m_q = 25$ linked pairs from each block and checking to see whether these sampled links were correct.

Table 6.4: Simulation results for estimators of the regression coefficients of the linear mixed model for longitudinal registers

Estimator	Relative Bias		Relative RMSE	
	Intercept	Slope	Intercept	Slope
Scenario 1: Linkage Probabilities Correctly Specified				
TR	0.48	-0.02	11.49	1.19
Naïve	0.51	-0.03	11.56	1.29
ANOVA	0.49	-0.02	11.51	1.22
MLE	0.48	-0.02	11.50	1.20
REML	0.49	-0.02	11.50	1.22
Scenario 2: Linkage Probabilities Estimated				
TR	-0.19	0.04	12.39	1.15
Naïve	-0.18	0.04	12.39	1.22
ANOVA	-0.21	0.05	12.43	1.22
MLE	-0.17	0.04	12.43	1.15
REML	-0.19	0.04	12.40	1.17

A total of 800 independent simulations were carried out. Table 6.4 illustrates the empirical values of relative biases and relative root mean squared errors of the coefficient estimators described in Chapter 5 whereas Table 6.5 reveals the empirical

Table 6.5: Simulation results for estimators of the variance components of the linear mixed model for longitudinal registers

Estimator	Relative Bias		Relative RMSE	
	Between-Group	Within-Group	Between-Group	Within-Group
Scenario 1: Linkage Probabilities Correctly Specified				
TR	0.15	0.26	23.52	4.31
Naïve	-13.29	121.01	45.48	123.27
ANOVA	0.68	-4.90	28.38	27.86
MLE	0.15	-3.99	26.85	18.12
REML	0.22	-3.64	26.96	18.13
Scenario 2: Linkage Probabilities Estimated				
TR	-0.09	0.15	22.88	4.40
Naïve	-13.63	121.79	45.92	124.20
ANOVA	0.51	-3.97	33.06	54.16
MLE	-5.42	18.18	34.81	38.77
REML	-5.36	18.54	34.84	39.10

Table 6.6: Actual coverages of nominal 95% confidence intervals for the parameters of the linear mixed model for longitudinal registers

Estimator	Coverage			
	Intercept	Slope	Between-Group	Within-Group
Scenario 1: Linkage Probabilities Correctly Specified				
TR	95.1	95.5	95.2	95.4
Naïve	94.0	99.1	56.2	0.0
ANOVA	94.6	93.4	92.7	99.5
MLE	94.9	95.6	87.1	73.5
REML	94.2	94.2	87.0	73.6
Scenario 2: Linkage Probabilities Estimated				
TR	93.9	96.6	94.5	94.4
Naïve	94.2	99.6	54.5	0.0
ANOVA	94.1	86.0	87.9	80.1
MLE	93.9	97.4	72.9	35.6
REML	93.8	96.5	73.4	35.6

values of relative biases and relative root mean squared errors of variance components estimators described in the same chapter. The actual coverages of the nominal 95% confidence intervals for all of the model parameters are illustrated in Table 6.6. Figure 6.3 shows boxplots of the distribution of relative errors for the various coefficient estimators whereas those for variance component estimators are exhibited in Figure 6.4.

The results set out in Table 6.4 show that all of the estimators were unbiased and there was no difference among the estimates, including the naive WLS estimator. Under Scenario 2 where linkage probabilities were estimated, we see that the results were similar to those for Scenario 1. Investigation of the results displayed in Table

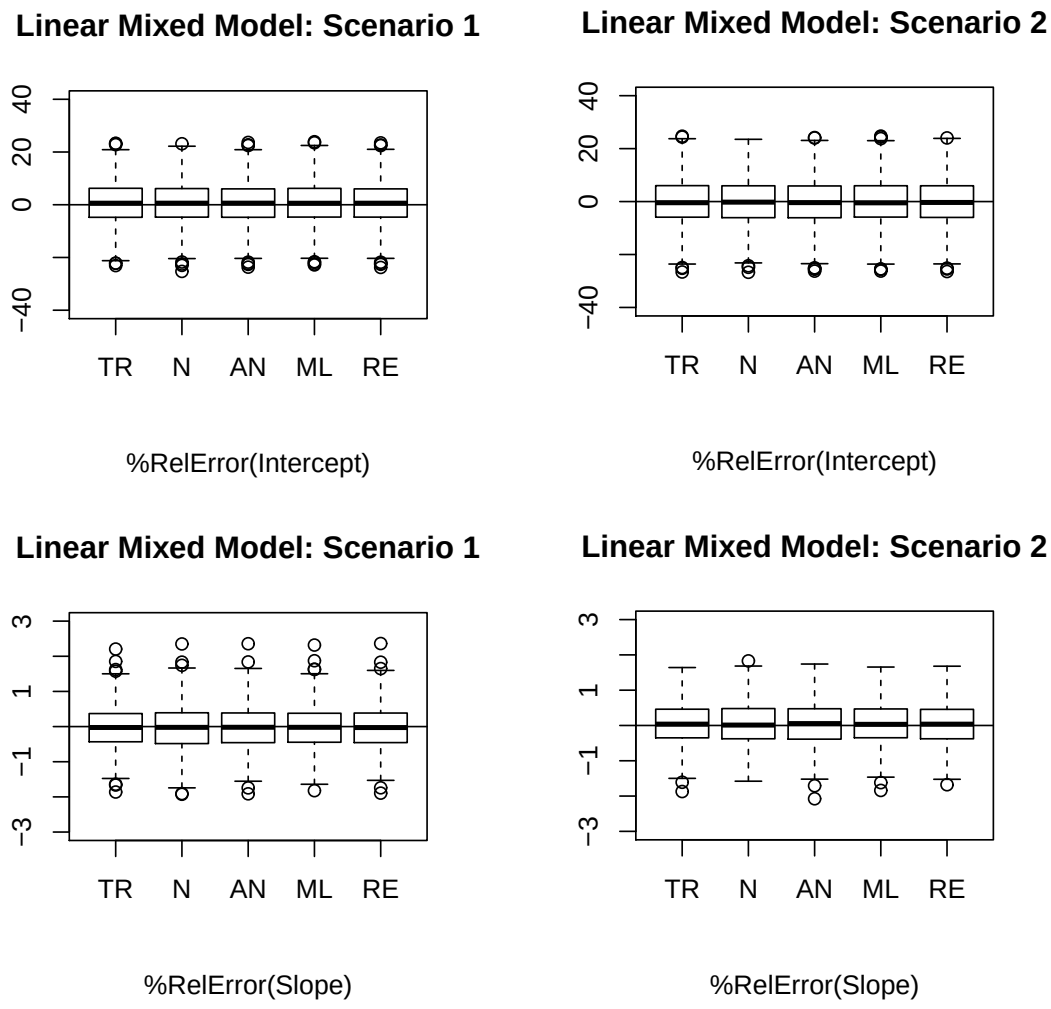

Figure 6.3: Boxplots of percentage relative errors of coefficient parameters generated by different estimators in linear mixed model simulations from longitudinal registers

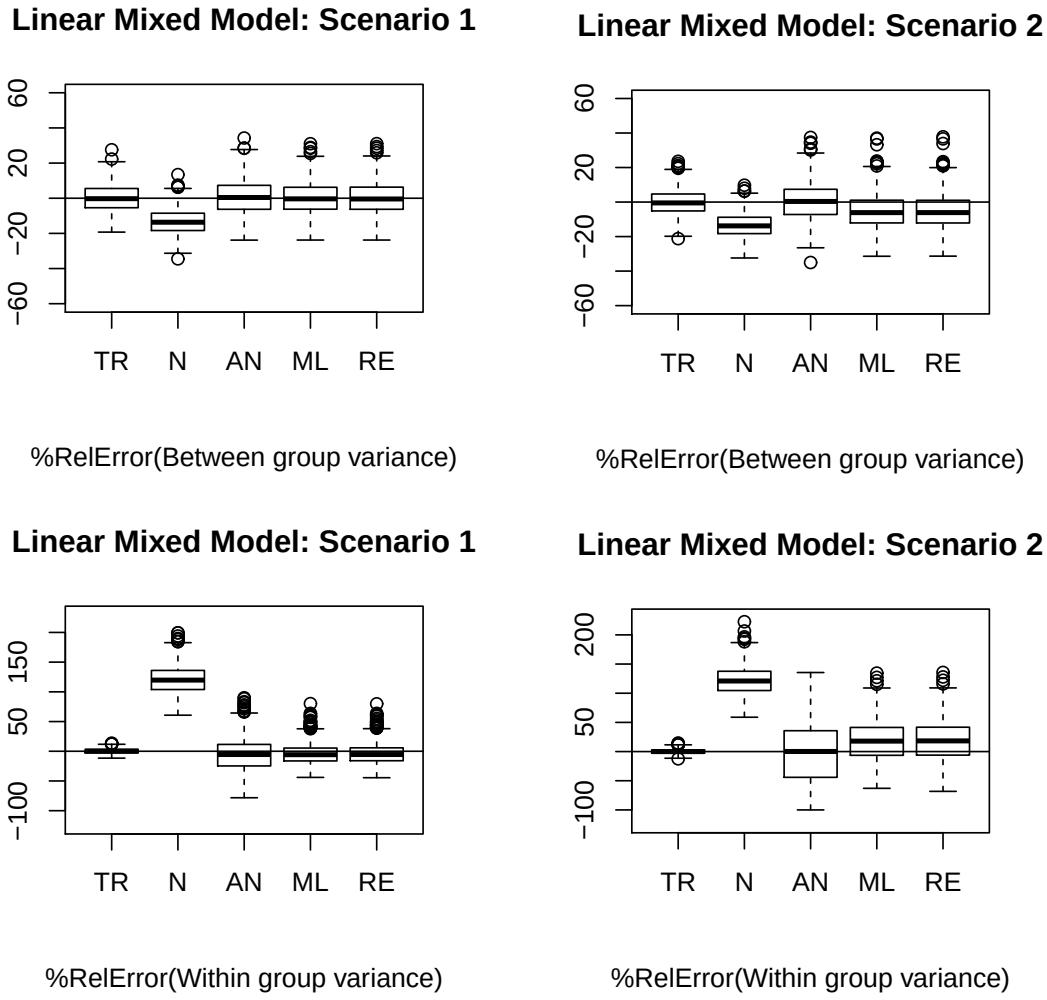

Figure 6.4: Boxplots of percentage relative errors of variance components parameters generated by different estimators in linear mixed model simulations from longitudinal registers

6.5 and Figure 6.4 shows however that the naive variance components estimators that just used the linked data were clearly biased, especially the naive within group variance estimator. All three of the adjusted estimators correct this bias, with MLE and REML being the most efficient under Scenario 1. Notice that although bias was corrected, the adjusted within group variance estimators are still a little bit biased. However, compared to the naive estimator it is worth correcting the bias. One issue arisen from simulation experiments is the sample size. Even though not reported here, the results showed that as sample sizes get larger, the adjusted estimators tend to be less biased.

The results under Scenario 2 were quite different to those for Scenario 1. We see that the estimators obtained by the likelihood approaches are more biased than the ANOVA estimator. Once again, although not reported here, simulation experiments showed that aside from the sample size issue, number of estimated parameters $\hat{\lambda}_{tq}$ is another issue that we have to be aware of in this case. This is because when the inference was carried out, the estimated values of λ_{tq} were treated as fixed and this might be a reason for biased results.

The results displayed in Table 6.6 illustrates that the ML and REML methods of estimation have led to confidence intervals with relatively poor coverage properties, especially when linkage probabilities were estimated. This has led to a further investigation regarding the variance estimator of all parameters. The results shown in Table 6.7 illustrates that the variance estimator of coefficient $\boldsymbol{\beta}$ obtained in Chapter 5 are generally unbiased. However, the variance estimator of the slope parameter based on ANOVA has higher variability compared to the others. This might explain why the slope parameter has poorer coverage than the others in Table 6.6.

Examination of the results displayed in Table 6.8 indicates that the variance estimator of variance components are biased, especially the likelihood approaches. Again, this might explain why variance components estimators had such poor coverage properties and this is another issue that needs further research.

Table 6.7: Bias and standard error of standard error estimators of the regression coefficients estimators when linkage probabilities were estimated from longitudinal registers

Estimator	Bias		Standard error	
	Intercept	Slope	Intercept	Slope
TR	-0.006	0.001	0.006	0.001
Naïve	-0.006	0.011	0.006	0.002
ANOVA	-0.006	-0.002	0.007	0.008
MLE	-0.009	0.004	0.008	0.003
REML	-0.008	0.002	0.008	0.004

Table 6.8: Bias and standard error of standard error estimators of the variance components estimators when linkage probabilities were estimated from longitudinal registers

Estimator	Bias		Standard error	
	Between-Group	Within-Group	Between-Group	Within-Group
TR	0.013	0.000	0.052	0.002
Naïve	0.052	-0.178	0.050	0.004
ANOVA	-0.196	-0.159	0.072	0.061
MLE	-0.322	-0.242	0.097	0.014
REML	-0.323	-0.243	0.098	0.014

6.3 Sensitivity of Correct Linkage Probabilities

We can see clearly from the results so far that the probability of correct linkage, λ, is a key parameter of the inference. In this section, we focus on the effects of variations of λ on the bias and variability of all estimators obtained from a linear mixed model.

To determine such effects, data were generated in the same way as in Section 6.1. For simplicity in comparison, however, we assume that blocks 2, 3 and 4 had the same values of λ. Three different values of $\hat{\lambda}$ were compared, $\hat{\lambda}$= 0.85, 0.75 and 0.65 under the true value of λ=0.75.

Table 6.9 sets out the empirical values of relative biases and relative root mean squared errors of the regression coefficients estimators at the three different values of $\hat{\lambda}$. Likewise, Table 6.10 reveals the empirical values of relative biases and relative root mean squared errors of variance components estimators at those different values of $\hat{\lambda}$. Table 6.11 and Table 6.12 illustrate the actual coverages of the nominal 95% confidence intervals for the regression coefficients and variance components parameters respectively.

Examination of the results set out in Table 6.9 shows that when λ were incorrectly specified, all adjusted estimators of the regression coefficients were biased. In particular, when λ were overestimated which is common in practice, the adjusted estimators were biased towards the naive estimate. On the other hand, when λ were underestimated, the bias moved towards the opposite direction. The variability of

all estimators also increased when λ were incorrectly specified.

Table 6.9: Simulation results for estimators of the regression coefficients under true value of $\lambda = 0.75$

Relative Bias						
Estimator	Intercept			Slope		
	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$
TR	-0.17	-0.17	-0.17	-0.11	-0.11	-0.11
Naïve	19.14	19.14	19.14	-19.46	-19.46	-19.46
R	9.09	0.73	-9.34	-9.39	-1.02	9.06
A	8.51	0.38	-8.33	-8.81	-0.67	8.05
C	8.34	0.30	-7.89	-8.63	-0.58	7.63
B	9.61	0.99	-10.29	-9.91	-1.28	10.02
MLE	8.19	0.15	-8.06	-8.49	-0.43	7.79
REML	8.19	0.15	-8.05	-8.49	-0.43	7.79
Relative RMSE						
TR	17.24	17.24	17.24	17.99	17.99	17.99
Naïve	31.69	31.69	31.69	42.66	42.66	42.66
R	21.65	18.31	23.46	27.11	21.41	29.66
A	21.19	18.29	22.61	26.25	21.10	27.98
C	21.15	18.48	22.49	26.06	21.17	27.53
B	22.14	18.60	24.82	27.98	22.10	32.00
MLE	21.05	18.49	22.63	25.91	21.21	27.75
REML	21.04	18.48	22.62	25.90	21.21	27.75

The results set out in Table 6.10 are also in the same direction as those in Table 6.9. However, note that when λ were overestimated, the variability of the between group variance estimators seemed to slightly decrease but obviously increase when λ were underestimated.

Table 6.10: Simulation results for estimators of the variance components under true value of $\lambda = 0.75$

Relative Bias						
Estimator	Between-Group			Within-Group		
	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$
TR	-1.99	-1.99	-1.99	-0.02	-0.02	-0.02
Naïve	-34.27	-34.27	-34.27	8.65	8.65	8.65
R	-16.96	-0.72	20.86	3.99	-0.15	-5.63
A	-16.96	-0.73	20.86	3.84	-0.24	-5.30
C	-16.96	-0.72	20.87	3.79	-0.27	-5.16
B	-16.95	-0.72	20.86	4.13	-0.08	-5.97
MLE	-18.10	-4.04	9.55	3.65	-0.16	-4.15
REML	-14.48	0.36	14.63	3.69	-0.20	-4.26
Relative RMSE						
TR	30.87	30.87	30.87	15.69	15.69	15.69
Naïve	42.42	42.42	42.42	30.65	30.65	30.65
R	34.78	36.30	48.84	20.34	17.00	24.85
A	34.79	36.30	48.85	20.08	17.05	24.20
C	34.79	36.31	48.86	20.01	17.07	23.94
B	34.78	36.29	48.83	20.60	17.08	25.77
MLE	35.59	35.75	41.11	19.87	17.13	21.84
REML	34.45	36.16	43.10	19.95	17.14	22.04

Finally, from Table 6.11 we see that incorrectly specified λ can lead to confidence intervals with poorer coverage properties for the coefficient parameters. We see that the average coverage rates for the intercept parameter obtained by ANOVA method when $\hat{\lambda} = 0.85, 0.75, 0.65$ are 93.5, 98.1 and 94.4 respectively where those obtained by the likelihood approaches are 94.0, 97.7 and 94.7 respectively. For the slope parameter, the average coverage rates obtained by ANOVA method are 89.0, 97.1 and 93.2 where those obtained by the likelihood approaches are 90.3, 97.7 and 93.1 respectively. Obviously, the average coverage rates are highest when λ were correctly identified and decrease when λ were incorrectly identified.

Nevertheless, the results in Table 6.12 are quite different. For the between group variance, variance estimators of the ANOVA approach lead to poorer coverages when λ were overestimated whereas those for the ML and REML approaches remain very

close to results when λ were correctly identified. That is, the average coverage rates for the ANOVA approach are 86.6, 94.5 and 97.2 where those for the likelihood approaches are 98.9, 97.2 and 94.8 respectively. For the within group variance, variance estimators of all adjusted estimators lead to poorer coverages when λ were underestimated. That is, the average coverage rates for the ANOVA approach are 96.3, 98.3 and 88.7 where those for the likelihood approaches are 92.2, 95.8 and 88.8 respectively.

Table 6.11: Actual coverages of nominal 95% confidence intervals for the regression coefficients estimators of the linear mixed model under true value of $\lambda = 0.75$

Estimator	Intercept			Slope		
	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$
TR	94.9	94.9	94.9	97.2	97.2	97.2
Naïve	66.4	66.4	66.4	48.4	48.4	48.4
R	94.0	98.2	94.5	88.5	96.8	93.5
A	93.5	97.7	94.0	90.3	97.2	93.5
C	93.5	98.2	94.9	90.3	97.7	93.1
B	93.1	98.2	94.0	87.1	96.8	92.6
MLE	94.0	97.7	94.5	90.3	97.7	93.1
REML	94.0	97.7	94.9	90.3	97.7	93.1

Table 6.12: Actual coverages of nominal 95% confidence intervals for the variance components estimators of the linear mixed model under true value of $\lambda = 0.75$

Estimator	Between-Group			Within-Group		
	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$	$\hat{\lambda} = 0.85$	$\hat{\lambda} = 0.75$	$\hat{\lambda} = 0.65$
TR	96.8	96.8	96.8	95.9	95.9	95.9
Naïve	96.7	96.7	96.7	60.4	60.4	60.4
R	86.6	94.5	97.2	96.3	98.2	88.9
A	86.6	94.5	97.2	96.3	98.6	88.5
C	86.6	94.5	97.2	96.3	98.6	89.4
B	86.6	94.5	97.2	96.3	97.7	88.0
MLE	99.1	97.2	94.9	92.2	96.8	89.4
REML	98.7	97.2	94.6	92.2	94.9	88.3

Chapter 7

Conclusions and Further Research

7.1 Summary

Record linkage is the process of bringing together of information from two records that are believed to relate to the same unit of population. Record linkage is widely used by both businesses and government agencies which are frequently concerned with large-scale sample surveys and censuses. Computerized record linkage models not only increase coverage and reduce the number of duplicate records, they also reduce the number of hours required for clerical review and cleanup and therefore reduce the overall cost of a survey or census.

Analyzing a linked data set based on the assumption that the linkage is perfect can lead to potential bias induced by linkage errors in the data set. In this thesis we developed framework for statistical analysis of the linked data. In particular, we focused on development of methodologies for efficient fitting of linear mixed models to the linked data, since, in reality, measurements are usually made on clusters of correlated statistical units and when analyzing such data linear mixed models are often used. An example of structure of the data was given in Chapter 3.

The aim of this thesis was to provide appropriate modifications to standard statistical analysis methods to ensure that they remain unbiased when used with probabilistically linked data. Since the linear mixed models have been the focus of inference, there were two types of parameters that need to be estimated: namely,

coefficient and variance components estimators. To achieve the aim, various bias adjusted methods have been used. In Chapter 3, we focused on bias correction for regression coefficient estimator. In this chapter, we developed four methods in order to achieve the aim. To calculate such estimators, one would need to obtain a variance-covariance matrix of the linked data. Consequently, in Chapter 4 we focus on bias correction for variance components estimators. In particular, we focused on three widely used methods: analysis of variance (ANOVA), maximum likelihood (ML) and restricted maximum likelihood (REML). However, some properties of the last two methods could not be satisfied. Thus, we referred to these methods as "pseudo-ML" and "pseudo-REML".

Once the framework was established, an empirical study under two scenarios comparing the performance of the bias adjusted estimators were carried out in Chapter 6. The first scenario was where the linkage probabilities were correctly specified. The second was where they were estimated which is more realistic. The computer package R was used throughout the simulation analysis. To summarize results:

1. When linkage errors are ignored, the weighted least squares estimator as well as the variance components estimators that just use the linked data are clearly biased.

2. As far as the estimation of regression coefficient parameter is concerned, the estimator obtained by the pseudo-ML and pseudo-REML approaches seem to perform better than the others.

3. The variance components estimator obtained using the pseudo-ML approach was slightly biased whereas all of the other adjusted estimators were unbiased, with pseudo-REML being the most efficient.

4. There is not much difference between pseudo-ML estimates and ML estimates obtained using `optim` function in `R`. Moreover, by using pseudo-ML method we can obtain the standard errors of the estimators and therefore the confidence intervals of the estimators. On the other hand, `optim` function cannot provide

the standard errors of the estimators and therefore the confidence intervals of the estimators. It is also time-consuming compared to the pseudo-ML method due to numerically intensive computation.

5. Even if all the methods developed here work reasonably well, they also show evidence of increased variability due to application of linear mixed models to the linked data.

These methods were then applied to longitudinal modeling in Chapter 5. In this chapter we considered a situation where registers are linked over time to create longitudinal data. Linear mixed models were still employed except that linkage errors occur on both response and explanatory variables.

A simulation study was then carried out in Chapter 6. The simulation results illustrated that all of the coefficient estimators were unbiased and there was no difference among the estimates, including the naive WLS estimator. This is not a surprise since linkage errors occur simultaneously on both explanatory variable and response variable. However, when we examined the results for variance components estimation there are some issues that need to be discussed.

Investigation of the variance components estimation results showed that the naive variance components estimators that just used the linked data were clearly biased, especially the naive within group variance estimator. All three of the adjusted estimators correct this bias, with MLE and REML being the most efficient under Scenario 1 where linkage probabilities were assumed to be known. Notice that although bias was corrected, the adjusted within group variance estimators were still a little bit biased. However, compared to the naive estimator it is worth correcting the bias.

The results under Scenario 2 where linkage probabilities were estimated were different to those for Scenario 1. We see that the estimators obtained by the likelihood approaches performed poorer than the ANOVA estimator. Furthermore, they also showed poor coverage properties since the variance estimator of variance components are biased. This issue will be discussed in the next section.

In conclusion, applying all methods proposed in this thesis to linked data allows bias induced by linkage errors to be corrected, both on coefficient estimator and variance component estimators.

7.2 Further Research

Although a framework for statistical analysis of the linked data has been well developed, there are some issues that require further research.

1. The exchangeable linkage error model used throughout this thesis may be too simple for many linkage situations. This depends on how much information available from the linkage process. In some situations, if more information is available, it might be more appropriate to formulate more complex models for linkage error.

2. In this thesis, the simplest form of linear mixed models was employed i.e. it has been assumed that the variance components of the linked data are constant across data set. In fact, this may not be the case since measurement variability of population in different groups might be different to one another.

3. The theory set out in Chapter 5 was based on balanced data only i.e. we assumed that each individual has the same sample size of repeated measurements and therefore the linkage is complete. However, one issue in the analysis of longitudinal data that requires attention is the nature of any missing data. That is, the linkage can be incomplete and consequently, there is a need to develop theory set out in this chapter to account for this kind of data.

4. When the parameters of the models were estimated based on linkage probabilities estimates in Chapter 5, the extra uncertainty arising from linkage probabilities estimation were not allowed for in the variance estimator of the parameters. It is far too complicated to take into account this uncertainty

analytically. A numerical method such as Monte Carlo could be further investigated.

5. Variance components estimation, especially the likelihood approach, has been developed based on the assumption that the variance components estimates follow log-normal distribution when in fact they rather follow chi-square distribution. This assumption has affected the parameters estimates in Chapter 5, particularly the confidence intervals. This estimation could be further developed by applying the bootstrap method which empirically estimates the sampling distribution of variance components.

6. The simulation study in this thesis was carried out on the basis of small sample size compared to the real life data. Even such a small sample size, the computation was relatively intensive. Application of all the methods we propose to real life linked data is required to see whether there is any computational problem in reality. If so, it is obvious that the computation needs to be improved. If not, it is also necessary to determine whether the results provided are in the similar way as those in the simulation study.

7. A key parameter used in the statistical analysis of linked data is the probability of correct linkage. In real life, this parameter is unknown and needs to be estimated. From Chapter 6, the sensitivity analysis of this probability has shown that inaccurate estimates of this parameter can result in biased inference. In this thesis, the estimation of this parameter has been developed by taking random samples from population and checking the correct sample links. In practice, however, this procedure will be relatively costly and time-consuming. On the other hand, if one knows the linkage process especially the levels of error associated with a linkage rule, the probability of correct linkage can be obtained directly by one minus probability of unmatched pair decision given the pair are in fact matched.

8. The developed methods so far assumed that the linked datasets cover the

same population. In practice linked datasets do not always have the same target population since the groupings in one population need not necessarily be the same as those in the other population or populations. Therefore, further research for this situation is required.

Appendix A

Approximating the K matrix

$$\mathbf{K}_q = E_X\left(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top\right)$$

To illustrate, suppose there are three groups in block q, then

$$\mathbf{A}_q = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix},\ \mathbf{A}_q^\top = \begin{bmatrix} \mathbf{A}_{11}^\top & \mathbf{A}_{21}^\top & \mathbf{A}_{31}^\top \\ \mathbf{A}_{12}^\top & \mathbf{A}_{22}^\top & \mathbf{A}_{32}^\top \\ \mathbf{A}_{13}^\top & \mathbf{A}_{23}^\top & \mathbf{A}_{33}^\top \end{bmatrix}$$

and

$$\mathbf{Z}_q\mathbf{Z}_q^\top = \begin{bmatrix} \mathbf{1}_1\mathbf{1}_1^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_2\mathbf{1}_2^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_3\mathbf{1}_3^\top \end{bmatrix}.$$

Therefore,

$$\mathbf{K}_q = E_X\left(\mathbf{A}_q\mathbf{Z}_q\mathbf{Z}_q^\top\mathbf{A}_q^\top\right) = E_X\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{K}_{23} \\ \mathbf{K}_{31} & \mathbf{K}_{32} & \mathbf{K}_{33} \end{bmatrix}$$

where

$$
\begin{aligned}
\mathbf{K}_{11} &= \mathbf{A}_{11}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{11}^\top + \mathbf{A}_{12}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{12}^\top + \mathbf{A}_{13}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{13}^\top \\
\mathbf{K}_{12} &= \mathbf{A}_{11}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{21}^\top + \mathbf{A}_{12}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{22}^\top + \mathbf{A}_{13}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{23}^\top \\
\mathbf{K}_{13} &= \mathbf{A}_{11}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{31}^\top + \mathbf{A}_{12}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{32}^\top + \mathbf{A}_{13}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{33}^\top \\
\mathbf{K}_{21} &= \mathbf{A}_{21}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{11}^\top + \mathbf{A}_{22}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{12}^\top + \mathbf{A}_{23}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{13}^\top \\
\mathbf{K}_{22} &= \mathbf{A}_{21}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{21}^\top + \mathbf{A}_{22}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{22}^\top + \mathbf{A}_{23}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{23}^\top \\
\mathbf{K}_{23} &= \mathbf{A}_{21}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{31}^\top + \mathbf{A}_{22}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{32}^\top + \mathbf{A}_{23}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{33}^\top \\
\mathbf{K}_{31} &= \mathbf{A}_{31}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{11}^\top + \mathbf{A}_{32}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{12}^\top + \mathbf{A}_{33}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{13}^\top \\
\mathbf{K}_{32} &= \mathbf{A}_{31}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{21}^\top + \mathbf{A}_{32}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{22}^\top + \mathbf{A}_{33}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{23}^\top \\
\mathbf{K}_{33} &= \mathbf{A}_{31}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{31}^\top + \mathbf{A}_{32}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{32}^\top + \mathbf{A}_{33}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{33}^\top.
\end{aligned}
$$

In general, $\mathbf{K}_q$ consists of five cases of $\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top$ as follows:

Case 1: $g = h = u = v$ e.g. $\mathbf{A}_{11}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{11}^\top$

Case 2: $g \neq h$ but $g = u,\ h = v$ e.g. $\mathbf{A}_{12}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{12}^\top$

Case 3: $g = h$ but $u \neq v$ e.g. $\mathbf{A}_{11}\mathbf{1}_1\mathbf{1}_1^\top\mathbf{A}_{21}^\top$

Case 4: $g \neq h$ but $u = v$ e.g. $\mathbf{A}_{12}\mathbf{1}_2\mathbf{1}_2^\top\mathbf{A}_{22}^\top$

Case 5: $g \neq h$, $u \neq v$ and ($g \neq u$ or $h \neq v$) e.g. $\mathbf{A}_{13}\mathbf{1}_3\mathbf{1}_3^\top\mathbf{A}_{23}^\top$.

If there are G groups in block q,

$$
\begin{aligned}
E_X(\mathbf{K}_{ss}) &= E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top : \text{Case1}) + (G-1)E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top : \text{Case2}) \\
E_X(\mathbf{K}_{st}) &= E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top : \text{Case3}) + E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top : \text{Case4}) \\
&\quad + (G-2)E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top : \text{Case5}).
\end{aligned}
$$

Here, we have to approximate $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ separately in five cases. We will use the method to approximate the $\mathbf{V}_q$ matrix in Chambers [7] to obtain these expectations.

Case 1: $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ when $g = h = u = v$

$$\begin{aligned} E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top) &= \left[E\left\{\sum_{k=1}^{n_{qh}} a_{ik}\sum_{l=1}^{n_{qh}} a_{jl}\right\}\right] \\ &= \left[\sum_{k=1}^{n_{qh}}\sum_{l=1}^{n_{qh}} E(a_{ik}a_{jl})\right] \\ &= \left[\sum_{k=1}^{n_{qh}}\sum_{l=1}^{n_{qh}} \Pr(a_{ik}=1,\ a_{jl}=1)\right] \end{aligned}$$

with

$$\begin{aligned} \Pr(a_{ik}=1,\ a_{jl}=1) &= \Pr(a_{jl}=1 \mid\ a_{ik}=1)\Pr(a_{ik}=1) \\ &= \Pr(a_{jl}=1 \mid\ a_{ik}=1)\ e_{ik}. \end{aligned}$$

When we assume an exchangeable first order linkage structure,

$$e_{ik} = \lambda I(i=k) + \frac{(1-\lambda)}{M_q - 1} I(i \neq k).$$

Once again, to extend this first order model to one for an exchangeable second order linkage error structure, we have to assume that M is large enough so that the linkage errors associated with two distinct records can be treated as approximately

independent as follows.

$$\begin{aligned}\Pr(a_{jl}=1|\ a_{ik}=1) &= I(i=j)I(k=l) \\ &\quad + I(i\neq j)I(k\neq l)\left\{\lambda I(j=l)+\frac{(1-\lambda)}{M_q-1}I(j\neq l)\right\} \\ &= I(i=j)I(k=l)+e_{jl|ik}\end{aligned}$$

Let $E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)=\mathbf{Z}=[z_{ij}]$. Then

$$\begin{aligned}z_{ii}=\sum_{k=1}^{n_{qh}}e_{ik} &= \lambda+\sum_{k\neq i}^{n_{qh}}\frac{(1-\lambda)}{M_q-1} \\ &= \lambda+(n_{qh}-1)\frac{(1-\lambda)}{M_q-1} \\ &= \lambda+O(M_q^{-1})\end{aligned}$$

and, for $i\neq j$,

$$\begin{aligned}z_{ij} &= \sum_{k=1}^{n_{qh}}\sum_{l=1}^{n_{qh}}e_{ik}e_{jl|ik} \\ &= \sum_{k=1}^{n_{qh}}\sum_{l=1}^{n_{qh}}\left\{\lambda I(i=k)+\frac{(1-\lambda)}{M_q-1}I(i\neq k)\right\} \\ &\quad \left[I(i\neq j)I(k\neq l)\left\{\lambda I(j=l)+\frac{(1-\lambda)}{M_q-1}I(j\neq l)\right\}\right] \\ &= \lambda^2+\sum_{k\neq i,j}^{n_{qh}}\lambda\frac{(1-\lambda)}{M_q-1}+\sum_{l\neq i,j}^{n_{qh}}\lambda\frac{(1-\lambda)}{M_q-1}+\sum_{k\neq i}^{n_{qh}}\sum_{l\neq j,k}^{n_{qh}}\left\{\frac{(1-\lambda)}{M_q-1}\right\}^2 \\ &= \lambda^2+(n_{qh}-2)\lambda\frac{(1-\lambda)}{M_q-1}+(n_{qh}-2)\lambda\frac{(1-\lambda)}{M_q-1}+\{(n_{qh}-1)(n_{qh}-2)+1\}\left\{\frac{(1-\lambda)}{M_q-1}\right\}^2 \\ &= \left\{\lambda+(n_{qh}-2)\frac{(1-\lambda)}{M_q-1}\right\}^2+(n_{qh}-1)\left\{\frac{(1-\lambda)}{M_q-1}\right\}^2 \\ &= \lambda^2+O(M_q^{-1}).\end{aligned}$$

Case 2: $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ when $g \neq h$ but $g = u,\ h = v$

Since all elements of $\mathbf{A}_{gh}$ are off-diagonal in matrix $\mathbf{A}_q$, so an exchangeable first order linkage structure becomes

$$e_{ik} = \frac{(1-\lambda)}{M_q - 1} I(i \neq k).$$

Note that all elements of $\mathbf{A}_{uv}$ are also off-diagonal in matrix $\mathbf{A}_q$, therefore

$$\begin{aligned}
\Pr(a_{jl} = 1 \mid\ a_{ik} = 1) &= I(i=j)I(k=l) \\
&\quad + I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\} \\
&= I(i=j)I(k=l) + e_{jl|ik}.
\end{aligned}$$

Once again, let $E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top) = \mathbf{Z} = [z_{ij}]$. Then

$$z_{ii} = \sum_k e_{ik} = n_{qh}\frac{(1-\lambda)}{M_q - 1} = O(M_q^{-1})$$

and, for $i \neq j$,

$$\begin{aligned}
z_{ij} &= \sum_k \sum_l e_{ik} e_{jl|ik} \\
&= \sum_k \sum_l \left\{\frac{(1-\lambda)}{M_q - 1} I(i \neq k)\right\} \left[I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\}\right] \\
&= \sum_{k \neq i} \sum_{l \neq j,k} \left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\
&= n_{qh}(n_{qh} - 1)\left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\
&= O(M_q^{-1}).
\end{aligned}$$

Case 3: $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ when $g = h$ but $u \neq v$

Since elements of $\mathbf{A}_{gh}$ in this situation can be either on-diagonal or off-diagonal in matrix $\mathbf{A}_q$, so an exchangeable first order linkage structure remains the same as in Case 1 i.e.

$$e_{ik} = \lambda I(i = k) + \frac{(1-\lambda)}{M_q - 1} I(i \neq k).$$

However, since a_{jl} and a_{ik} cannot be the same element, and all elements of $\mathbf{A}_{uv}$ are off-diagonal in matrix $\mathbf{A}_q$ therefore

$$\begin{aligned} \Pr(a_{jl} = 1 \mid \ a_{ik} = 1) &= I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\} \\ &= e_{jl|ik}. \end{aligned}$$

If we put $E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top) = \mathbf{Z} = [z_{ij}]$, $\mathbf{Z}$ is not always a square matrix. Therefore, for all i, j

$$\begin{aligned} z_{ij} &= \sum_k \sum_l e_{ik} e_{jl|ik} \\ &= \sum_k \sum_l \left\{\lambda I(i = k) + \frac{(1-\lambda)}{M_q - 1} I(i \neq k)\right\} \left[I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\}\right] \\ &= \sum_{l \neq i,j} \frac{\lambda(1-\lambda)}{M_q - 1} + \sum_{k \neq i} \sum_{l \neq j,k} \left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\ &= (n_{qh} - 1)\left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \left\{\frac{\lambda(M_q - 1)}{1-\lambda} + (n_{qh} - 1)\right\} \\ &= O(M_q^{-1}). \end{aligned}$$

Case 4: $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ when $g \neq h$ but $u = v$

Similar to Case 2, elements of $\mathbf{A}_{gh}$ in this situation are off-diagonal in matrix $\mathbf{A}_q$, so an exchangeable first order linkage structure is given by

$$e_{ik} = \frac{(1-\lambda)}{M_q - 1} I(i \neq k).$$

Similar to Case 3, a_{jl} and a_{ik} cannot be the same element, therefore

$$\begin{aligned} \Pr(a_{jl} = 1 \mid \ a_{ik} = 1) &= I(i \neq j)I(k \neq l)\left\{\lambda I(j = l) + \frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\} \\ &= e_{jl|ik}. \end{aligned}$$

Once again, if we put $E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top) = \mathbf{Z} = [z_{ij}]$, then $\mathbf{Z}$ needs not to be a square matrix. Thus, for all i, j

$$\begin{aligned} z_{ij} &= \sum_k \sum_l e_{ik} e_{jl|ik} \\ &= \sum_k \sum_l \left\{\frac{(1-\lambda)}{M_q - 1} I(i \neq k)\right\} \left[I(i \neq j)I(k \neq l)\left\{\lambda I(j = l) + \frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\}\right] \\ &= \sum_{k \neq i,j} \frac{\lambda(1-\lambda)}{M_q - 1} + \sum_{k \neq i,l} \sum_{l \neq j} \left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\ &= (n_{qh} - 1)\left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \left\{\frac{\lambda(M_q - 1)}{1-\lambda} + (n_{qh} - 1)\right\} \\ &= O(M_q^{-1}). \end{aligned}$$

Case 5: $E_X(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top)$ when $g \neq h$, $u \neq v$ and ($g \neq u$ or $h \neq v$)

Clearly, all elements of $\mathbf{A}_{gh}$ and $\mathbf{A}_{uv}$ in this situation are off-diagonal in matrix $\mathbf{A}_q$, so an exchangeable first order linkage structure is given by

$$e_{ik} = \frac{(1-\lambda)}{M_q - 1} I(i \neq k)$$

and,

$$\begin{aligned} \Pr(a_{jl} = 1 \mid \ a_{ik} = 1) &= I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\} \\ &= e_{jl|ik}. \end{aligned}$$

Similarly, if $E(\mathbf{A}_{gh}\mathbf{1}_h\mathbf{1}_h^\top\mathbf{A}_{uv}^\top) = \mathbf{Z} = [z_{ij}]$, then for all i, j

$$\begin{aligned} z_{ij} &= \sum_k \sum_l e_{ik} e_{jl|ik} \\ &= \sum_k \sum_l \left\{\frac{(1-\lambda)}{M_q - 1} I(i \neq k)\right\} \left[I(i \neq j)I(k \neq l)\left\{\frac{(1-\lambda)}{M_q - 1} I(j \neq l)\right\}\right] \\ &= \sum_{k \neq i} \sum_{l \neq j,k} \left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\ &= n_{qh}(n_{qh} - 1)\left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \\ &= O(M_q^{-1}). \end{aligned}$$

That is, for large M_q we can approximate $\mathbf{K}_q$ by

$$\mathbf{K}_q \approx \{_d\ \lambda^2 \mathbf{J}_{n_{qg}} + \lambda(1-\lambda)\mathbf{I}_{n_{qg}}\}.$$

However, if M_q is not large and G_q is number of groups in block q, $\mathbf{K}_q = [k_{ij}]$ can be accurately defined as

$$[k_{ij}] = \begin{cases} \lambda + \frac{(1-\lambda)}{M_q - 1}(G_q n_{qh} - 1), & \text{if } i = j \\ \left\{\lambda + (n_{qh} - 2)\frac{(1-\lambda)}{M_q - 1}\right\}^2 + (n_{qh} - 1)\left\{\frac{(1-\lambda)}{M_q - 1}\right\}^2 \{1 + (G_q - 1)n_{qh}\}, & \text{if } i \neq j; \\ & i, j \in g;\ g = h \\ (n_{qh} - 1)\frac{(1-\lambda)}{M_q - 1}\left\{2\lambda + \frac{(1-\lambda)}{M_q - 1}(G_q n_{qh} - 2)\right\}, & \text{if } i \neq j;\ i \in g \\ & j \in h;\ g \neq h. \end{cases}$$

Appendix B

ANOVA Estimation

B.1 Variance Components Estimation for Perfect Linkage

We first develop an expression for $\mathrm{E}_X(\mathrm{SSA})$. The corresponding expression for $\mathrm{E}_X(\mathrm{SSE})$ follows similarly. In the case that linkage is perfect, we have

$$\begin{aligned} \mathrm{SSA} &= \mathbf{y}^\top \mathbf{B}\mathbf{y} \\ &= \sum_q \sum_r \mathbf{y}_q^\top \mathbf{B}_{qr} \mathbf{y}_r. \end{aligned} \tag{B.1}$$

Thus,

$$\begin{aligned} E_X(\mathrm{SSA}) &= E_X\left(\sum_q \sum_r \mathbf{y}_q^\top \mathbf{B}_{qr} \mathbf{y}_r\right) \\ &= \sum_q E_X\left(\mathbf{y}_q^\top \mathbf{B}_{qq} \mathbf{y}_q\right) + \sum_q \sum_{r \neq q} E_X\left(\mathbf{y}_q^\top \mathbf{B}_{qr} \mathbf{y}_r\right). \end{aligned} \tag{B.2}$$

Note that, by using a quadratic form theorem [60]

$$\begin{aligned}
\sum_q E_X\left(\mathbf{y}_q^\top \mathbf{B}_{qq}\mathbf{y}_q\right) &= \sum_q \left[\mathrm{tr}\left\{\mathbf{B}_{qq}Var(\mathbf{y}_q)\right\} + \mathbf{f}_q^\top \mathbf{B}_{qq}\mathbf{f}_q\right] \\
&= \sum_q \left[\mathrm{tr}\left\{\mathbf{B}_{qq}(\sigma_u^2\mathbf{Z}_q\mathbf{Z}_q^\top + \sigma_e^2\mathbf{I}_q)\right\} + \mathbf{f}_q^\top \mathbf{B}_{qq}\mathbf{f}_q\right] \\
&= \sigma_u^2 \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{Z}_q\mathbf{Z}_q^\top\right) + \sigma_e^2 \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\right) + \sum_q \mathbf{f}_q^\top \mathbf{B}_{qq}\mathbf{f}_q.
\end{aligned}$$

Then, we consider the second term of (B.2). As indicated in (3.3) and (3.4),

$$\begin{bmatrix} \mathbf{y}_q \\ \mathbf{y}_r \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{f}_q \\ \mathbf{f}_r \end{bmatrix}, \begin{bmatrix} \mathbf{V}_{qq} & \mathbf{V}_{qr} \\ \mathbf{V}_{rq} & \mathbf{V}_{rr} \end{bmatrix}\right).$$

Since $\mathbf{y}_q$ and $\mathbf{y}_r$ are not independent, the conditional expectation of $\mathbf{y}_q$ given $\mathbf{y}_r$ is then $E_X(\mathbf{y}_q \mid \mathbf{y}_r) = \mathbf{f}_q + \mathbf{V}_{qr}\mathbf{V}_{rr}^{-1}(\mathbf{y}_r - \mathbf{f}_r)$. Therefore,

$$\begin{aligned}
\sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_q^\top \mathbf{B}_{qr}\mathbf{y}_r\right) &= \sum_q \sum_{r\neq q} E_X\left\{E_{X_{y_r}}\left(\mathbf{y}_q^\top \mathbf{B}_{qr}\mathbf{y}_r\right)\right\} \\
&= \sum_q \sum_{r\neq q} E_X\left[\left\{\mathbf{f}_q + \mathbf{V}_{qr}\mathbf{V}_{rr}^{-1}(\mathbf{y}_r - \mathbf{f}_r)\right\}^\top \mathbf{B}_{qr}\mathbf{y}_r\right] \\
&= \sum_q \sum_{r\neq q} E_X\left[\left\{\mathbf{f}_q^\top + (\mathbf{y}_r^\top - \mathbf{f}_r^\top)\mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top\right\}\mathbf{B}_{qr}\mathbf{y}_r\right] \\
&= \sum_q \sum_{r\neq q} E_X\left(\mathbf{f}_q^\top \mathbf{B}_{qr}\mathbf{y}_r\right) + \sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_r^\top \mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\mathbf{y}_r\right) \\
&\quad - \sum_q \sum_{r\neq q} E_X\left(\mathbf{f}_r^\top \mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\mathbf{y}_r\right) \\
&= \sum_q \sum_{r\neq q} \mathbf{f}_q^\top \mathbf{B}_{qr}\mathbf{f}_r + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\mathbf{V}_{rr}\right) + \\
&\quad \sum_q \sum_{r\neq q} \mathbf{f}_r^\top \mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\mathbf{f}_r - \sum_q \sum_{r\neq q} \mathbf{f}_r^\top \mathbf{V}_{rr}^{-1}\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\mathbf{f}_r
\end{aligned}$$

$$= \sum_q \sum_{r \neq q} \mathbf{f}_q^\top \mathbf{B}_{qr} \mathbf{f}_r + \sum_q \sum_{r \neq q} \mathrm{tr}\left(\mathbf{V}_{qr}^\top \mathbf{B}_{qr}\right)$$

$$= \sum_q \sum_{r \neq q} \mathbf{f}_q^\top \mathbf{B}_{qr} \mathbf{f}_r + \sigma_u^2 \sum_q \sum_{r \neq q} \mathrm{tr}\left\{\mathbf{Z}_r \mathbf{Z}_q^\top \mathbf{B}_{qr}\right\}$$

As a result,

$$\begin{aligned} E_X(\mathrm{SSA}) &= \sigma_u^2 \left[\sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{Z}_q\mathbf{Z}_q^\top\right) + \sum_q \sum_{r \neq q} \mathrm{tr}\left(\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{B}_{qr}\right)\right] \\ &\quad + \sigma_e^2 \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\right) + \sum_q \sum_r \mathbf{f}_q^\top \mathbf{B}_{qr}\mathbf{f}_r \\ &= \sigma_u^2 a + \sigma_e^2 b + m_0 \end{aligned} \tag{B.3}$$

where

$$a = \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{Z}_q\mathbf{Z}_q^\top\right) + \sum_q \sum_{r \neq q} \mathrm{tr}\left(\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{B}_{qr}\right) = N - \frac{\sum_g N_g^2}{N}$$

$$b = \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\right) = G - 1$$

$$m_0 = \sum_q \sum_r \mathbf{f}_q^\top \mathbf{B}_{qr} \mathbf{f}_r.$$

Similarly,

$$\begin{aligned} E_X(\mathrm{SSE}) &= \sigma_u^2 \left[\sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{Z}_q\mathbf{Z}_q^\top\right) + \sum_q \sum_{r \neq q} \mathrm{tr}\left(\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{C}_{qr}\right)\right] \\ &\quad + \sigma_e^2 \sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\right) + \sum_q \sum_r \mathbf{f}_q^\top \mathbf{C}_{qr}\mathbf{f}_r \\ &= \sigma_u^2 c + \sigma_e^2 d + n_0 \end{aligned} \tag{B.4}$$

where

$$\begin{aligned} c &= \sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{Z}_q\mathbf{Z}_q^\top\right) + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{C}_{qr}\right) = 0 \\ d &= \sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\right) = N - G \\ n_0 &= \sum_q \sum_r \mathbf{f}_q^\top \mathbf{C}_{qr}\mathbf{f}_r. \end{aligned}$$

Having derived (B.3) and (B.4), we use the equate sums of squares to their expected values principle, which is called the ANOVA method of estimation [60]. The two equations, which are linear in the variance components, are now written using the estimators $\hat{\sigma}_u^2$ and $\hat{\sigma}_e^2$, so that $\mathrm{SSA} = \hat{\sigma}_u^2 a + \hat{\sigma}_e^2 b + m_0$ and $\mathrm{SSE} = \hat{\sigma}_u^2 c + \hat{\sigma}_e^2 d + n_0$. By solving two linear equations,

$$\hat{\sigma}_u^2 a + \hat{\sigma}_e^2 b = m \quad ; m = \mathrm{SSA} - m_0$$

and

$$\hat{\sigma}_e^2 d = n \quad ; n = \mathrm{SSE} - n_0$$

it yields the estimators

$$\hat{\sigma}_e^2 = \frac{n}{d} \tag{B.5}$$

and

$$\hat{\sigma}_u^2 = \frac{m - \hat{\sigma}_e^2 b}{a}. \tag{B.6}$$

However, the linkage is not usually perfect and therefore these estimates cannot be calculated. Instead, what is usually done is to substitute the linked data values $\mathbf{y}_q^*$ for $\mathbf{y}_q$ which leads to the naïve variance component estimates.

B.2 Variance Components Estimation for Imperfect Linkage

If the linkage is not perfect, then we have

$$\begin{aligned} \text{SSA} &= \mathbf{y}^{*\top}\mathbf{B}\mathbf{y}^{*} \\ &= \sum_{q}\sum_{r}\mathbf{y}_{q}^{*\top}\mathbf{B}_{qr}\mathbf{y}_{r}^{*} \end{aligned} \tag{B.7}$$

$$\begin{aligned} E_X(\text{SSA}) &= E_X\left(\sum_{q}\sum_{r}\mathbf{y}_{q}^{*\top}\mathbf{B}_{qr}\mathbf{y}_{r}^{*}\right) \\ &= E_X\left(\sum_{q}\mathbf{y}_{q}^{*\top}\mathbf{B}_{qq}\mathbf{y}_{q}^{*}\right) + E_X\left(\sum_{q}\sum_{r\neq q}\mathbf{y}_{q}^{*\top}\mathbf{B}_{qr}\mathbf{y}_{r}^{*}\right) \\ &= \sum_{q}E_X\left(\mathbf{y}_{q}^{*\top}\mathbf{B}_{qq}\mathbf{y}_{q}^{*}\right) + \sum_{q}\sum_{r\neq q}E_X\left(\mathbf{y}_{q}^{*\top}\mathbf{B}_{qr}\mathbf{y}_{r}^{*}\right). \end{aligned} \tag{B.8}$$

Consider the first term on the right hand side of (B.8). This can be written

$$\begin{aligned} \sum_{q}E_X\left(\mathbf{y}_{q}^{*\top}\mathbf{B}_{qq}\mathbf{y}_{q}^{*}\right) &= \sum_{q}\text{tr}\left\{\mathbf{B}_{qq}Var(\mathbf{y}_{q}^{*})\right\} + \sum_{q}\mathbf{f}_{q}^{\top}\mathbf{E}_{q}^{\top}\mathbf{B}_{qq}\mathbf{E}_{q}\mathbf{f}_{q} \\ &= \sum_{q}\text{tr}\left\{\mathbf{B}_{qq}(\sigma_u^2\mathbf{K}_q + \sigma_e^2\mathbf{I}_q + \mathbf{V}_q)\right\} + \sum_{q}\mathbf{f}_{q}^{\top}\mathbf{E}_{q}^{\top}\mathbf{B}_{qq}\mathbf{E}_{q}\mathbf{f}_{q} \\ &= \sigma_u^2\sum_{q}\text{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q\right) + \sigma_e^2\sum_{q}\text{tr}\left(\mathbf{B}_{qq}\right) \\ &\quad + \sum_{q}\text{tr}\left\{(\mathbf{B}_{qq}\mathbf{V}_q) + \mathbf{f}_{q}^{\top}\mathbf{E}_{q}^{\top}\mathbf{B}_{qq}\mathbf{E}_{q}\mathbf{f}_{q}\right\} \end{aligned}$$

Then, we consider the second term of (B.8). From (3.9), (3.12) and (3.15),

$$\begin{bmatrix} \mathbf{y}_{q}^{*} \\ \mathbf{y}_{r}^{*} \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{E}_q\mathbf{f}_q \\ \mathbf{E}_r\mathbf{f}_r \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{qq} & \mathbf{\Sigma}_{qr} \\ \mathbf{\Sigma}_{rq} & \mathbf{\Sigma}_{rr} \end{bmatrix}\right).$$

Again $\mathbf{y}_q^*$ and $\mathbf{y}_r^*$ are not independent, the conditional expectation of $\mathbf{y}_q^*$ given $\mathbf{y}_r^*$ is then $E_X(\mathbf{y}_q^* \mid \mathbf{y}_r^*) = \mathbf{E}_q\mathbf{f}_q + \boldsymbol{\Sigma}_{qr}\boldsymbol{\Sigma}_{rr}^{-1}(\mathbf{y}_r - \mathbf{E}_r\mathbf{f}_r)$. Therefore,

$$\begin{aligned}
\sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^*\right) &= \sum_q \sum_{r\neq q} E_X\left\{E_{Xy_r^*}\left(\mathbf{y}_q^{*\top}\mathbf{B}_{qr}\mathbf{y}_r^*\right)\right\} \\
&= \sum_q \sum_{r\neq q} E_X\left[\left\{\mathbf{E}_q\mathbf{f}_q + \boldsymbol{\Sigma}_{qr}\boldsymbol{\Sigma}_{rr}^{-1}(\mathbf{y}_r^* - \mathbf{E}_r\mathbf{f}_r)\right\}^\top \mathbf{B}_{qr}\mathbf{y}_r^*\right] \\
&= \sum_q \sum_{r\neq q} E_X\left[\left\{\mathbf{f}_q^\top\mathbf{E}_q^\top + (\mathbf{y}_r^{*\top} - \mathbf{f}_r^\top\mathbf{E}_r^\top)\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\right\}\mathbf{B}_{qr}\mathbf{y}_r^*\right] \\
&= \sum_q \sum_{r\neq q} E_X\left(\mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{y}_r^*\right) + \sum_q \sum_{r\neq q} E_X\left(\mathbf{y}_r^{*\top}\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\mathbf{B}_{qr}\mathbf{y}_r^*\right) \\
&\quad - \sum_q \sum_{r\neq q} E_X\left(\mathbf{f}_r^\top\mathbf{E}_r^\top\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\mathbf{B}_{qr}\mathbf{y}_r^*\right) \\
&= \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\mathbf{B}_{qr}\boldsymbol{\Sigma}_{rr}\right) \\
&\quad + \sum_q \sum_{r\neq q} \mathbf{f}_r^\top\mathbf{E}_r^\top\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r - \sum_q \sum_{r\neq q} \mathbf{f}_r^\top\mathbf{E}_r^\top\boldsymbol{\Sigma}_{rr}^{-1}\boldsymbol{\Sigma}_{qr}^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r \\
&= \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r + \sigma_u^2 \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\right).
\end{aligned}$$

It immediately follows that

$$\begin{aligned}
E_X(\mathrm{SSA}) &= \sigma_u^2\left\{\sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\right)\right\} + \sigma_e^2 \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\right) \\
&\quad + \sum_q \left\{\mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qq}\mathbf{E}_q\mathbf{f}_q\right\} + \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r \\
&= \sigma_u^2 a + \sigma_e^2 b + m_0 \qquad \text{(B.9)}
\end{aligned}$$

where

$$\begin{aligned} a &= \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\right) \\ b &= \sum_q \mathrm{tr}\left(\mathbf{B}_{qq}\right) \\ m_0 &= \sum_q \left\{\mathrm{tr}\left(\mathbf{B}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qq}\mathbf{E}_q\mathbf{f}_q\right\} + \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{B}_{qr}\mathbf{E}_r\mathbf{f}_r. \end{aligned}$$

Similarly,

$$\begin{aligned} E_X(\mathrm{SSE}) &= \sigma_u^2\left\{\sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qr}\right)\right\} + \sigma_e^2\sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\right) \\ &\quad + \sum_q \left\{\mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qq}\mathbf{E}_q\mathbf{f}_q\right\} + \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qr}\mathbf{E}_r\mathbf{f}_r \\ &= \sigma_u^2 c + \sigma_e^2 d + n_0 \end{aligned} \tag{B.10}$$

where

$$\begin{aligned} c &= \sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{K}_q\right) + \sum_q \sum_{r\neq q} \mathrm{tr}\left(\mathbf{E}_r\mathbf{Z}_r\mathbf{Z}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qr}\right) \\ d &= \sum_q \mathrm{tr}\left(\mathbf{C}_{qq}\right) \\ n_0 &= \sum_q \left\{\mathrm{tr}\left(\mathbf{C}_{qq}\mathbf{V}_q\right) + \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qq}\mathbf{E}_q\mathbf{f}_q\right\} + \sum_q \sum_{r\neq q} \mathbf{f}_q^\top\mathbf{E}_q^\top\mathbf{C}_{qr}\mathbf{E}_r\mathbf{f}_r. \end{aligned}$$

Once again, by solving two linear equations,

$$\hat{\sigma}_u^2 a + \hat{\sigma}_e^2 b = m \quad ; m = \mathrm{SSA} - m_0$$

and

$$\hat{\sigma}_u^2 c + \hat{\sigma}_e^2 d = n \quad ; n = \mathrm{SSE} - n_0$$

it yields the estimators

$$\hat{\sigma}_e^2 = \frac{mc - na}{bc - da} \tag{B.11}$$

and

$$\hat{\sigma}_u^2 = \frac{m - \hat{\sigma}_e^2 b}{a}. \tag{B.12}$$

Appendix C

Optimal Estimating Function

Suppose that $Y_1, ..., Y_T$ are independent random variables with $E(Y_t) = \alpha_t(\theta)$, $Var(Y_t) = \sigma_t^2(\theta)$, the α_t's and σ_t^2's being specified differentiable functions. We can restate the Gauss-Markov theorem in terms of estimating function $G \in G_0$ such that

$$G = \sum_{t=1}^{T} b_t(\theta)\{y_t - \alpha_t(\theta)\}.$$

Note that the estimating functions kG, k constant and G produce the same estimator, so some standardization is necessary if variances are to be compared [30]. One possible standardization is defined [24] as

$$G_s = \frac{G}{E\left(\frac{\partial G}{\partial \theta}\right)} = \frac{\sum_{t=1}^{T} b_t(\theta)\{y_t - \alpha_t(\theta)\}}{-\sum_{t=1}^{T} b_t(\theta)\frac{\partial \alpha_t}{\partial \theta}}$$

and

$$Var(G_s) = \frac{\sum_{t=1}^{T} b_t^2(\theta)\sigma_t^2(\theta)}{\left\{\sum_{t=1}^{T} b_t(\theta)\frac{\partial \alpha_t}{\partial \theta}\right\}^2}.$$

By Definition 2.1 of Godambe [24], an estimating function $G^* \in G_0$ is said to be optimum if, for all $G \in G_0$

$$Var(G_s^*) \leq Var(G_s). \tag{C.1}$$

It has been claimed [25] that $Var(G_s)$ is minimized for $b_t(\theta) = \dfrac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta),\ t = 1, 2, ..., T$ which leads to the optimum estimating function in G_0 given by

$$G^* = \sum_{t=1}^{T} \frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta)\{y_t - \alpha_t(\theta)\}. \tag{C.2}$$

The standardization of G^* is then defined as

$$G_s^* = \frac{\displaystyle\sum_{t=1}^{T} \frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta)\{y_t - \alpha_t(\theta)\}}{\displaystyle -\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)}$$

and

$$\begin{aligned} Var(G_s^*) &= \frac{\displaystyle\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \left\{\sigma_t^{-2}(\theta)\right\}^2 \sigma_t^2(\theta)}{\displaystyle\left\{\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)\right\}^2} \\ &= \frac{\displaystyle\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)}{\displaystyle\left\{\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)\right\}^2} \\ &= \frac{1}{\displaystyle\sum_{t=1}^{T} \left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)}. \end{aligned} \tag{C.3}$$

To prove if this is a minimum variance, let $\tilde{b}_t(\theta) = \dfrac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta) + a_t(\theta)$ be another weighting function of $b_t(\theta)$ where $a_t(\theta) \neq 0$ and leads to an unbiased estimating

function i.e. $E(\tilde{G}) = 0$.

Thus,

$$
\begin{aligned}
Var(\tilde{G}_s) &= \frac{\sum_{t=1}^{T}\left\{\frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta) + a_t(\theta)\right\}^2 \sigma_t^2(\theta)}{\left[\sum_{t=1}^{T}\left\{\frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta) + a_t(\theta)\right\}\frac{\partial \alpha_t}{\partial \theta}\right]^2} \\
&= \frac{\sum_{t=1}^{T}\left[\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \{\sigma_t^{-2}(\theta)\}^2 + 2\frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta)a_t(\theta) + a_t^2(\theta)\right]\sigma_t^2(\theta)}{\left[\sum_{t=1}^{T}\left\{\frac{\partial \alpha_t}{\partial \theta}\sigma_t^{-2}(\theta) + a_t(\theta)\right\}\frac{\partial \alpha_t}{\partial \theta}\right]^2} \\
&= \frac{\sum_{t=1}^{T}\left\{\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta) + 2\frac{\partial \alpha_t}{\partial \theta}a_t(\theta) + a_t^2(\theta)\sigma_t^2(\theta)\right\}}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta) + \frac{\partial \alpha_t}{\partial \theta}a_t(\theta)\right\}\right]^2} \\
&= \frac{1}{\sum_{t=1}^{T}\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)} + \frac{\sum_{t=1}^{T}\left\{\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta) + 2\frac{\partial \alpha_t}{\partial \theta}a_t(\theta) + a_t^2(\theta)\sigma_t^2(\theta)\right\}}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta) + \frac{\partial \alpha_t}{\partial \theta}a_t(\theta)\right\}\right]^2} \\
&\quad - \frac{1}{\sum_{t=1}^{T}\left(\frac{\partial \alpha_t}{\partial \theta}\right)^2 \sigma_t^{-2}(\theta)}
\end{aligned}
$$

$$
\begin{aligned}
= & \frac{1}{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)} \\
& + \frac{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+2\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)+a_t^2(\theta)\sigma_t^2(\theta)\right\}\right]\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}} \\
& - \frac{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}}
\end{aligned}
$$

$$
\begin{aligned}
= & \frac{1}{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)} \\
& + \left[\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}^2+2\left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}\right. \\
& + \left\{\sum_{t=1}^{T}a_t^2(\theta)\sigma_t^2(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}-\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}^2 \\
& \left.-2\left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}-\left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}^2\right] \\
& \Big/\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}
\end{aligned}
$$

$$
= \frac{1}{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)} + \frac{\left\{\sum_{t=1}^{T}a_t^2(\theta)\sigma_t^2(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}-\left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}^2}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)+\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}}
$$

By the Cauchy-Schwarz inequality,

$$\left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}^2 \leq \left\{\sum_{t=1}^{T}a_t^2(\theta)\sigma_t^2(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}.$$

Thus,

$$Var(\tilde{G}_s) = Var(G_s^*) + A$$

where

$$A = \frac{\left\{\sum_{t=1}^{T}a_t^2(\theta)\sigma_t^2(\theta)\right\}\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\} - \left\{\sum_{t=1}^{T}\frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}^2}{\left[\sum_{t=1}^{T}\left\{\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta) + \frac{\partial\alpha_t}{\partial\theta}a_t(\theta)\right\}\right]^2\left\{\sum_{t=1}^{T}\left(\frac{\partial\alpha_t}{\partial\theta}\right)^2\sigma_t^{-2}(\theta)\right\}} \quad , A \geq 0.$$

Since A is a positive constant such that (C.1) holds, therefore (C.2) is an optimal estimating function.

Appendix D

Simulation Study Code

The simulation analysis in Chapter 6 has been carried out using the software R. Four R codes were run corresponding to two situations under two scenarios. The first set of R codes is for cross-sectional linked data. Another set of R codes is for longitudinal linked data. The first scenario is where the linkage probabilities are known. The second scenario is where they are estimated by taking random samples and checking to see whether these sampled links were correct.

D.1 Simulation of Linear Mixed Model with Cross-Sectional Linked Data

D.1.1 Simulation of Known Lambda Case

```
The following code assumes
1. 1-1 linking of registers
2. No measurement error on the blocking variable
3. Exchangeable linkage errors within blocks

library(nlme)
library(psych)
LinEstPlus <- function(lambda,var.u,var.e,X,Ystar,Block,Group,alpha=0.05,
            nits=5,m){

# Setting up
    G <- length(unique(Group))
    Q <- length(lambda)
    R <- length(lambda)
    N <- nrow(X)
    p <- ncol(X)
```

```
    U1 <- matrix(0,nrow=p,ncol=p)
    U2 <- matrix(0,nrow=p,ncol=1)
    U3 <- matrix(0,nrow=p,ncol=p)
    U4 <- matrix(0,nrow=p,ncol=p)
    U5 <- matrix(0,nrow=p,ncol=1)
    U6 <- matrix(0,nrow=p,ncol=1)

# Loop to calculate primitives for initial b_R, b_A, b_C and b_B
    W <- VInv(Group=Group,v.u=var.u,v.e=var.e)
    idnum <- c(1:N)
    for (q in 1:Q){
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Xq <- X[Block==q,]
        Hq <- Eq%*%Xq
        for (r in 1:R){
            Mr <- sum(Block==r)
            lambdar <- lambda[r]
            Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
            diag(Er) <- lambdar
            Xr <- X[Block==r,]
            Ystarr <- Ystar[Block==r]
            Hr <- Er%*%Xr
            Wqr <- W[idnum[Block==q],idnum[Block==r]]
            U1 <- U1+(t(Xq)%*%Wqr%*%Hr)
            U2 <- U2+(t(Xq)%*%Wqr%*%matrix(Ystarr,ncol=1))
            U3 <- U3+(t(Xq)%*%Wqr%*%Xr)
            U4 <- U4+(t(Hq)%*%Wqr%*%Hr)
            U5 <- U5+(t(Hq)%*%Wqr%*%matrix(Ystarr,ncol=1))
            U6 <- U6+(t(Xq)%*%Wqr%*%solve(t(Er)%*%Er)%*%t(Er)
                    %*%matrix(Ystarr,ncol=1))
        }
    }

# Calculation of initial b_R, b_A, b_C and b_B estimates
    U.1Inv <- solve(U1)
    U.3Inv <- solve(U3)
    b_R.int <- U.1Inv%*%U2
    U.4Inv <- solve(U4)
    b_A.int <- U.4Inv%*%U5
    b_B.int <- U.3Inv%*%U6

# Iterative calculation of b_R, b_A, b_C and b_B
    b_R <- b_R.int
    b_A <- b_A.int
    b_B <- b_B.int
```

```
    b_C <- b_A

# Initial estimates of sigmasq for b_R, b_A, b_C and b_B
    rb_R <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=Group,
            beta=b_R)
    rb_A <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=Group,
            beta=b_A)
    rb_B <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=Group,
            beta=b_B)
    rb_C <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=Group,
            beta=b_C)

# Start iterative calculation loop for b_R, b_A, b_C and b_B
    for (iter in 1:5) {
        U9 <- matrix(0,nrow=p,ncol=p)
        U10 <- matrix(0,nrow=p,ncol=1)
        U11 <- matrix(0,nrow=p,ncol=p)
        U12 <- matrix(0,nrow=p,ncol=1)
        U13 <- matrix(0,nrow=p,ncol=p)
        U14 <- matrix(0,nrow=p,ncol=1)
        U15 <- matrix(0,nrow=p,ncol=p)
        U16 <- matrix(0,nrow=p,ncol=1)

# Loop to compute primitives for b_R, b_A, b_C and b_B
        W.R <- VInv(Group=Group,v.u=rb_R$sigmasq.u,v.e=rb_R$sigmasq.e)
        W.A <- VInv(Group=Group,v.u=rb_A$sigmasq.u,v.e=rb_A$sigmasq.e)
        W.B <- VInv(Group=Group,v.u=rb_B$sigmasq.u,v.e=rb_B$sigmasq.e)
        P <- VstarInv(lambda=lambda,Group=Group,Block=Block,beta=b_C,
                v.u=rb_C$sigmasq.u,v.e=rb_C$sigmasq.e)
        for (q in 1:Q){
            Mq <- sum(Block==q)
            lambdaq <- lambda[q]
            Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
            diag(Eq) <- lambdaq
            Xq <- X[Block==q,]
            Hq <- Eq%*%Xq
            for (r in 1:R){
                Mr <- sum(Block==r)
                lambdar <- lambda[r]
                Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
                diag(Er) <- lambdar
                Xr <- X[Block==r,]
                Ystarr <- Ystar[Block==r]
                Hr <- Er%*%Xr
                Wqr.R <- W.R[idnum[Block==q],idnum[Block==r]]
                Wqr.A <- W.A[idnum[Block==q],idnum[Block==r]]
                Wqr.B <- W.B[idnum[Block==q],idnum[Block==r]]
                Pqr <- P[idnum[Block==q],idnum[Block==r]]
```

```
                U9 <- U9+(t(Xq)%*%Wqr.R%*%Hr)
                U10 <- U10+(t(Xq)%*%Wqr.R%*%matrix(Ystarr,ncol=1))
                U11 <- U11+(t(Hq)%*%Wqr.A%*%Hr)
                U12 <- U12+(t(Hq)%*%Wqr.A%*%matrix(Ystarr,ncol=1))
                U13 <- U13+(t(Xq)%*%Wqr.B%*%Xr)
                U14 <- U14+(t(Xq)%*%Wqr.B%*%solve(t(Er)%*%Er)%*%t(Er)
                        %*%matrix(Ystarr,ncol=1))
                U15 <- U15+(t(Hq)%*%Pqr%*%Hr)
                U16 <- U16+(t(Hq)%*%Pqr%*%matrix(Ystarr,ncol=1))
            }
        }

# Updated estimates of b_R, b_A, b_C and b_B
        U.9Inv <- solve(U9)
        U.11Inv <- solve(U11)
        U.13Inv <- solve(U13)
        U.15Inv <- solve(U15)
        b_R <- U.9Inv%*%U10
        b_A <- U.11Inv%*%U12
        b_B <- U.13Inv%*%U14
        b_C <- U.15Inv%*%U16

# Updated estimate of sigmasq
        rb_R <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,
                Group=Group,beta=b_R)
        rb_A <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,
                Group=Group,beta=b_A)
        rb_B <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,
                Group=Group,beta=b_B)
        rb_C <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,
                Group=Group,beta=b_C)

# End iterative calculation loop
    }

# Confidence limits of variance components
    sigmasqR <- rbind(rb_R$sigmasq.u,rb_R$sigmasq.e)
    sigmasqA <- rbind(rb_A$sigmasq.u,rb_A$sigmasq.e)
    sigmasqC <- rbind(rb_C$sigmasq.u,rb_C$sigmasq.e)
    sigmasqB <- rbind(rb_B$sigmasq.u,rb_B$sigmasq.e)
    var.sigmasqR <- rbind(rb_R$var.sigmasq.u,rb_R$var.sigmasq.e)
    var.sigmasqA <- rbind(rb_A$var.sigmasq.u,rb_A$var.sigmasq.e)
    var.sigmasqC <- rbind(rb_C$var.sigmasq.u,rb_C$var.sigmasq.e)
    var.sigmasqB <- rbind(rb_B$var.sigmasq.u,rb_B$var.sigmasq.e)
    SEsigmasqR <- sqrt(var.sigmasqR)
    SEsigmasqA <- sqrt(var.sigmasqA)
    SEsigmasqC <- sqrt(var.sigmasqC)
    SEsigmasqB <- sqrt(var.sigmasqB)
```

```
    z <- qnorm(1-alpha/2)
    CLVar_R <- cbind(sigmasqR-z*SEsigmasqR,sigmasqR+z*SEsigmasqR)
    CLVar_A <- cbind(sigmasqA-z*SEsigmasqA,sigmasqA+z*SEsigmasqA)
    CLVar_C <- cbind(sigmasqC-z*SEsigmasqC,sigmasqC+z*SEsigmasqC)
    CLVar_B <- cbind(sigmasqB-z*SEsigmasqB,sigmasqB+z*SEsigmasqB)

# ML and REML estimators
# E matrix
    Eq.l <- list()
    for (q in 1:Q){
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Eq.l[[q]] <- Eq
    }
    E <- bdiag(Eq.l[1:Q])

# Setting up initial values
    b_mle <- b_Naive
    b_REML <- b_Naive
    sigmau <- var.u
    sigmae <- var.e
    sigmau2 <- var.u
    sigmae2 <- var.e
    theta <- matrix(c(b_mle,sigmau,sigmae),ncol=1)
    theta2 <- matrix(c(sigmau2,sigmae2),ncol=1)
    DiffSigma <- DeVBigsigma(Block=Block,Group=Group,lambda=lambda)
    H <- E%*%X

# Iterative calculation of b_mle, b_REML and variance components
    for (i in 1:5){
        SigmaInv <- VstarInv(lambda=lambda,Group=Group,Block=Block,
                 beta=b_mle,v.u=sigmau,v.e=sigmae)
        SigmaInv2 <- VstarInv(lambda=lambda,Group=Group,Block=Block,
                 beta=b_REML,v.u=sigmau2,v.e=sigmae2)
        M <- SigmaInv2-SigmaInv2%*%H%*%solve(t(H)%*%SigmaInv2%*%H)%*%t(H)
            %*%SigmaInv2
        I11 <- t(H)%*%SigmaInv%*%H
        I12 <- matrix(0,nrow=2,ncol=1)
        I13 <- I12
        I21 <- t(I12)
        I22 <- 0.5*tr(SigmaInv%*%DiffSigma%*%SigmaInv%*%DiffSigma)
        I23 <- 0.5*tr(SigmaInv%*%SigmaInv%*%DiffSigma)
        I31 <- t(I13)
        I32 <- I23
        I33 <- 0.5*tr(SigmaInv%*%SigmaInv)
        I1 <- cbind(I11,I12,I13)
```

```
        I2 <- cbind(I21,I22,I23)
        I3 <- cbind(I31,I32,I33)
        IR11 <- 0.5*tr(M%*%DiffSigma%*%M%*%DiffSigma)
        IR12 <- 0.5*tr(M%*%M%*%DiffSigma)
        IR21 <- IR12
        IR22 <- 0.5*tr(M%*%M)
        IR1 <- cbind(IR11,IR12)
        IR2 <- cbind(IR21,IR22)
        Information <- rbind(I1,I2,I3)
        Information2 <- rbind(IR1,IR2)
        f <- X%*%matrix(b_mle,ncol=1)
        Diffl_Beta <- t(H)%*%SigmaInv%*%(Ystar-E%*%f)
        Diffl_sigmau <- -0.5*tr(SigmaInv%*%DiffSigma)+0.5*t(Ystar-E%*%f)%*%
                    SigmaInv%*%DiffSigma%*%SigmaInv%*%(Ystar-E%*%f)
        Diffl_sigmae <- -0.5*tr(SigmaInv)+0.5*t(Ystar-E%*%f)%*%SigmaInv%*%
                    SigmaInv%*%(Ystar-E%*%f)
        Difftheta <- rbind(Diffl_Beta,Diffl_sigmau,Diffl_sigmae)
        DifflR_sigmau <- -0.5*tr(M%*%DiffSigma)+0.5*t(Ystar)%*%M
                        %*%DiffSigma%*%M%*%Ystar
        DifflR_sigmae <- -0.5*tr(M)+0.5*t(Ystar)%*%M%*%M%*%Ystar
        Difftheta2 <- rbind(DifflR_sigmau,DifflR_sigmae)
        theta <- theta+solve(Information)%*%Difftheta
        theta2 <- theta2+solve(Information2)%*%Difftheta2
        b_mle <- theta[1:2,]
        sigmau <- theta[3,]
        sigmae <- theta[4,]
        b_REML <- solve(t(H)%*%SigmaInv2%*%H)%*%t(H)%*%SigmaInv2%*%Ystar
        sigmau2 <- theta2[1,]
        sigmae2 <- theta2[2,]
    }

# Estimated variances, SEs and confidence limits for beta estimators
    if(missing(m)) temp <- LinVarPlus(b_R=b_R,b_A=b_A,b_C=b_C,b_B=b_B,
                        b_mle=b_mle,b_REML=b_REML,rb_R=rb_R,rb_A=rb_A,
                        rb_C=rb_C,rb_B=rb_B,sigmau=sigmau,sigmae=sigmae,
                        sigmau2=sigmau2,sigmae2=sigmae2,lambda=lambda,
                        X=X,Ystar=Ystar,Block=Block,Group=Group,
                        alpha=alpha,W.R=W.R,W.A=W.A,W.B=W.B)

    else temp <- LinVarPlus(b_R=b_R,b_A=b_A,b_C=b_C,b_B=b_B,b_mle=b_mle,
                b_REML=b_REML,rb_R=rb_R,rb_A=rb_A,rb_C=rb_C,rb_B=rb_B,
                sigmau=sigmau,sigmae=sigmae,sigmau2=sigmau2,sigmae2=
                sigmae2,lambda=lambda,m=m,X=X,Ystar=Ystar,Block=Block,
                Group=Group,alpha=alpha,W.R=W.R,W.A=W.A,W.B=W.B)

# Confidence limits of ML and REML estimators
    SigmaInv <- VstarInv(lambda=lambda,Group=Group,Block=Block,
                beta=b_mle,v.u=sigmau,v.e=sigmae)
```

```
SigmaInv2 <- VstarInv(lambda=lambda,Group=Group,Block=Block,
            beta=c(b_REML),v.u=sigmau2,v.e=sigmae2)
M <- SigmaInv2-SigmaInv2%*%H%*%solve(t(H)%*%SigmaInv2%*%H)%*%t(H)%*%
    SigmaInv2
f <- X%*%matrix(b_mle,ncol=1)
Info22 <- 2*sigmau*tr(SigmaInv%*%DiffSigma)-2*(sigmau^2)*tr(SigmaInv
        %*%DiffSigma%*%SigmaInv%*%DiffSigma)-2*sigmau*t(Ystar-E%*%f)
        %*%SigmaInv%*%DiffSigma%*%SigmaInv%*%(Ystar-E%*%f)+4*(sigmau^2)
        *t(Ystar-E%*%f)%*%SigmaInv%*%DiffSigma%*%SigmaInv%*%DiffSigma
        %*%SigmaInv%*%(Ystar-E%*%f)
Info23 <- -2*sigmau*sigmae*tr(SigmaInv%*%SigmaInv%*%DiffSigma)+4*sigmau
        *sigmae*t(Ystar-E%*%f)%*%SigmaInv%*%SigmaInv%*%DiffSigma%*%
        SigmaInv%*%(Ystar-E%*%f)
Info33 <- 2*sigmae*tr(SigmaInv)-2*(sigmae^2)*tr(SigmaInv%*%SigmaInv)
        -2*sigmae*t(Ystar-E%*%f)%*%SigmaInv%*%SigmaInv%*%(Ystar-E%*%f)
        +4*(sigmae^2)*t(Ystar-E%*%f)%*%SigmaInv%*%SigmaInv%*%SigmaInv
        %*%(Ystar-E%*%f)

InfoR11 <- 2*sigmau2*tr(M%*%DiffSigma)-2*(sigmau2^2)*
        tr(M%*%DiffSigma%*%M%*%DiffSigma)-2*sigmau2*matrix(Ystar,nrow=1)
        %*%M%*%DiffSigma%*%M%*%matrix(Ystar,ncol=1)+4*(sigmau2^2)
        *matrix(Ystar,nrow=1)%*%M%*%DiffSigma%*%M%*%DiffSigma%*%M%*%
        matrix(Ystar,ncol=1)
InfoR12 <- -2*sigmau2*sigmae2*tr(M%*%M%*%DiffSigma)+4*sigmau2*sigmae2*
            matrix(Ystar,nrow=1)%*%M%*%M%*%DiffSigma%*%M%*%
            matrix(Ystar,ncol=1)
InfoR21 <- InfoR12
InfoR22 <- 2*sigmae2*tr(M)-2*(sigmae2^2)*tr(M%*%M)-2*sigmae2*
        matrix(Ystar,nrow=1)%*%M%*%M%*%matrix(Ystar,ncol=1)
        +4*(sigmae2^2)*matrix(Ystar,nrow=1)%*%M%*%M%*%M%*%
        matrix(Ystar,ncol=1)
Info <- matrix(c(Info22,Info23,Info23,Info33),nrow=2)
InfoR <- matrix(c(InfoR11,InfoR21,InfoR12,InfoR22),nrow=2)

sigmasqML <- rbind(sigmau,sigmae)
var.sigmasqML <- solve(Info)
SEsigmasqML <- sqrt(diag(var.sigmasqML))
CLVaru_ML <- c(sigmasqML[1]*(exp(-z*SEsigmasqML[1]))^2,
            sigmasqML[1]*(exp(z*SEsigmasqML[1]))^2)
CLVare_ML <- c(sigmasqML[2]*(exp(-z*SEsigmasqML[2]))^2,
            sigmasqML[2]*(exp(z*SEsigmasqML[2]))^2)
CLVar_ML <- rbind(CLVaru_ML,CLVare_ML)

sigmasqREML <- rbind(sigmau2,sigmae2)
var.sigmasqREML <- solve(InfoR)
SEsigmasqREML <- sqrt(diag(var.sigmasqREML))
CLVaru_REML <- c(sigmasqREML[1]*(exp(-z*SEsigmasqREML[1]))^2,
                sigmasqREML[1]*(exp(z*SEsigmasqREML[1]))^2)
```

```
    CLVare_REML <- c(sigmasqREML[2]*(exp(-z*SEsigmasqREML[2]))^2,
                     sigmasqREML[2]*(exp(z*SEsigmasqREML[2]))^2)
    CLVar_REML <- rbind(CLVaru_REML,CLVare_REML)

# Output
    list(b_R=b_R,SEb_R=temp$SEb_R,CLb_R=temp$CLb_R,
        rb_R=rb_R,SEsigmasqR=SEsigmasqR,CLVar_R=CLVar_R,
        b_A=b_A,SEb_A=temp$SEb_A,CLb_A=temp$CLb_A,
        rb_A=rb_A,SEsigmasqA=SEsigmasqA,CLVar_A=CLVar_A,
        b_C=b_C,SEb_C=temp$SEb_C,CLb_C=temp$CLb_C,
        rb_C=rb_C,SEsigmasqC=SEsigmasqC,CLVar_C=CLVar_C,
        b_B=b_B,SEb_B=temp$SEb_B,CLb_B=temp$CLb_B,
        rb_B=rb_B,SEsigmasqB=SEsigmasqB,CLVar_B=CLVar_B,
        b_mle=b_mle,SEb_mle=temp$SEb_mle,CLb_mle=temp$CLb_mle,
        sigmau=sigmau,sigmae=sigmae,SEsigmasqML=SEsigmasqML,
        CLVar_ML=CLVar_ML,
        b_REML=b_REML,SEb_REML=temp$SEb_REML,CLb_REML=temp$CLb_REML,
        sigmau2=sigmau2,sigmae2=sigmae2,SEsigmasqREML=SEsigmasqREML,
        CLVar_REML=CLVar_REML)

# End of function LinEstPlus
}

LinVarPlus <- function(b_R,b_A,b_C,b_B,b_mle,b_REML,rb_R,rb_A,rb_C,
            rb_B,sigmau,sigmae,sigmau2,sigmae2,lambda,m,X,Ystar,
            Block,Group,alpha,W.R,W.A,W.B)
{

# Estimated variance-covariance matrices, standard errors and confidence
limits for beta linear fits under exchangeable linkage errors and
estimated lambda

# Embedded function to compute contribution to variance kernel due to
estimation of lambda
    Delta <- function(beta,X,lambda,M,m) {
        f <- c(X%*%matrix(beta,ncol=1))
        r <- f-mean(f)
        ((M/(M-1))^2*lambda*(1-lambda)/m)*outer(r,r)
    }

# E matrix
    Eq.l <- list()
    for (q in 1:Q){
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Eq.l[[q]] <- Eq
```

```
    }
    E <- bdiag(Eq.l)

# Setting up
    Q <- length(lambda)
    R <- length(lambda)
    N <- nrow(X)
    p <- ncol(X)
    DelR <- matrix(0,nrow=p,ncol=p)
    DelA <- matrix(0,nrow=p,ncol=p)
    DelC <- matrix(0,nrow=p,ncol=p)
    DelB <- matrix(0,nrow=p,ncol=p)
    Delmle <- matrix(0,nrow=p,ncol=p)
    DelREML <- matrix(0,nrow=p,ncol=p)
    Vstarb_R <- solve(VstarInv(lambda=lambda,Group=Group,Block=Block,
                beta=b_R,v.u=rb_R$sigmasq.u,v.e=rb_R$sigmasq.e))
    Vstarb_A <- solve(VstarInv(lambda=lambda,Group=Group,Block=Block,
                beta=b_A,v.u=rb_A$sigmasq.u,v.e=rb_A$sigmasq.e))
    Vstarb_C <- VstarInv(lambda=lambda,Group=Group,Block=Block,beta=b_C,
                v.u=rb_C$sigmasq.u,v.e=rb_C$sigmasq.e)
    Vstarb_B <- solve(VstarInv(lambda=lambda,Group=Group,Block=Block,
                beta=b_B,v.u=rb_B$sigmasq.u,v.e=rb_B$sigmasq.e))
    Vstarb_mle <- VstarInv(lambda=lambda,Group=Group,Block=Block,
                    beta=b_mle,v.u=sigmau,v.e=sigmae)
    Vstarb_REML <- VstarInv(lambda=lambda,Group=Group,Block=Block,
                    beta=b_REML,v.u=sigmau2,v.e=sigmae2)

# Loop over blocks for variance components
    for (q in 1:Q){
        for (r in 1:R){
            Mq <- sum(Block==q)
            Mr <- sum(Block==r)
            if(!missing(m)) mr <- m[r]
            lambdaq <- lambda[q]
            lambdar <- lambda[r]
            Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
            diag(Eq) <- lambdaq
            Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
            diag(Er) <- lambdar
            Xq <- X[Block==q,]
            Xr <- X[Block==r,]
            Ystarr <- Ystar[Block==r]
            idnum <- c(1:N)
            Wqr.R <- W.R[idnum[Block==q],idnum[Block==r]]
            Wqr.A <- W.A[idnum[Block==q],idnum[Block==r]]
            Wqr.B <- W.B[idnum[Block==q],idnum[Block==r]]
            Sigmaqr <- Vstarb_C[idnum[Block==q],idnum[Block==r]]
            Sigmaqrmle <- Vstarb_mle[idnum[Block==q],idnum[Block==r]]
```

```
            SigmaqrREML <- Vstarb_REML[idnum[Block==q],idnum[Block==r]]
            if(!missing(m) & lambdar<1) Deltarb_R <- Delta(beta=b_R,X=Xr,
                lambda=lambdar,M=Mr,m=mr)
            else Deltarb_R <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            if(!missing(m) & lambdar<1) Deltarb_A <- Delta(beta=b_A,X=Xr,
                lambda=lambdar,M=Mr,m=mr)
            else Deltarb_A <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            if(!missing(m) & lambdar<1) Deltarb_C <- Delta(beta=b_C,X=Xr,
                lambda=lambdar,M=Mr,m=mr)
            else Deltarb_C <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            if(!missing(m) & lambdar<1) Deltarb_B <- Delta(beta=b_B,X=Xr,
                lambda=lambdar,M=Mr,m=mr)
            else Deltarb_B <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            if(!missing(m) & lambdar<1) Deltarb_mle <- Delta(beta=b_mle,
                X=Xr,lambda=lambdar,M=Mr,m=mr)
            else Deltarb_mle <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            if(!missing(m) & lambdar<1) Deltarb_REML <- Delta(beta=b_REML,
                X=Xr,lambda=lambdar,M=Mr,m=mr)
            else Deltarb_REML <- matrix(0,ncol=dim(Xr)[1],nrow=dim(Xr)[1])
            DelR <- DelR+t(Xq)%*%Wqr.R%*%Deltarb_R%*%t(t(Xq)%*%Wqr.R)
            DelA <- DelA+t(Xq)%*%t(Eq)%*%Wqr.A%*%Deltarb_A%*%t(t(Xq)
                    %*%t(Eq)%*%Wqr.A)
            DelC <- DelC+t(Xq)%*%t(Eq)%*%Sigmaqr%*%Deltarb_C%*%t(t(Xq)
                    %*%t(Eq)%*%Sigmaqr)
            DelB <- DelB+t(Xq)%*%Wqr.B%*%solve(t(Er)%*%Er)%*%t(Er)
                    %*%Deltarb_B%*%t(t(Xq)%*%Wqr.B%*%solve(t(Er)%*%Er)
                    %*%t(Er))
            Delmle <- Delmle+t(Xq)%*%t(Eq)%*%Sigmaqrmle%*%Deltarb_mle%*%
                        t(t(Xq)%*%t(Eq)%*%Sigmaqrmle)
            DelREML <- DelREML+t(Xq)%*%t(Eq)%*%SigmaqrREML%*%Deltarb_REML
                        %*%t(t(Xq)%*%t(Eq)%*%SigmaqrREML)
        }
    }

# Variance-covariance matrices
    VHb_R <- t(X)%*%W.R%*%Vstarb_R%*%t(W.R)%*%X
    VHb_A <- t(X)%*%t(E)%*%W.A%*%Vstarb_A%*%t(W.A)%*%E%*%X
    VHb_C <- t(X)%*%t(E)%*%t(Vstarb_C)%*%E%*%X
    VHb_B <- t(X)%*%W.B%*%solve(t(E)%*%E)%*%t(E)%*%Vstarb_B%*%E%*%
             solve(t(E)%*%E)%*%t(W.B)%*%X
    VHb_mle <- t(X)%*%t(E)%*%t(Vstarb_mle)%*%E%*%X
    VHb_REML <- t(X)%*%t(E)%*%t(Vstarb_REML)%*%E%*%X
    Vb_R <- solve(t(X)%*%W.R%*%E%*%X)%*%(VHb_R+DelR)%*%
            t(solve(t(X)%*%W.R%*%E%*%X))
    Vb_A <- solve(t(X)%*%t(E)%*%W.A%*%E%*%X)%*%(VHb_A+DelA)%*%
            t(solve(t(X)%*%t(E)%*%W.A%*%E%*%X))
    Vb_C <- solve(t(X)%*%t(E)%*%Vstarb_C%*%E%*%X)%*%(VHb_C+DelC)%*%
            t(solve(t(X)%*%t(E)%*%Vstarb_C%*%E%*%X))
```

```
    Vb_B <- solve(t(X)%*%W.B%*%X)%*%(VHb_B+DelB)%*%
            t(solve(t(X)%*%W.B%*%X))
    Vb_mle <- solve(t(X)%*%t(E)%*%Vstarb_mle%*%E%*%X)%*%(VHb_mle+Delmle)
            %*%t(solve(t(X)%*%t(E)%*%Vstarb_mle%*%E%*%X))
    Vb_REML <- solve(t(X)%*%t(E)%*%Vstarb_REML%*%E%*%X)%*%(VHb_REML+
                DelREML)%*%t(solve(t(X)%*%t(E)%*%Vstarb_REML%*%E%*%X))

# Standard errors
    SEb_R <- sqrt(diag(Vb_R))
    SEb_A <- sqrt(diag(Vb_A))
    SEb_C <- sqrt(diag(Vb_C))
    SEb_B <- sqrt(diag(Vb_B))
    SEb_mle <- sqrt(diag(Vb_mle))
    SEb_REML <- sqrt(diag(Vb_REML))

# Confidence limits
    z <- qnorm(1-alpha/2)
    CLb_R <- cbind(b_R-z*SEb_R,b_R+z*SEb_R)
    CLb_A <- cbind(b_A-z*SEb_A,b_A+z*SEb_A)
    CLb_C <- cbind(b_C-z*SEb_C,b_C+z*SEb_C)
    CLb_B <- cbind(b_B-z*SEb_B,b_B+z*SEb_B)
    CLb_mle <- cbind(b_mle-z*SEb_mle,b_mle+z*SEb_mle)
    CLb_REML <- cbind(b_REML-z*SEb_REML,b_REML+z*SEb_REML)

# Output
    list(Vb_R=Vb_R,SEb_R=SEb_R,CLb_R=CLb_R,
        Vb_A=Vb_A,SEb_A=SEb_A,CLb_A=CLb_A,
        Vb_C=Vb_C,SEb_C=SEb_C,CLb_C=CLb_C,
        Vb_B=Vb_B,SEb_B=SEb_B,CLb_B=CLb_B,
        Vb_mle=Vb_mle,SEb_mle=SEb_mle,CLb_mle=CLb_mle,
        Vb_REML=Vb_REML,SEb_REML=SEb_REML,CLb_REML=CLb_REML)
}

# Block-diagonal matrix function
bdiag <- function(x){
    if(!is.list(x)) stop("x not a list")
    n <- length(x)
    if(n==0) return(NULL)
    x <- lapply(x, function(y) if(length(y)) as.matrix(y))
    d <- array(unlist(lapply(x, dim)), c(2, n))
    rr <- d[1,]
    cc <- d[2,]
    rsum <- sum(rr)
    csum <- sum(cc)
    out <- array(0, c(rsum, csum))
    ind <- array(0, c(4, n))
    rcum <- cumsum(rr)
    ccum <- cumsum(cc)
```

```
    ind[1,-1] <- rcum[-n]
    ind[2,] <- rcum
    ind[3,-1] <- ccum[-n]
    ind[4,] <- ccum
    imat <- array(1:(rsum * csum), c(rsum, csum))
    iuse <- apply(ind, 2, function(y, imat) imat[(y[1]+1):y[2],
            (y[3]+1):y[4]], imat=imat)
    iuse <- as.vector(unlist(iuse))
    out[iuse] <- unlist(x)
    return(out)
}

# Inverse of variance-covariance matrix of y function
VInv <- function(Group,v.u,v.e){
    G <- length(unique(Group))
    N <- length(Group)
    Z <- matrix(0,nrow=N,ncol=G)
    for (i in 1:N){
        j <- Group[i]
        Z[i,j] <- 1
    }
    zeta <- v.u*diag(G)
    R <- v.e*diag(N)
    V <- Z%*%zeta%*%t(Z)+R
    VInv <- solve(V)
    VInv
}

# J diagonal function
Jdiag <- function(Group,Block,q,r){
    G <- length(unique(Group))
    J.l <- list()
    for (g in 1:G) {
        nqg <- sum(Block==q & Group==g)
        nrg <- sum(Block==r & Group==g)
        J.g <- matrix(1,nrow=nqg,ncol=nrg)
        J.l[[g]] <- J.g
    }
    J.diag <- bdiag(J.l[1:G])
    J.diag
}

# B and C matrices function
BCfun <- function(Group,N){
    G <- length(unique(Group))
    H <- length(unique(Group))
    B <- matrix(0,ncol=N,nrow=N)
    C <- matrix(0,ncol=N,nrow=N)
```

```
    for (i in 1:N){
        for (j in 1:N){
            if(Group[i]==Group[j])
            {
                B[i,j] <- 1/sum(Group==Group[i])-1/N
                if(i==j) C[i,j] <- 1-1/sum(Group==Group[i])
                else C[i,j] <- -1/sum(Group==Group[i])
            }
            else
            {
                B[i,j] <- -1/N
                C[i,j] <- 0
            }
        }
    }
    list(B=B,C=C)
}

# Function to calculate Vq matrix
Vfun3 <- function(f,lambda) {
    M <- length(f)
    fbar1 <- mean(f)
    fbar2 <- mean(f^2)
    diag(c((1-lambda)*(lambda*(f-fbar1)^2+fbar2-fbar1^2)))
}

# Simulates value of random permutation matrix A of size N given
exchangeable errors with lambda=Pr(correct linkage)

Afun <- function(N,lambda) {
    A <- matrix(0,nrow=N,ncol=N)
    set <- NULL
    for (i in 1:N) {
        r <- runif(1)
        if(r <= lambda){
            set <- c(set,i)
            A[i,i] <- 1
        }
    }
    wrong <- (1:N)[-set]
    for (i in wrong) {
        valid <- unique((1:N)[-c(i,set)])
        if(length(valid)>1) j <- sample(valid,size=1)
        else {
            if(length(valid)==0) j <- i
            else j <- valid
        }
        A[i,j] <- 1
```

```
        set <- c(set,j)
    }
    A
}

# Given unbiased estimate of beta, calculates ANOVA estimate
of sigmasq.
Rhohat <- function(lambda,X,Ystar,Block,Group,beta) {
    Q <- length(lambda)
    R <- length(lambda)
    G <- length(unique(Group))
    N <- length(Group)
    a1 <- 0
    a2 <- 0
    b <- 0
    c1 <- 0
    c2 <- 0
    d <- 0
    m1.1 <- 0
    m1.2 <- 0
    m2 <- 0
    m3 <- 0
    n1.1 <- 0
    n1.2 <- 0
    n2 <- 0
    n3 <- 0
    BC <- BCfun(Group=Group,N=N)
    B <- BC$B
    C <- BC$C
    idnum <- c(1:N)
    Eq.l <- list()
    for (q in 1:Q) {
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Eq.l[[q]] <- Eq
        Xq <- X[Block==q,]
        Yq <- Y[Block==q]
        Ystarq <- Ystar[Block==q]
        fq <- c(Xq%*%matrix(beta,ncol=1))
        Vq <- Vfun3(f=fq,lambda=lambdaq)
        Kq <- Kfun(M=Mq,lambda=lambdaq,Group=Group,q=q)
        for (r in 1:R) {
            Mr <- sum(Block==r)
            lambdar <- lambda[r]
            Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
            diag(Er) <- lambdar
```

```
            Xr <- X[Block==r,]
            Yr <- Y[Block==r]
            Ystarr <- Ystar[Block==r]
            fr <- c(Xr%*%matrix(beta,ncol=1))
            Jq <- Jdiag(Group=Group,Block=Block,q=q,r=r)
            Bqr <- BC$B[idnum[Block==q],idnum[Block==r]]
            Cqr <- BC$C[idnum[Block==q],idnum[Block==r]]
            if (q==r){
                a1 <- a1+tr(Bqr%*%Kq)
                b <- b+tr(Bqr)
                c1 <- c1+tr(Cqr%*%Kq)
                d <- d+tr(Cqr)
                m1.1 <- m1.1+(matrix(Ystarq,nrow=1)%*%Bqr%*%
                        matrix(Ystarq,ncol=1))
                m2 <- m2+tr(Bqr%*%Vq)+(matrix(fq,nrow=1)%*%
                        t(Eq)%*%Bqr%*%Eq%*%matrix(fq,ncol=1))
                n1.1 <- n1.1+(matrix(Ystarq,nrow=1)%*%Cqr%*%
                        matrix(Ystarq,ncol=1))
                n2 <- n2+tr(Cqr%*%Vq)+(matrix(fq,nrow=1)%*%
                        t(Eq)%*%Cqr%*%Eq%*%matrix(fq,ncol=1))
            }
            else {
                a2 <- a2+tr(t(Jq)%*%t(Eq)%*%Bqr%*%Er)
                c2 <- c2+tr(t(Jq)%*%t(Eq)%*%Cqr%*%Er)
                m1.2 <- m1.2+(matrix(Ystarq,nrow=1)%*%Bqr%*%
                        matrix(Ystarr,ncol=1))
                m3 <- m3+(matrix(fq,nrow=1)%*%t(Eq)%*%Bqr%*%
                        Er%*%matrix(fr,ncol=1))
                n1.2 <- n1.2+(matrix(Ystarq,nrow=1)%*%Cqr%*%
                        matrix(Ystarr,ncol=1))
                n3 <- n3+(matrix(fq,nrow=1)%*%t(Eq)%*%Cqr%*%Er%*%
                        matrix(fr,ncol=1))
            }
        }
    }
    a <- a1+a2
    c <- c1+c2
    m1 <- m1.1+m1.2
    n1 <- n1.1+n1.2
    m <- c(m1)-c(m2)-c(m3)
    n <- c(n1)-c(n2)-c(n3)

    if((m*c-n*a)/(b*c-d*a)<0)
    sigmasq.e <- 0
    else sigmasq.e <- (m*c-n*a)/(b*c-d*a)
    if((m-sigmasq.e*b)/a<0) sigmasq.u <- 0
    else sigmasq.u <- (m-sigmasq.e*b)/a
```

```
    D <- B*c-C*a
    Sigma <- solve(VstarInv(lambda=lambda,Group=Group,Block=Block,
            beta=beta,v.u=sigmasq.u,v.e=sigmasq.e))
    f <- X%*%matrix(beta,ncol=1)
    E <- bdiag(Eq.l[1:Q])
    var.sigmasq.e <- 1/(b*c-d*a)^2*(2*tr(D%*%Sigma%*%D%*%Sigma)+
                    4*t(f)%*%t(E)%*%D%*%Sigma%*%D%*%E%*%f)
    var.sigmasq.u <- 1/a^2*(2*tr(B%*%Sigma%*%B%*%Sigma)+4*t(f)%*%t(E)
                    %*%B%*%Sigma%*%B%*%E%*%f-b^2*var.sigmasq.e)

# Output
    list(sigmasq.u=sigmasq.u,sigmasq.e=sigmasq.e,
    var.sigmasq.e=var.sigmasq.e,var.sigmasq.u=var.sigmasq.u)
}

# Kq matrix.
Kfun <- function(M,lambda,Group,q){
    G <- length(unique(Group))
    H <- length(unique(Group))
    K <- matrix(0,ncol=M,nrow=M)
    for (i in 1:M){
        for (j in 1:M){
            nqh <- sum(Block==q & Group==Group[j])
            if(Group[i]==Group[j])
            {
                if(i==j) K[i,j] <- lambda+(1-lambda)/(M-1)*(G*nqh-1)
                else K[i,j] <- (lambda+(nqh-2)*(1-lambda)/(M-1))^2+(nqh-1)
                                *((1-lambda)/(M-1))^2*(1+(G-1)*nqh)
            }
            else K[i,j] <- (nqh-1)*(1-lambda)/(M-1)*(2*lambda+(1-lambda)
                            /(M-1)*(G*nqh-2))
        }
    }
    K
}

# Inverse of variance-covariance matrix of ystar
VstarInv <- function(lambda,Group,Block,beta,v.u,v.e){
    Q <- length(unique(Block))
    R <- length(unique(Block))
    Vstar <- NULL
    for (q in 1:Q){
        Vstarrow <- NULL
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Iq <- diag(Mq)
```

```
        Xq <- X[Block==q,]
        fq <- c(Xq%*%matrix(beta,ncol=1))
        Kq <- Kfun(M=Mq,lambda=lambdaq,Group=Group,q=q)
        Vq <- Vfun3(f=fq,lambda=lambdaq)
        for (r in 1:R) {
            Mr <- sum(Block==r)
            lambdar <- lambda[r]
            Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
            diag(Er) <- lambdar
            Jq <- Jdiag(Group=Group,Block=Block,q=q,r=r)
            if (q==r) {
                Vstarqq <- v.u*Kq+v.e*Iq+Vq
                Vstarrow <- cbind(Vstarrow,Vstarqq)
            }
            else {
                Vstarqr <- v.u*(Eq%*%Jq%*%t(Er))
                Vstarrow <- cbind(Vstarrow,Vstarqr)
            }
        }
        Vstar <- rbind(Vstar,Vstarrow)
    }
    Vstar.Inv <- solve(Vstar)
    Vstar.Inv
}

# First derivative of variance-covarinace matrix of ystar with respect
to sigmasq_u
DeVBigsigma <- function(Block,Group,lambda){
    Q <- length(unique(Block))
    R <- length(unique(Block))
    Firstdiff <- NULL
    for (q in 1:Q){
        Firstdiffrow <- NULL
        Mq <- sum(Block==q)
        lambdaq <- lambda[q]
        Eq <- matrix((1-lambdaq)/(Mq-1),nrow=Mq,ncol=Mq)
        diag(Eq) <- lambdaq
        Kq <- Kfun(M=Mq,lambda=lambdaq,Group=Group,q=q)
        for (r in 1:R) {
            Mr <- sum(Block==r)
            lambdar <- lambda[r]
            Er <- matrix((1-lambdar)/(Mr-1),nrow=Mr,ncol=Mr)
            diag(Er) <- lambdar
            Jq <- Jdiag(Group=Group,Block=Block,q=q,r=r)
            if (q==r) {
                Firstdiffqq <- Kq
                Firstdiffrow <- cbind(Firstdiffrow,Firstdiffqq)
            }
```

```
            else {
                Firstdiffqr <- Eq%*%Jq%*%t(Er)
                Firstdiffrow <- cbind(Firstdiffrow,Firstdiffqr)
            }
        }
        Firstdiff <- rbind(Firstdiff,Firstdiffrow)
    }
    Firstdiff
}

# Set up initial values and simulation parameters

# Number of blocks
Q <- 4
R <- Q

# Number of population in each block
M <- 200
Block <- NULL
for (q in 1:Q){
    Block <- c(Block,rep(q,M))
}

# Number of groups
G <- 50

# Set up Grouping variable
nqg <- M/G
Group <- rep(rep(1:50,each=nqg),Q)
M <- rep(M,Q)
N <- sum(M)

# Actual value of beta and lambda (yij=2+4xij+uj+eij)
a <- 2
b <- 4
Truebeta <- c(a,b)
p <- length(Truebeta)
Truelambda <- c(1.0,0.95,0.85,0.75)

#Initialising
norow <- 800
BMat <- matrix(0,nrow=norow,ncol=16)
VMat <- matrix(0,nrow=norow,ncol=16)
CMat <- matrix(0,nrow=norow,ncol=16)
SMat <- matrix(0,nrow=norow,ncol=16)
SVMat <- matrix(0,nrow=norow,ncol=16)
KMat <- matrix(0,nrow=norow,ncol=16)
```

```
# Start simulation loop
for (iter in 1:norow) {

# Generating population values
    X <- cbind(rep(1,N),runif(N))
    sigma <- c(1,3)
    TrueVar <- sigma^2
    Yfixed <- X%*%matrix(Truebeta,ncol=1)+rnorm(N,0,sigma[2])
    Y <- NULL
    r <- rnorm(G,0,sigma[1])
    for (q in 1:Q){
        for (g in 1:G){
            Y <- c(Y,c(Yfixed[Block==q & Group==g]+r[g]))
        }
    }

# Estimates of beta and variance under perfect linkage
    fit <- lme(Y ~ X[,2], random=~1|Group)
    TrueB <- fit$coefficients$fixed
    Truesigmasq <- c(as.numeric(VarCorr(fit)[1,1]),
                     as.numeric(VarCorr(fit)[2,1]))

# Estimates of standard errors and confidence limits under perfect linkage
    TrueSE <- sqrt(diag(fit$varFix))
    z <- qnorm(0.975)
    TrueLB <- TrueB-z*TrueSE
    TrueUB <- TrueB+z*TrueSE
    TrueLbe.var <- intervals(fit)$reStruct$Group[1]
    TrueUbe.var <- intervals(fit)$reStruct$Group[3]
    TrueLwithin.var <- intervals(fit)$sigma[1]
    TrueUwithin.var <- intervals(fit)$sigma[3]
    TrueLvar <- c(TrueLbe.var,TrueLwithin.var)
    TrueUvar <- c(TrueUbe.var,TrueUwithin.var)
    selog <- sqrt(diag(fit$apVar))
    senolog <- exp(attr(fit$apVar,'Pars')) * selog
    SEsigmasqTrue <- 2 * exp(attr(fit$apVar,'Pars')) * senolog

# Storing perfect linkage results
    BMat[iter,1:2] <- TrueB
    VMat[iter,1:2] <- TrueSE
    CMat[iter,1:2] <- c((TrueLB<Truebeta)*(TrueUB>Truebeta))
    SMat[iter,1:2] <- Truesigmasq
    SVMat[iter,1:2] <- c(SEsigmasqTrue)
    KMat[iter,1:2] <- c((TrueLvar<sigma)*(TrueUvar>sigma))

# Generating actual linked Y values (exchangeable linkage errors within
blocks). Block 1 is known to be perfectly linked
```

```
# Generating ystar
    Ystar <- Y[Block==1]
    for (q in 2:Q){
        Aq <- Afun(N=M[q],lambda=Truelambda[q])
        Ystar <- c(Ystar,c(Aq%*%matrix(Y[Block==q])))
    }
    f4 <- lme(Ystar ~ X[,2], random=~1|Group)
    b_Naive <- f4$coefficients$fixed
    var.u <- as.numeric(VarCorr(f4)[1,1])
    var.e <- as.numeric(VarCorr(f4)[2,1])
    Naivesigmasq <- c(var.u,var.e)

# Estimates of standard errors and confidence limits under imperfect
linkage
    SEb_Naive <- sqrt(diag(f4$varFix))
    z <- qnorm(0.975)
    NaiveLB <- b_Naive-z*SEb_Naive
    NaiveUB <- b_Naive+z*SEb_Naive
    NaiveLbe.var <- intervals(f4)$reStruct$Group[1]
    NaiveUbe.var <- intervals(f4)$reStruct$Group[3]
    NaiveLwithin.var <- intervals(f4)$sigma[1]
    NaiveUwithin.var <- intervals(f4)$sigma[3]
    NaiveLvar <- c(NaiveLbe.var,NaiveLwithin.var)
    NaiveUvar <- c(NaiveUbe.var,NaiveUwithin.var)
    selogNaive <- sqrt(diag(f4$apVar))
    senologNaive <- exp(attr(f4$apVar,'Pars')) * selogNaive
    SEsigmasqNaive <- 2 * exp(attr(fit$apVar,'Pars')) * senologNaive

# Storing imperfect linkage results
    BMat[iter,3:4] <- b_Naive
    VMat[iter,3:4] <- SEb_Naive
    CMat[iter,3:4] <- c((NaiveLB<Truebeta)*(NaiveUB>Truebeta))
    SMat[iter,3:4] <- c(var.u,var.e)
    SVMat[iter,3:4] <- c(SEsigmasqNaive)
    KMat[iter,3:4] <- c((NaiveLvar<sigma)*(NaiveUvar>sigma))

# Calculation and storage of beta estimates plus associated
standard errors and confidence limits given actual values of lambda
    temp <- LinEstPlus(lambda=Truelambda,var.u=var.u,var.e=var.e,X=X,
            Ystar=Ystar,Block=Block,Group=Group)
    BMat[iter,5:6] <- c(temp$b_R)
    BMat[iter,7:8] <- c(temp$b_A)
    BMat[iter,9:10] <- c(temp$b_C)
    BMat[iter,11:12] <- c(temp$b_B)
    BMat[iter,13:14] <- c(temp$b_mle)
    BMat[iter,15:16] <- c(temp$b_REML)

    VMat[iter,5:6] <- c(temp$SEb_R)
```

```
VMat[iter,7:8] <- c(temp$SEb_A)
VMat[iter,9:10] <- c(temp$SEb_C)
VMat[iter,11:12] <- c(temp$SEb_B)
VMat[iter,13:14] <- c(temp$SEb_mle)
VMat[iter,15:16] <- c(temp$SEb_REML)

CMat[iter,5:6] <- c((temp$CLb_R[,1]<Truebeta)
                    *(temp$CLb_R[,2]>Truebeta))
CMat[iter,7:8] <- c((temp$CLb_A[,1]<Truebeta)
                    *(temp$CLb_A[,2]>Truebeta))
CMat[iter,9:10] <- c((temp$CLb_C[,1]<Truebeta)
                    *(temp$CLb_C[,2]>Truebeta))
CMat[iter,11:12] <- c((temp$CLb_B[,1]<Truebeta)
                    *(temp$CLb_B[,2]>Truebeta))
CMat[iter,13:14] <- c((temp$CLb_mle[,1]<Truebeta)
                    *(temp$CLb_mle[,2]>Truebeta))
CMat[iter,15:16] <- c((temp$CLb_REML[,1]<Truebeta)
                    *(temp$CLb_REML[,2]>Truebeta))

SMat[iter,5:6] <- c(temp$rb_R$sigmasq.u,temp$rb_R$sigmasq.e)
SMat[iter,7:8] <- c(temp$rb_A$sigmasq.u,temp$rb_A$sigmasq.e)
SMat[iter,9:10] <- c(temp$rb_C$sigmasq.u,temp$rb_C$sigmasq.e)
SMat[iter,11:12] <- c(temp$rb_B$sigmasq.u,temp$rb_B$sigmasq.e)
SMat[iter,13:14] <- c(temp$sigmau,temp$sigmae)
SMat[iter,15:16] <- c(temp$sigmau2,temp$sigmae2)

SVMat[iter,5:6] <- c(temp$SEsigmasqR)
SVMat[iter,7:8] <- c(temp$SEsigmasqA)
SVMat[iter,9:10] <- c(temp$SEsigmasqC)
SVMat[iter,11:12] <- c(temp$SEsigmasqB)
SVMat[iter,13:14] <- c(temp$SEsigmasqML)
SVMat[iter,15:16] <- c(temp$SEsigmasqREML)

KMat[iter,5:6] <- c((temp$CLVar_R[,1]<TrueVar)
                    *(temp$CLVar_R[,2]>TrueVar))
KMat[iter,7:8] <- c((temp$CLVar_A[,1]<TrueVar)
                    *(temp$CLVar_A[,2]>TrueVar))
KMat[iter,9:10] <- c((temp$CLVar_C[,1]<TrueVar)
                    *(temp$CLVar_C[,2]>TrueVar))
KMat[iter,11:12] <- c((temp$CLVar_B[,1]<TrueVar)
                    *(temp$CLVar_B[,2]>TrueVar))
KMat[iter,13:14] <- c((temp$CLVar_ML[,1]<TrueVar)
                    *(temp$CLVar_ML[,2]>TrueVar))
KMat[iter,15:16] <- c((temp$CLVar_REML[,1]<TrueVar)
                    *(temp$CLVar_REML[,2]>TrueVar))

print(c(iter,BMat[iter,c(2,4,6,8,10,12,14,16)]))
print(c(iter,SMat[iter,c(2,4,6,8,10,12,14,16)]))
```

```
# End simulation loop
}

# Storage of simulation results
NewSim1 <- list(Truebeta=Truebeta,Truelambda=Truelambda,BMat=BMat,
            SMat=SMat,VMat=VMat,CMat=CMat,KMat=KMat)
done <- 1:norow

# Simulation Performance

# % Relative bias on intercept
round(100*apply(NewSim1$BMat[done,c(1,3,5,7,9,11,13,15)]-
NewSim1$Truebeta[1],2,mean)/NewSim1$Truebeta[1],digits=2)

# % Relative bias on slope
round(100*apply(NewSim1$BMat[done,c(2,4,6,8,10,12,14,16)]-
NewSim1$Truebeta[2],2,mean)/NewSim1$Truebeta[2],digits=2)

# % Relative RMSE on intercept
round(100*sqrt(apply((NewSim1$BMat[done,c(1,3,5,7,9,11,13,15)]-
NewSim1$Truebeta[1])^2,2,mean)/NewSim1$Truebeta[1]),digits=2)

# % Relative RMSE on slope
round(100*sqrt(apply((NewSim1$BMat[done,c(2,4,6,8,10,12,14,16)]-
NewSim1$Truebeta[2])^2,2,mean)/NewSim1$Truebeta[2]),digits=2)

# % Relative bias on between-group variance
round(100*apply(NewSim1$SMat[done,c(1,3,5,7,9,11,13,15)]-
sigma[1]^2,2,mean)/sigma[1]^2,digits=2)

# % Relative bias on within-group variance
round(100*apply(NewSim1$SMat[done,c(2,4,6,8,10,12,14,16)]-
sigma[2]^2,2,mean)/sigma[2]^2,digits=2)

# % Relative RMSE on between group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(1,3,5,7,9,11,13,15)]-
sigma[1]^2)^2,2,mean)/sigma[1]^2),digits=2)

# % Relative RMSE on within group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(2,4,6,8,10,12,14,16)]-
sigma[2]^2)^2,2,mean)/sigma[2]^2),digits=2)

# Actual % coverage of nominal 95% CI on intercept
round(100*apply(NewSim1$CMat[done,c(1,3,5,7,9,11,13,15)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on slope
round(100*apply(NewSim1$CMat[done,c(2,4,6,8,10,12,14,16)],2,mean),digits=1)
```

```
# Actual % coverage of nominal 95% CI on between group variance
round(100*apply(NewSim1$KMat[done,c(1,3,5,7,9,11,13,15)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on within group variance
round(100*apply(NewSim1$KMat[done,c(2,4,6,8,10,12,14,16)],2,mean),digits=1)

# Distribution of relative errors of intercept parameter estimates

par(mfrow = c(2, 2))
boxplot(list(
TR=100*(NewSim1$BMat[done,1]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
N=100*(NewSim1$BMat[done,3]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
R=100*(NewSim1$BMat[done,5]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
A=100*(NewSim1$BMat[done,7]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
C=100*(NewSim1$BMat[done,9]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
B=100*(NewSim1$BMat[done,11]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
ML=100*(NewSim1$BMat[done,13]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
RE=100*(NewSim1$BMat[done,15]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1]),
main="Linear Mixed Model: Scenario 1",sub="%RelError(Intercept)",
ylim=c(-50,50))
abline(h=0)

# Distribution of relative errors of slope parameter estimates
boxplot(list(
TR=100*(NewSim1$BMat[done,2]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
N=100*(NewSim1$BMat[done,4]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
R=100*(NewSim1$BMat[done,6]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
A=100*(NewSim1$BMat[done,8]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
C=100*(NewSim1$BMat[done,10]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
B=100*(NewSim1$BMat[done,12]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
ML=100*(NewSim1$BMat[done,14]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
RE=100*(NewSim1$BMat[done,16]-
```

```
NewSim1$Truebeta[2])/NewSim1$Truebeta[2]),
main="Linear Mixed Model: Scenario 1",sub="%RelError(Slope)",
ylim=c(-50,50))
abline(h=0)

# Distribution of relative errors of between-group variance
parameter estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,1]-
sigma[1]^2)/sigma[1]^2,
N=100*(NewSim1$SMat[done,3]-
sigma[1]^2)/sigma[1]^2,
R=100*(NewSim1$SMat[done,5]-
sigma[1]^2)/sigma[1]^2,
A=100*(NewSim1$SMat[done,7]-
sigma[1]^2)/sigma[1]^2,
C=100*(NewSim1$SMat[done,9]-
sigma[1]^2)/sigma[1]^2,
B=100*(NewSim1$SMat[done,11]-
sigma[1]^2)/sigma[1]^2,
ML=100*(NewSim1$SMat[done,13]-
sigma[1]^2)/sigma[1]^2,
RE=100*(NewSim1$SMat[done,15]-
sigma[1]^2)/sigma[1]^2),
main="Linear Mixed Model: Scenario 1",
sub="%RelError(Between group variance)",ylim=c(-130,130))
abline(h=0)

# Distribution of relative errors of within-group variance
parameter estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,2]-
sigma[2]^2)/sigma[2]^2,
N=100*(NewSim1$SMat[done,4]-
sigma[2]^2)/sigma[2]^2,
R=100*(NewSim1$SMat[done,6]-
sigma[2]^2)/sigma[2]^2,
A=100*(NewSim1$SMat[done,8]-
sigma[2]^2)/sigma[2]^2,
C=100*(NewSim1$SMat[done,10]-
sigma[2]^2)/sigma[2]^2,
B=100*(NewSim1$SMat[done,12]-
sigma[2]^2)/sigma[2]^2,
ML=100*(NewSim1$SMat[done,14]-
sigma[2]^2)/sigma[2]^2,
RE=100*(NewSim1$SMat[done,16]-
```

```
sigma[2]^2)/sigma[2]^2),
main="Linear Mixed Model: Scenario 1",
sub="%RelError(Within group variance)",ylim=c(-25,25))
abline(h=0)
```

D.1.2 Simulation of Estimated Lambda Case

```
# Set up initial values and simulation parameters

# Number of blocks
Q <- 4
R <- Q

# Number of people in each block
M <- 200
Block <- NULL
for (q in 1:Q){
    Block <- c(Block,rep(q,M))
}

# Number of groups
G <- 50

# Set up Grouping variable
nqg <- M/G
Group <- rep(rep(1:50,each=nqg),Q)
M <- rep(M,Q)
N <- sum(M)

# Audit sample sizes
m <- rep(25,Q)

# Actual value of beta and lambda (yij=2+4xij+uj+eij)
a <- 2
b <- 4
Truebeta <- c(a,b)
p <- length(Truebeta)
Truelambda <- c(1.0,0.95,0.85,0.75)

#Initialising
norow <- 800
BMat <- matrix(0,nrow=norow,ncol=16)
VMat <- matrix(0,nrow=norow,ncol=16)
CMat <- matrix(0,nrow=norow,ncol=16)
SMat <- matrix(0,nrow=norow,ncol=16)
SVMat <- matrix(0,nrow=norow,ncol=16)
KMat <- matrix(0,nrow=norow,ncol=16)
```

```
LMat <- matrix(0,nrow=norow,ncol=Q)


# Start simulation loop
for (iter in 1:norow) {

# Generating population values
    X <- cbind(rep(1,N),runif(N))
    sigma <- c(1,3)
    TrueVar <- sigma^2
    Yfixed <- X%*%matrix(Truebeta,ncol=1)+rnorm(N,0,sigma[2])
    Y <- NULL
    r <- rnorm(G,0,sigma[1])
    for (q in 1:Q){
        for (g in 1:G){
            Y <- c(Y,c(Yfixed[Block==q & Group==g]+r[g]))
        }
    }

# Estimates of beta and variance under perfect linkage
    fit <- lme(Y ~ X[,2], random=~1|Group)
    TrueB <- fit$coefficients$fixed
    Truesigmasq <- c(as.numeric(VarCorr(fit)[1,1]),
                     as.numeric(VarCorr(fit)[2,1]))

# Estimates of standard errors and confidence limits under perfect linkage
    TrueSE <- sqrt(diag(fit$varFix))
    z <- qnorm(0.975)
    TrueLB <- TrueB-z*TrueSE
    TrueUB <- TrueB+z*TrueSE
    TrueLbe.var <- intervals(fit)$reStruct$Group[1]
    TrueUbe.var <- intervals(fit)$reStruct$Group[3]
    TrueLwithin.var <- intervals(fit)$sigma[1]
    TrueUwithin.var <- intervals(fit)$sigma[3]
    TrueLvar <- c(TrueLbe.var,TrueLwithin.var)
    TrueUvar <- c(TrueUbe.var,TrueUwithin.var)
    selog <- sqrt(diag(fit$apVar))
    senolog <- exp(attr(fit$apVar,'Pars')) * selog
    SEsigmasqTrue <- 2 * exp(attr(fit$apVar,'Pars')) * senolog

# Storing perfect linkage results
    BMat[iter,1:2] <- TrueB
    SMat[iter,1:2] <- Truesigmasq
    VMat[iter,1:2] <- TrueSE
    CMat[iter,1:2] <- c((TrueLB<Truebeta)*(TrueUB>Truebeta))
    SVMat[iter,1:2] <- c(SEsigmasqTrue)
    KMat[iter,1:2] <- c((TrueLvar<sigma)*(TrueUvar>sigma))
```

```
# Generating actual linked Y values (exchangeable linkage errors
within blocks). Block 1 is known to be perfectly linked
# Generating ystar
    Ystar <- Y[Block==1]
    lambdaHat <- rep(1,Q)
    for (q in 2:Q){
        Aq <- Afun(N=M[q],lambda=Truelambda[q])
        Ystar <- c(Ystar,c(Aq%*%matrix(Y[Block==q])))
        lambdaq <- mean(sample(diag(Aq),size=m[q]))
# Top-truncated estimate of lambda avoids being over-optimistic when
audit sample shows no linkage errors
        lambdaHat[q] <- min((m[q]-0.5)/m[q],max(1/M[q],lambdaq))
    }

# Storage of estimated lambda values
    LMat[iter,] <- lambdaHat

# Estimated lambda used instead of actual value
    Estlambda <- lambdaHat

# Estimates of beta and variance under imperfect linkage
    f4 <- lme(Ystar ~ X[,2], random=~1|Group)
    b_Naive <- f4$coefficients$fixed
    var.u <- as.numeric(VarCorr(f4)[1,1])
    var.e <- as.numeric(VarCorr(f4)[2,1])
    Naivesigmasq <- c(var.u,var.e)

# Estimates of standard errors and confidence limits under imperfect
linkage
    SEb_Naive <- sqrt(diag(f4$varFix))
    z <- qnorm(0.975)
    NaiveLB <- b_Naive-z*SEb_Naive
    NaiveUB <- b_Naive+z*SEb_Naive
    NaiveLbe.var <- intervals(f4)$reStruct$Group[1]
    NaiveUbe.var <- intervals(f4)$reStruct$Group[3]
    NaiveLwithin.var <- intervals(f4)$sigma[1]
    NaiveUwithin.var <- intervals(f4)$sigma[3]
    NaiveLvar <- c(NaiveLbe.var,NaiveLwithin.var)
    NaiveUvar <- c(NaiveUbe.var,NaiveUwithin.var)
    selogNaive <- sqrt(diag(f4$apVar))
    senologNaive <- exp(attr(f4$apVar,'Pars')) * selogNaive
    SEsigmasqNaive <- 2 * exp(attr(fit$apVar,'Pars')) * senologNaive

# Storing imperfect linkage results
    BMat[iter,3:4] <- b_Naive
    SMat[iter,3:4] <- c(var.u,var.e)
    VMat[iter,3:4] <- SEb_Naive
    CMat[iter,3:4] <- c((NaiveLB<Truebeta)*(NaiveUB>Truebeta))
```

```
    SVMat[iter,3:4] <- c(SEsigmasqNaive)
    KMat[iter,3:4] <- c((NaiveLvar<sigma)*(NaiveUvar>sigma))

# Calculation and storage of beta estimates plus associated
standard errors and confidence limits given estimate values of lambda
    temp <- LinEstPlus(lambda=Estlambda,var.u=var.u,var.e=var.e,X=X,
            Ystar=Ystar,Block=Block,Group=Group,m=m)
    BMat[iter,5:6] <- c(temp$b_R)
    BMat[iter,7:8] <- c(temp$b_A)
    BMat[iter,9:10] <- c(temp$b_C)
    BMat[iter,11:12] <- c(temp$b_B)
    BMat[iter,13:14] <- c(temp$b_mle)
    BMat[iter,15:16] <- c(temp$b_REML)

    VMat[iter,5:6] <- c(temp$SEb_R)
    VMat[iter,7:8] <- c(temp$SEb_A)
    VMat[iter,9:10] <- c(temp$SEb_C)
    VMat[iter,11:12] <- c(temp$SEb_B)
    VMat[iter,13:14] <- c(temp$SEb_mle)
    VMat[iter,15:16] <- c(temp$SEb_REML)

    CMat[iter,5:6] <- c((temp$CLb_R[,1]<Truebeta)
                        *(temp$CLb_R[,2]>Truebeta))
    CMat[iter,7:8] <- c((temp$CLb_A[,1]<Truebeta)
                        *(temp$CLb_A[,2]>Truebeta))
    CMat[iter,9:10] <- c((temp$CLb_C[,1]<Truebeta)
                        *(temp$CLb_C[,2]>Truebeta))
    CMat[iter,11:12] <- c((temp$CLb_B[,1]<Truebeta)
                        *(temp$CLb_B[,2]>Truebeta))
    CMat[iter,13:14] <- c((temp$CLb_mle[,1]<Truebeta)
                        *(temp$CLb_mle[,2]>Truebeta))
    CMat[iter,15:16] <- c((temp$CLb_REML[,1]<Truebeta)
                        *(temp$CLb_REML[,2]>Truebeta))

    SMat[iter,5:6] <- c(temp$rb_R$sigmasq.u,temp$rb_R$sigmasq.e)
    SMat[iter,7:8] <- c(temp$rb_A$sigmasq.u,temp$rb_A$sigmasq.e)
    SMat[iter,9:10] <- c(temp$rb_C$sigmasq.u,temp$rb_C$sigmasq.e)
    SMat[iter,11:12] <- c(temp$rb_B$sigmasq.u,temp$rb_B$sigmasq.e)
    SMat[iter,13:14] <- c(temp$sigmau,temp$sigmae)
    SMat[iter,15:16] <- c(temp$sigmau2,temp$sigmae2)

    SVMat[iter,5:6] <- c(temp$SEsigmasqR)
    SVMat[iter,7:8] <- c(temp$SEsigmasqA)
    SVMat[iter,9:10] <- c(temp$SEsigmasqC)
    SVMat[iter,11:12] <- c(temp$SEsigmasqB)
    SVMat[iter,13:14] <- c(temp$SEsigmasqML)
    SVMat[iter,15:16] <- c(temp$SEsigmasqREML)
```

```
    KMat[iter,5:6] <- c((temp$CLVar_R[,1]<TrueVar)
                        *(temp$CLVar_R[,2]>TrueVar))
    KMat[iter,7:8] <- c((temp$CLVar_A[,1]<TrueVar)
                        *(temp$CLVar_A[,2]>TrueVar))
    KMat[iter,9:10] <- c((temp$CLVar_C[,1]<TrueVar)
                        *(temp$CLVar_C[,2]>TrueVar))
    KMat[iter,11:12] <- c((temp$CLVar_B[,1]<TrueVar)
                        *(temp$CLVar_B[,2]>TrueVar))
    KMat[iter,13:14] <- c((temp$CLVar_ML[,1]<TrueVar)
                        *(temp$CLVar_ML[,2]>TrueVar))
    KMat[iter,15:16] <- c((temp$CLVar_REML[,1]<TrueVar)
                        *(temp$CLVar_REML[,2]>TrueVar))

    print(c(iter,BMat[iter,c(2,4,6,8,10,12,14,16)]))
    print(c(iter,SMat[iter,c(2,4,6,8,10,12,14,16)]))

# End simulation loop
}

# Storage of simulation results

NewSim1 <- list(Truebeta=Truebeta,Truelambda=Truelambda,BMat=BMat,
SMat=SMat,VMat=VMat,CMat=CMat,LMat=LMat,KMat=KMat)
done <- 1:norow

# Simulation Performance

# % Relative bias on intercept
round(100*apply(NewSim1$BMat[done,c(1,3,5,7,9,11,13,15)]-
NewSim1$Truebeta[1],2,mean)/NewSim1$Truebeta[1],digits=2)

# % Relative bias on slope
round(100*apply(NewSim1$BMat[done,c(2,4,6,8,10,12,14,16)]-
NewSim1$Truebeta[2],2,mean)/NewSim1$Truebeta[2],digits=2)

# % Relative RMSE on intercept
round(100*sqrt(apply((NewSim1$BMat[done,c(1,3,5,7,9,11,13,15)]-
NewSim1$Truebeta[1])^2,2,mean)/NewSim1$Truebeta[1]),digits=2)

# % Relative RMSE on slope
round(100*sqrt(apply((NewSim1$BMat[done,c(2,4,6,8,10,12,14,16)]-
NewSim1$Truebeta[2])^2,2,mean)/NewSim1$Truebeta[2]),digits=2)

# % Relative bias on between-group variance
round(100*apply(NewSim1$SMat[done,c(1,3,5,7,9,11,13,15)]-
sigma[1]^2,2,mean)/sigma[1]^2,digits=2)

# % Relative bias on within-group variance
```

```
round(100*apply(NewSim1$SMat[done,c(2,4,6,8,10,12,14,16)]-
sigma[2]^2,2,mean)/sigma[2]^2,digits=2)

# % Relative RMSE on between group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(1,3,5,7,9,11,13,15)]-
sigma[1]^2)^2,2,mean)/sigma[1]^2),digits=2)

# % Relative RMSE on within group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(2,4,6,8,10,12,14,16)]-
sigma[2]^2)^2,2,mean)/sigma[2]^2),digits=2)

# Actual % coverage of nominal 95% CI on intercept
round(100*apply(NewSim1$CMat[done,c(1,3,5,7,9,11,13,15)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on slope
round(100*apply(NewSim1$CMat[done,c(2,4,6,8,10,12,14,16)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on between group variance
round(100*apply(NewSim1$KMat[done,c(1,3,5,7,9,11,13,15)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on within group variance
round(100*apply(NewSim1$KMat[done,c(2,4,6,8,10,12,14,16)],2,mean),digits=1)

# Distribution of relative errors of intercept parameter estimates
boxplot(list(
TR=100*(NewSim1$BMat[done,1]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
N=100*(NewSim1$BMat[done,3]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
R=100*(NewSim1$BMat[done,5]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
A=100*(NewSim1$BMat[done,7]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
C=100*(NewSim1$BMat[done,9]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
B=100*(NewSim1$BMat[done,11]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
ML=100*(NewSim1$BMat[done,13]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
RE=100*(NewSim1$BMat[done,15]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1]),
main="Linear Mixed Model: Scenario 2",sub="%RelError(Intercept)",
ylim=c(-50,50))
abline(h=0)

# Distribution of relative errors of slope parameter estimates
boxplot(list(
TR=100*(NewSim1$BMat[done,2]-
```

```
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
N=100*(NewSim1$BMat[done,4]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
R=100*(NewSim1$BMat[done,6]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
A=100*(NewSim1$BMat[done,8]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
C=100*(NewSim1$BMat[done,10]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
B=100*(NewSim1$BMat[done,12]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
ML=100*(NewSim1$BMat[done,14]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
RE=100*(NewSim1$BMat[done,16]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2]),
main="Linear Mixed Model: Scenario 2",sub="%RelError(Slope)",
ylim=c(-50,50))
abline(h=0)

# Distribution of relative errors of between-group variance
parameter estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,1]-
sigma[1]^2)/sigma[1]^2,
N=100*(NewSim1$SMat[done,3]-
sigma[1]^2)/sigma[1]^2,
R=100*(NewSim1$SMat[done,5]-
sigma[1]^2)/sigma[1]^2,
A=100*(NewSim1$SMat[done,7]-
sigma[1]^2)/sigma[1]^2,
C=100*(NewSim1$SMat[done,9]-
sigma[1]^2)/sigma[1]^2,
B=100*(NewSim1$SMat[done,11]-
sigma[1]^2)/sigma[1]^2,
ML=100*(NewSim1$SMat[done,13]-
sigma[1]^2)/sigma[1]^2,
RE=100*(NewSim1$SMat[done,15]-
sigma[1]^2)/sigma[1]^2),
main="Linear Mixed Model: Scenario 2",
sub="%RelError(Between group variance)",ylim=c(-130,130))
abline(h=0)

# Distribution of relative errors of within-group variance
parameter estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,2]-
```

```
sigma[2]^2)/sigma[2]^2,
N=100*(NewSim1$SMat[done,4]-
sigma[2]^2)/sigma[2]^2,
R=100*(NewSim1$SMat[done,6]-
sigma[2]^2)/sigma[2]^2,
A=100*(NewSim1$SMat[done,8]-
sigma[2]^2)/sigma[2]^2,
C=100*(NewSim1$SMat[done,10]-
sigma[2]^2)/sigma[2]^2,
B=100*(NewSim1$SMat[done,12]-
sigma[2]^2)/sigma[2]^2,
ML=100*(NewSim1$SMat[done,14]-
sigma[2]^2)/sigma[2]^2,
RE=100*(NewSim1$SMat[done,16]-
sigma[2]^2)/sigma[2]^2),
main="Linear Mixed Model: Scenario 2",
sub="%RelError(Within group variance)",ylim=c(-25,25))
abline(h=0)
```

D.2 Simulation of Linear Mixed Model with Longitudinal Linked Data

D.2.1 Simulation of Known Lambda Case

```
rm(list = ls())
library(nlme)
library(psych)

LinEstPlus <- function(lambda,var.u,var.e,X,Xstar,Ystar,Block,Group,
    Time,alpha=0.05,nits=5,m){

# Setting up
    G <- length(unique(Group))
    Q <- length(unique(Block))
    T <- length(unique(Time))
    N <- nrow(X)
    p <- ncol(X)
    U1 <- matrix(0,nrow=p,ncol=p)
    U2 <- matrix(0,nrow=p,ncol=1)

# Loop to calculate primitives for initial beta
    for (q in 1:Q){
        Xstarq <- Xstar[Block==q,]
        Ystarq <- Ystar[Block==q]
        Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
        Vq <- Vfun3(X=X,beta=b_Naive,q=q,lambda=lambda,Time=Time,
```

```
            Block=Block)
        Sigmaq <- var.u*Kq+var.e*diag(M[q]*T)+Vq
        SigmaInvq <- solve(Sigmaq)
        U1 <- U1+(t(Xstarq)%*%SigmaInvq%*%Xstarq)
        U2 <- U2+(t(Xstarq)%*%SigmaInvq%*%matrix(Ystarq,ncol=1))
    }

# Calculation of initial beta estimates
    U.1Inv <- solve(U1)
    b_int <- U.1Inv%*%U2

# Iterative calculation of beta
    b <- b_int

# Initial estimates of sigmasq for beta
    rb <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=
        Group,beta=b,Time=Time)

# Start iterative calculation loop for beta
    for (iter in 1:5) {
        U3 <- matrix(0,nrow=p,ncol=p)
        U4 <- matrix(0,nrow=p,ncol=1)
        for (q in 1:Q){
            Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
            Vq <- Vfun3(X=X,beta=b,q=q,lambda=lambda,Time=Time,Block=Block)
            Sigmaq <- rb$sigmasq.u*Kq+rb$sigmasq.e*diag(M[q]*T)+Vq
            SigmaInvq <- solve(Sigmaq)
            Xstarq <- Xstar[Block==q,]
            Ystarq <- Ystar[Block==q]
            U3 <- U3+(t(Xstarq)%*%SigmaInvq%*%Xstarq)
            U4 <- U4+(t(Xstarq)%*%SigmaInvq%*%matrix(Ystarq,ncol=1))
        }

# Updated estimates of beta
        U.3Inv <- solve(U3)
        b <- U.3Inv%*%U4

# Updated estimate of sigmasq
        rb <- Rhohat(lambda=lambda,X=X,Ystar=Ystar,Block=Block,Group=Group,
            beta=b,Time=Time)

# End iterative calculation loop
    }

# Confidence limits of variance components
    sigmasq <- rbind(rb$sigmasq.u,rb$sigmasq.e)
    var.sigmasq <- rbind(rb$var.sigmasq.u,rb$var.sigmasq.e)
    SEsigmasq <- sqrt(var.sigmasq)
```

```
    CLVar <- cbind(sigmasq-z*SEsigmasq,sigmasq+z*SEsigmasq)

# ML and REML estimators
# Setting up initial values
    b_mle <- b_Naive
    b_REML <- b_Naive
    sigmau <- var.u
    sigmae <- var.e
    logsigmau <- log(sqrt(sigmau))
    logsigmae <- log(sqrt(sigmae))
    sigmau2 <- var.u
    sigmae2 <- var.e
    logsigmau2 <- log(sqrt(sigmau2))
    logsigmae2 <- log(sqrt(sigmae2))
    theta <- matrix(c(b_mle,logsigmau,logsigmae),ncol=1)
    theta2 <- matrix(c(logsigmau2,logsigmae2),ncol=1)

# Iterative calculation of b_mle and variance components
    for (i in 1:5){
        I11 <- matrix(0,nrow=p,ncol=p)
        I22 <- 0
        I23 <- 0
        I33 <- 0
        IR11 <- 0
        IR12 <- 0
        IR21 <- 0
        IR22 <- 0
        Diffl_Beta <- matrix(0,nrow=p,ncol=1)
        Diffl_sigmau <- 0
        Diffl_sigmae <- 0
        DifflR_sigmau <- 0
        DifflR_sigmae <- 0
        I12 <- matrix(0,nrow=2,ncol=1)
        I13 <- I12
        I21 <- t(I12)
        I31 <- t(I13)
        U5 <- matrix(0,nrow=p,ncol=p)
        U6 <- matrix(0,nrow=p,ncol=1)
        for (q in 1:Q){
            Eq <- Eqfun(q=q,lambda=lambda,Time=Time)
            Xq <- X[Block==q,]
            Xstarq <- Xstar[Block==q,]
            Ystarq <- Ystar[Block==q]
            Hq <- Eq%*%Xq
            fq <- Xq%*%matrix(b_mle,ncol=1)
            Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
            Vq1 <- Vfun3(X=X,beta=b_mle,q=q,lambda=lambda,Time=Time,
                    Block=Block)
```

```
Vq2 <- Vfun3(X=X,beta=b_REML,q=q,lambda=lambda,Time=Time,
        Block=Block)
Sigmaq1 <- sigmau*Kq+sigmae*diag(M[q]*T)+Vq1
SigmaInvq1 <- solve(Sigmaq1)
Sigmaq2 <- sigmau2*Kq+sigmae2*diag(M[q]*T)+Vq2
SigmaInvq2 <- solve(Sigmaq2)
Mq <- SigmaInvq2-SigmaInvq2%*%Hq%*%solve(t(Hq)%*%SigmaInvq2
        %*%Hq)%*%t(Hq)%*%SigmaInvq2
I11 <- I11+t(Hq)%*%SigmaInvq1%*%Hq
I22 <- I22+2*sigmau*tr(SigmaInvq1%*%Kq)-2*(sigmau^2)*
        tr(SigmaInvq1%*%Kq%*%SigmaInvq1%*%Kq)-
        2*sigmau*t(Ystarq-Eq%*%fq)%*%SigmaInvq1%*%Kq%*%
        SigmaInvq1%*%(Ystarq-Eq%*%fq)+4*(sigmau^2)*t(Ystarq-
        Eq%*%fq)%*%SigmaInvq1%*%Kq%*%SigmaInvq1%*%Kq%*%
        SigmaInvq1%*%(Ystarq-Eq%*%fq)
I23 <- I23-2*sigmau*sigmae*tr(SigmaInvq1%*%SigmaInvq1%*%Kq)+
        4*sigmau*sigmae*t(Ystarq-Eq%*%fq)%*%SigmaInvq1%*%
        SigmaInvq1%*%Kq%*%SigmaInvq1%*%(Ystarq-Eq%*%fq)
I32 <- I23
I33 <- I33+2*sigmae*tr(SigmaInvq1)-2*(sigmae^2)*tr(SigmaInvq1
        %*%SigmaInvq1)-2*sigmae*t(Ystarq-Eq%*%fq)%*%SigmaInvq1
        %*%SigmaInvq1%*%(Ystarq-Eq%*%fq)+4*(sigmae^2)*t(Ystarq
        -Eq%*%fq)%*%SigmaInvq1%*%SigmaInvq1%*%SigmaInvq1%*%
        (Ystarq-Eq%*%fq)

IR11 <- IR11+2*sigmau2*tr(Mq%*%Kq)-2*(sigmau2^2)*tr(Mq%*%Kq%*%
        Mq%*%Kq)-2*sigmau2*matrix(Ystarq,nrow=1)%*%Mq%*%Kq%*%
        Mq%*%matrix(Ystarq,ncol=1)+4*(sigmau2^2)*matrix(Ystarq,
        nrow=1)%*%Mq%*%Kq%*%Mq%*%Kq%*%Mq%*%
        matrix(Ystarq,ncol=1)
IR12 <- IR12-2*sigmau2*sigmae2*tr(Mq%*%Mq%*%Kq)+4*sigmau2*
        sigmae2*matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%Kq%*%Mq%*%
        matrix(Ystarq,ncol=1)
IR21 <- IR12
IR22 <- IR22+2*sigmae2*tr(Mq)-2*(sigmae2^2)*tr(Mq%*%Mq)-2*
        sigmae2*matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%
        matrix(Ystarq,ncol=1)+4*(sigmae2^2)*matrix(Ystarq,
        nrow=1)%*%Mq%*%Mq%*%Mq%*%matrix(Ystarq,ncol=1)
fq <- Xq%*%matrix(b_mle,ncol=1)
Diffl_Beta <- Diffl_Beta+t(Hq)%*%SigmaInvq1%*%(Ystarq-Eq%*%fq)
Diffl_sigmau <- Diffl_sigmau-sigmau*tr(SigmaInvq1%*%Kq)+
                sigmau*t(Ystarq-Eq%*%fq)%*%SigmaInvq1%*%Kq%*%
                SigmaInvq1%*%(Ystarq-Eq%*%fq)
Diffl_sigmae <- Diffl_sigmae-sigmae*tr(SigmaInvq1)+sigmae*
                t(Ystarq-Eq%*%fq)%*%SigmaInvq1%*%SigmaInvq1
                %*%(Ystarq-Eq%*%fq)
DifflR_sigmau <- DifflR_sigmau-sigmau2*tr(Mq%*%Kq)+sigmau2*
                matrix(Ystarq,nrow=1)%*%Mq%*%Kq%*%Mq%*%
```

```
                        matrix(Ystarq,ncol=1)
            DifflR_sigmae <- DifflR_sigmae-sigmae2*tr(Mq)+sigmae2*
                        matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%
                        matrix(Ystarq,ncol=1)
            U5 <- U5+(t(Xstarq)%*%SigmaInvq2%*%Xstarq)
            U6 <- U6+(t(Xstarq)%*%SigmaInvq2%*%matrix(Ystarq,ncol=1))
        }
        I1 <- cbind(I11,I12,I13)
        I2 <- cbind(I21,I22,I23)
        I3 <- cbind(I31,I32,I33)
        IR1 <- cbind(IR11,IR12)
        IR2 <- cbind(IR21,IR22)
        Information <- rbind(I1,I2,I3)
        Information2 <- rbind(IR1,IR2)
        Difftheta <- rbind(Diffl_Beta,Diffl_sigmau,Diffl_sigmae)
        Difftheta2 <- rbind(DifflR_sigmau,DifflR_sigmae)
        theta <- theta+solve(Information)%*%Difftheta
        theta2 <- theta2+solve(Information2)%*%Difftheta2
        b_mle <- theta[1:2,]
        b_REML <- c(solve(U5)%*%U6)
        sigmau <- exp(2*theta[3,])
        sigmae <- exp(2*theta[4,])
        sigmau2 <- exp(2*theta2[1,])
        sigmae2 <- exp(2*theta2[2,])
    }

# Confidence limits of beta estimators
    U7 <- matrix(0,nrow=p,ncol=p)
    U8 <- matrix(0,nrow=p,ncol=p)
    U9 <- matrix(0,nrow=p,ncol=p)
    U10 <- matrix(0,nrow=p,ncol=p)
    for (q in 1:Q){
        Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
        Vqb <- Vfun3(X=X,beta=b,q=q,lambda=lambda,Time=Time,Block=Block)
        Sigmaq <- rb$sigmasq.u*Kq+rb$sigmasq.e*diag(M[q]*T)+Vqb
        SigmaInvq <- solve(Sigmaq)
        VqREML <- Vfun3(X=X,beta=b_REML,q=q,lambda=lambda,Time=Time,
                Block=Block)
        SigmaqREML <- sigmau2*Kq+sigmae2*diag(M[q]*T)+VqREML
        SigmaInvqREML <- solve(SigmaqREML)
        Eq <- Eqfun(q=q,lambda=lambda,Time=Time)
        Xq <- X[Block==q,]
        Ystarq <- Ystar[Block==q]
        U7 <- U7+(t(Xq)%*%Eq%*%SigmaInvq%*%Eq%*%Xq)
        U8 <- U8+(rb$sigmasq.u*t(Xq)%*%Eq%*%SigmaInvq%*%Kq%*%SigmaInvq
                %*%Eq%*%Xq)+(rb$sigmasq.e*t(Xq)%*%Eq%*%SigmaInvq%*%
                SigmaInvq%*%Eq%*%Xq)
        U9 <- U9+(t(Xq)%*%Eq%*%SigmaInvqREML%*%Eq%*%Xq)
```

```
        U10 <- U10+(sigmau2*t(Xq)%*%Eq%*%SigmaInvqREML%*%Kq%*%
                SigmaInvqREML%*%Eq%*%Xq)+(sigmae2*t(Xq)%*%Eq%*%
                SigmaInvqREML%*%SigmaInvqREML%*%Eq%*%Xq)
    }
    U7Inv <- solve(U7)
    Vb <- U7Inv%*%U8%*%U7Inv
    SEb <- sqrt(diag(Vb))
    z <- qnorm(1-alpha/2)
    CLb <- cbind(b-z*SEb,b+z*SEb)

    Vb_mle <- solve(I11)
    SEb_mle <- sqrt(diag(Vb_mle))
    CLb_mle <- cbind(b_mle-z*SEb_mle,b_mle+z*SEb_mle)

    U9Inv <- solve(U9)
    Vb_REML <- U9Inv%*%U10%*%U9Inv
    SEb_REML <- sqrt(diag(Vb_REML))
    CLb_REML <- cbind(b_REML-z*SEb_REML,b_REML+z*SEb_REML)

# Information matrix for confidence interval of variance components
    Info22 <- 0
    Info23 <- 0
    Info33 <- 0
    InfoR11 <- 0
    InfoR12 <- 0
    InfoR22 <- 0
    for (q in 1:Q){
        Eq <- Eqfun(q=q,lambda=lambda,Time=Time)
        Xq <- X[Block==q,]
        Hq <- Eq%*%Xq
        fq <- Xq%*%matrix(b_mle,ncol=1)
        Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
        Vq <- Vfun3(X=X,beta=b_mle,q=q,lambda=lambda,Time=Time,Block=Block)
        Vq2 <- Vfun3(X=X,beta=b_REML,q=q,lambda=lambda,Time=Time,
                Block=Block)
        Sigmaq <- sigmau*Kq+sigmae*diag(M[q]*T)+Vq
        SigmaInvq <- solve(Sigmaq)
        Sigmaq2 <- sigmau2*Kq+sigmae2*diag(M[q]*T)+Vq2
        SigmaInvq2 <- solve(Sigmaq2)
        Mq <- SigmaInvq2-SigmaInvq2%*%Hq%*%solve(t(Hq)%*%SigmaInvq2%*%Hq)
                %*%t(Hq)%*%SigmaInvq2
        Ystarq <- Ystar[Block==q]
        Info22 <- Info22+2*sigmau*tr(SigmaInvq%*%Kq)-2*(sigmau^2)*
                tr(SigmaInvq%*%Kq%*%SigmaInvq%*%Kq)-2*sigmau*t(Ystarq-Eq
                %*%fq)%*%SigmaInvq%*%Kq%*%SigmaInvq%*%(Ystarq-Eq%*%fq)+
                4*(sigmau^2)*t(Ystarq-Eq%*%fq)%*%SigmaInvq%*%Kq%*%
                SigmaInvq%*%Kq%*%SigmaInvq%*%(Ystarq-Eq%*%fq)
        Info23 <- Info23-2*sigmau*sigmae*tr(SigmaInvq%*%SigmaInvq%*%Kq)+
```

```
            4*sigmau*sigmae*t(Ystarq-Eq%*%fq)%*%SigmaInvq%*%SigmaInvq
            %*%Kq%*%SigmaInvq%*%(Ystarq-Eq%*%fq)
    Info33 <- Info33+2*sigmae*tr(SigmaInvq)-2*(sigmae^2)*tr(SigmaInvq
            %*%SigmaInvq)-2*sigmae*t(Ystarq-Eq%*%fq)%*%SigmaInvq%*%
            SigmaInvq%*%(Ystarq-Eq%*%fq)+4*(sigmae^2)*t(Ystarq-Eq%*%fq)
            %*%SigmaInvq%*%SigmaInvq%*%SigmaInvq%*%(Ystarq-Eq%*%fq)

    InfoR11 <- InfoR11+2*sigmau2*tr(Mq%*%Kq)-2*(sigmau2^2)*tr(Mq%*%Kq
            %*%Mq%*%Kq)-2*sigmau2*matrix(Ystarq,nrow=1)%*%Mq%*%Kq%*%
            Mq%*%matrix(Ystarq,ncol=1)+4*(sigmau2^2)*matrix(Ystarq,
            nrow=1)%*%Mq%*%Kq%*%Mq%*%Kq%*%Mq%*%matrix(Ystarq,ncol=1)
    InfoR12 <- InfoR12-2*sigmau2*sigmae2*tr(Mq%*%Mq%*%Kq)+4*sigmau2*
            sigmae2*matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%Kq%*%Mq%*%
            matrix(Ystarq,ncol=1)
    InfoR21 <- InfoR12
    InfoR22 <- InfoR22+2*sigmae2*tr(Mq)-2*(sigmae2^2)*tr(Mq%*%Mq)-
            2*sigmae2*matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%matrix(Ystarq,
            ncol=1)+4*(sigmae2^2)*matrix(Ystarq,nrow=1)%*%Mq%*%Mq%*%
            Mq%*%matrix(Ystarq,ncol=1)
}
Info <- matrix(c(Info22,Info23,Info23,Info33),nrow=2)
InfoR <- matrix(c(InfoR11,InfoR21,InfoR12,InfoR22),nrow=2)

sigmasqML <- rbind(sigmau,sigmae)
var.sigmasqML <- solve(Info)
SEsigmasqML <- sqrt(diag(var.sigmasqML))
CLVaru_ML <- c(sigmasqML[1]*(exp(-z*SEsigmasqML[1]))^2,sigmasqML[1]*
            (exp(z*SEsigmasqML[1]))^2)
CLVare_ML <- c(sigmasqML[2]*(exp(-z*SEsigmasqML[2]))^2,sigmasqML[2]*
            (exp(z*SEsigmasqML[2]))^2)
CLVar_ML <- rbind(CLVaru_ML,CLVare_ML)
varlogu <- diag(var.sigmasqML)[1]
varloge <- diag(var.sigmasqML)[2]
sesigmasquML <- sigmau*exp(2*varlogu)*sqrt(exp(4*varlogu)-1)
sesigmasqeML <- sigmae*exp(2*varloge)*sqrt(exp(4*varloge)-1)
sesigmasqML <- c(sesigmasquML,sesigmasqeML)

sigmasqREML <- rbind(sigmau2,sigmae2)
var.sigmasqREML <- solve(InfoR)
SEsigmasqREML <- sqrt(diag(var.sigmasqREML))
CLVaru_REML <- c(sigmasqREML[1]*(exp(-z*SEsigmasqREML[1]))^2,
                 sigmasqREML[1]*(exp(z*SEsigmasqREML[1]))^2)
CLVare_REML <- c(sigmasqREML[2]*(exp(-z*SEsigmasqREML[2]))^2,
                 sigmasqREML[2]*(exp(z*SEsigmasqREML[2]))^2)
CLVar_REML <- rbind(CLVaru_REML,CLVare_REML)
varlogu2 <- diag(var.sigmasqREML)[1]
varloge2 <- diag(var.sigmasqREML)[2]
sesigmasquREML <- sigmau2*exp(2*varlogu2)*sqrt(exp(4*varlogu2)-1)
```

```
    sesigmasqeREML <- sigmae2*exp(2*varloge2)*sqrt(exp(4*varloge2)-1)
    sesigmasqREML <- c(sesigmasquREML,sesigmasqeREML)

# Output
    list(b=b,SEb=SEb,CLb=CLb,rb=rb,SEsigmasq=SEsigmasq,CLVar=CLVar,
        b_mle=b_mle,SEb_mle=SEb_mle,CLb_mle=CLb_mle,sigmasqML=
        sigmasqML,sesigmasqML=sesigmasqML,CLVar_ML=CLVar_ML,
        b_REML=b_REML,SEb_REML=SEb_REML,CLb_REML=CLb_REML,sigmasqREML
        =sigmasqREML,sesigmasqREML=sesigmasqREML,CLVar_REML=CLVar_REML)

# End of function LinEstPlus
}

# Block-diagonal matrix function
bdiag <- function(x){
    if(!is.list(x)) stop("x not a list")
    n <- length(x)
    if(n==0) return(NULL)
    x <- lapply(x, function(y) if(length(y)) as.matrix(y))
    d <- array(unlist(lapply(x, dim)), c(2, n))
    rr <- d[1,]
    cc <- d[2,]
    rsum <- sum(rr)
    csum <- sum(cc)
    out <- array(0, c(rsum, csum))
    ind <- array(0, c(4, n))
    rcum <- cumsum(rr)
    ccum <- cumsum(cc)
    ind[1,-1] <- rcum[-n]
    ind[2,] <- rcum
    ind[3,-1] <- ccum[-n]
    ind[4,] <- ccum
    imat <- array(1:(rsum * csum), c(rsum, csum))
    iuse <- apply(ind, 2, function(y, imat) imat[(y[1]+1):y[2],
            (y[3]+1):y[4]], imat=imat)
    iuse <- as.vector(unlist(iuse))
    out[iuse] <- unlist(x)
    return(out)
}

# B and C matrices function
BCfun <- function(Group,N){
    G <- length(unique(Group))
    H <- length(unique(Group))
    B <- matrix(0,ncol=N,nrow=N)
    C <- matrix(0,ncol=N,nrow=N)
    for (i in 1:N){
        for (j in 1:N){
```

```
            if(Group[i]==Group[j])
            {
                B[i,j] <- 1/sum(Group==Group[i])-1/N
                if(i==j) C[i,j] <- 1-1/sum(Group==Group[i])
                else C[i,j] <- -1/sum(Group==Group[i])
            }
            else
            {
                B[i,j] <- -1/N
                C[i,j] <- 0
            }
        }
    }
    list(B=B,C=C)
}

# Simulates value of random permutation matrix A of size N given
exchangeable errors with lambda=Pr(correct linkage)

Afun <- function(N,lambda) {
    A <- matrix(0,nrow=N,ncol=N)
    set <- sample(1:N,round(N*lambda))
    for (i in 1:N) {
        A[set[i],set[i]] <- 1
    }
    wrong <- (1:N)[-set]
    for (i in wrong) {
        valid <- unique((1:N)[-c(i,set)])
        if(length(valid)>1) j <- sample(valid,size=1)
        else {
            if(length(valid)==0) j <- i
            else j <- valid
        }
        A[i,j] <- 1
        set <- c(set,j)
    }
    A
}

# Eq function
Eqfun <- function(q,lambda,Time){
    T <- length(unique(Time))
    Eqt.l <- list()
    for (t in 1:T){
        Mqt <- M[q]
        lambdaqt <- lambda[q,t]
        Eqt <- matrix((1-lambdaqt)/(Mqt-1),nrow=Mqt,ncol=Mqt)
        diag(Eqt) <- lambdaqt
```

```
        Eqt.l[[t]] <- Eqt
    }
    E <- bdiag(Eqt.l[1:T])
    E
}

# Function to calculate Vq matrix
Vfun3 <- function(X,beta,q,lambda,Time,Block) {
    T <- length(unique(Time))
    Vi <- NULL
    for (t in 1:T){
        Mqt <- M[q]
        lambdaqt <- lambda[q,t]
        Xqt <- X[Block==q & Time==t,]
        fqt <- c(Xqt%*%matrix(beta,ncol=1))
        fbar1 <- mean(fqt)
        fbar2 <- mean(fqt^2)
        Vi <- c(Vi,c((1-lambdaqt)*(lambdaqt*(fqt-fbar1)^2+fbar2-
            fbar1^2)))
    }
    V <- diag(Vi)
    V
}

# Kq matrix.
Kqfun <- function(Time,lambda,q){
    T <- length(unique(Time))
    S <- length(unique(Time))
    Ez <- NULL
    for (t in 1:T){
        Ezrow <- NULL
        Mqt <- M[q]
        lambdaqt <- lambda[q,t]
        Eqt <- matrix((1-lambdaqt)/(Mqt-1),nrow=Mqt,ncol=Mqt)
        diag(Eqt) <- lambdaqt
        for (s in 1:S){
            Mqs <- M[q]
            lambdaqs <- lambda[q,s]
            Eqs <- matrix((1-lambdaqs)/(Mqs-1),nrow=Mqs,ncol=Mqs)
            diag(Eqs) <- lambdaqs
            if (t==s){
                Eztt <- diag(Mqs)
                Ezrow <- cbind(Ezrow,Eztt)
            }
            else {
                Ezts <- Eqt%*%Eqs
                Ezrow <- cbind(Ezrow,Ezts)
            }
```

```
        }
        Ez <- rbind(Ez,Ezrow)
    }
    Ez
}

# Sigma function
Sigmafun <- function(X,Time,Block,lambda,v.u,v.e,beta){
    Q <- length(unique(Block))
    T <- length(unique(Time))
    Sigma.l <- list()
    for (q in 1:Q){
        Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
        Vq <- Vfun3(X=X,beta=beta,q=q,lambda=lambda,Time=Time,Block=Block)
        Sigmaq <- c(v.u)*Kq+c(v.e)*diag(M[q]*T)+Vq
        Sigma.l[[q]] <- Sigmaq
    }
    Sigma <- bdiag(Sigma.l[1:Q])
    Sigma
}

# Given unbiased estimate of beta, calculates method of moments estimate
of sigmasq.

Rhohat <- function(lambda,X,Ystar,Block,Group,beta,Time) {
    Q <- length(unique(Block))
    R <- length(unique(Block))
    N <- length(Group)
    a <- 0
    b <- 0
    c <- 0
    d <- 0
    m1 <- 0
    m2 <- 0
    n <- 0
    BC <- BCfun(Group=Group)
    B <- BC$B
    C <- BC$C
    idnum <- c(1:N)
    Eq.l <- list()
    for (q in 1:Q) {
        Eq <- Eqfun(q=q,lambda=lambda,Time=Time)
        Kq <- Kqfun(Time=Time,lambda=lambda,q=q)
        Eq.l[[q]] <- Eq
        Xq <- X[Block==q,]
        Ystarq <- Ystar[Block==q]
        fq <- Xq%*%matrix(beta,ncol=1)
        Vq <- Vfun3(X=X,beta=beta,q=q,lambda=lambda,Time=Time,Block=Block)
```

```
        Bqq <- BC$B[idnum[Block==q],idnum[Block==q]]
        Cqq <- BC$C[idnum[Block==q],idnum[Block==q]]
        a <- a+tr(Bqq%*%Kq)
        b <- b+tr(Bqq)
        c <- c+tr(Cqq%*%Kq)
        d <- d+tr(Cqq)
        m2 <- m2-tr(Bqq%*%Vq)
        n <- n+(matrix(Ystarq,nrow=1)%*%Cqq%*%matrix(Ystarq,ncol=1))-
            tr(Cqq%*%Vq)-(t(fq)%*%Eq%*%Cqq%*%Eq%*%fq)
        for (r in 1:R) {
            Er <- Eqfun(q=r,lambda=lambda,Time=Time)
            Xr <- X[Block==r,]
            Ystarr <- Ystar[Block==r]
            fr <- Xr%*%matrix(beta,ncol=1)
            Bqr <- BC$B[idnum[Block==q],idnum[Block==r]]
            m1 <- m1+(matrix(Ystarq,nrow=1)%*%Bqr%*%matrix(Ystarr,ncol=1))
                -(t(fq)%*%Eq%*%Bqr%*%Er%*%fr)
        }
    }
    m <- c(m1)+c(m2)

    if((m*c-n*a)/(b*c-d*a)<0)
    sigmasq.e <- 0
    else sigmasq.e <- (m*c-n*a)/(b*c-d*a)
    if((m-sigmasq.e*b)/a<0) sigmasq.u <- 0
    else sigmasq.u <- (m-sigmasq.e*b)/a

    D <- B*c-C*a
    Sigma <- Sigmafun(X=X,Time=Time,Block=Block,lambda=lambda,v.u=
            sigmasq.u,v.e=sigmasq.e,beta=beta)
    f <- X%*%matrix(beta,ncol=1)
    E <- bdiag(Eq.l[1:Q])
    var.sigmasq.e <- 1/(b*c-d*a)^2*(2*tr(D%*%Sigma%*%D%*%Sigma)+
                     4*t(f)%*%t(E)%*%D%*%Sigma%*%D%*%E%*%f)
    var.sigmasq.u <- 1/a^2*(2*tr(B%*%Sigma%*%B%*%Sigma)+4*t(f)%*%
                    t(E)%*%B%*%Sigma%*%B%*%E%*%f-b^2*var.sigmasq.e)

# Output
    list(sigmasq.u=c(sigmasq.u),sigmasq.e=c(sigmasq.e),
    var.sigmasq.e=var.sigmasq.e,var.sigmasq.u=var.sigmasq.u)
}

### 1. Simulation of known lambda case ###

# Set up initial values and simulation parameters

# Number of blocks
Q <- 3
```

```
R <- Q

# Number of population in each block
M <- c(130,120,100)
n <- sum(M)
T <- 4
N <- n*T
Block <- NULL
for (q in 1:Q){
    Block <- c(Block,rep(q,M[q]*T))
}

# Set up Grouping variable and time
Group <- c(rep(1:M[1],T),rep((M[1]+1):(M[1]+M[2]),T),
        rep((M[1]+M[2]+1):n,T))
G <- length(unique(Group))
Time <- NULL
for (q in 1:Q){
    Time <- c(Time,c(rep(1:T,each=M[q])))
}
Gr <- NULL
for (q in 1:Q){
    Gr <- c(Gr,c(rep(1:M[q],T)))
}

# Actual value of beta and lambda (yij=2+4xij+uj+eij)
a <- 2
b <- 4
Truebeta <- c(a,b)
p <- length(Truebeta)
Truelambda <- matrix(c(rep(1,3),rep(c(0.95,0.9,0.85),3)),nrow=3)

#Initialising
norow <- 800
BMat <- matrix(0,nrow=norow,ncol=10)
VMat <- matrix(0,nrow=norow,ncol=10)
CMat <- matrix(0,nrow=norow,ncol=10)
SMat <- matrix(0,nrow=norow,ncol=10)
SVMat <- matrix(0,nrow=norow,ncol=10)
KMat <- matrix(0,nrow=norow,ncol=10)
FMat <- matrix(0,nrow=norow,ncol=20)

for (iter in 1:norow) {

# Generating population values
    X2 <- NULL
    X2t1 <- runif(n)
    X2t2 <- X2t1+1
```

```
    X2t3 <- X2t2+1
    X2t4 <- X2t3+1
    Bl <- NULL
    for (q in 1:Q){
        Bl <- c(Bl,c(rep(q,M[q])))
    }
    for (q in 1:Q){
        X2 <- c(X2,c(X2t1[Bl==q],X2t2[Bl==q],X2t3[Bl==q],X2t4[Bl==q]))
    }
    X <- cbind(rep(1,N),X2)
    sigma <- c(3,1)
    TrueVar <- sigma^2
    Yfixed <- X%*%matrix(Truebeta,ncol=1)+rnorm(N,0,sigma[2])
    Y <- NULL
    for (q in 1:Q){
        r <- rnorm(M[q],0,sigma[1])
        for (t in 1:T){
            for (g in 1:M[q]){
                Y <- c(Y,c(Yfixed[Block==q & Time==t & Gr==g]+r[g]))
            }
        }
    }

# Estimates of beta and variance under perfect linkage
    fit <- lme(Y ~ X[,2], random=~1|Group, method="REML")
    TrueB <- fit$coefficients$fixed
    Truesigmasq <- c(as.numeric(VarCorr(fit)[1,1]),
                     as.numeric(VarCorr(fit)[2,1]))

# Estimates of standard errors and confidence limits under perfect linkage
    TrueSE <- sqrt(diag(fit$varFix))
    z <- qnorm(0.975)
    TrueLB <- TrueB-z*TrueSE
    TrueUB <- TrueB+z*TrueSE
    TrueLbe.var <- intervals(fit)$reStruct$Group[1]
    TrueUbe.var <- intervals(fit)$reStruct$Group[3]
    TrueLwithin.var <- intervals(fit)$sigma[1]
    TrueUwithin.var <- intervals(fit)$sigma[3]
    TrueLvar <- c(TrueLbe.var,TrueLwithin.var)
    TrueUvar <- c(TrueUbe.var,TrueUwithin.var)
    selog <- sqrt(diag(fit$apVar))
    senolog <- exp(attr(fit$apVar,'Pars')) * selog
    SEsigmasqTrue <- 2 * exp(attr(fit$apVar,'Pars')) * senolog

# Storing perfect linkage results
    BMat[iter,1:2] <- TrueB
    SMat[iter,1:2] <- Truesigmasq
    VMat[iter,1:2] <- TrueSE
```

```
    CMat[iter,1:2] <- c((TrueLB<Truebeta)*(TrueUB>Truebeta))
    SVMat[iter,1:2] <- c(SEsigmasqTrue)
    KMat[iter,1:2] <- c((TrueLvar<sigma)*(TrueUvar>sigma))
    FMat[iter,1:2] <- c(TrueLbe.var^2,TrueUbe.var^2)
    FMat[iter,3:4] <- c(TrueLwithin.var^2,TrueUwithin.var^2)

# Generating actual linked Y and X values

# Generating xstar and ystar
    Ystar <- NULL
    X2star <- NULL
    for (q in 1:Q){
        for (t in 1:T){
            Atq <- Afun(N=M[q],lambda=Truelambda[q,t])
            Ystar <- c(Ystar,c(Atq%*%matrix(Y[Block==q & Time==t])))
            X2star <- c(X2star,c(Atq%*%matrix(X2[Block==q & Time==t])))
        }
    }
    Xstar <- cbind(rep(1,N),X2star)
    f4 <- lme(Ystar ~ Xstar[,2], random=~1|Group, method="REML")
    b_Naive <- f4$coefficients$fixed
    var.u <- as.numeric(VarCorr(f4)[1,1])
    var.e <- as.numeric(VarCorr(f4)[2,1])
    Naivesigmasq <- c(var.u,var.e)

# Estimates of standard errors and confidence limits under imperfect
linkage
    SEb_Naive <- sqrt(diag(f4$varFix))
    z <- qnorm(0.975)
    NaiveLB <- b_Naive-z*SEb_Naive
    NaiveUB <- b_Naive+z*SEb_Naive
    NaiveLbe.var <- intervals(f4)$reStruct$Group[1]
    NaiveUbe.var <- intervals(f4)$reStruct$Group[3]
    NaiveLwithin.var <- intervals(f4)$sigma[1]
    NaiveUwithin.var <- intervals(f4)$sigma[3]
    NaiveLvar <- c(NaiveLbe.var,NaiveLwithin.var)
    NaiveUvar <- c(NaiveUbe.var,NaiveUwithin.var)
    selogNaive <- sqrt(diag(f4$apVar))
    senologNaive <- exp(attr(f4$apVar,'Pars')) * selogNaive
    SEsigmasqNaive <- 2 * exp(attr(fit$apVar,'Pars')) * senologNaive

# Storing imperfect linkage results
    BMat[iter,3:4] <- b_Naive
    SMat[iter,3:4] <- c(var.u,var.e)
    VMat[iter,3:4] <- SEb_Naive
    CMat[iter,3:4] <- c((NaiveLB<Truebeta)*(NaiveUB>Truebeta))
    SVMat[iter,3:4] <- c(SEsigmasqNaive)
    KMat[iter,3:4] <- c((NaiveLvar<sigma)*(NaiveUvar>sigma))
```

```
    FMat[iter,5:6] <- c(NaiveLbe.var^2,NaiveUbe.var^2)
    FMat[iter,7:8] <- c(NaiveLwithin.var^2,NaiveUwithin.var^2)

# Calculation and storage of b, b_mle and b_REML estimates plus
associated standard errors and confidence limits given actual values
of lambda

    temp <- LinEstPlus(lambda=Truelambda,var.u=var.u,var.e=var.e,X=X,
            Xstar=Xstar,Ystar=Ystar,Block=Block,Group=Group,Time=Time)
    BMat[iter,5:6] <- c(temp$b)
    BMat[iter,7:8] <- c(temp$b_mle)
    BMat[iter,9:10] <- c(temp$b_REML)

    VMat[iter,5:6] <- c(temp$SEb)
    VMat[iter,7:8] <- c(temp$SEb_mle)
    VMat[iter,9:10] <- c(temp$SEb_REML)

    CMat[iter,5:6] <- c((temp$CLb[,1]<Truebeta)
                    *(temp$CLb[,2]>Truebeta))
    CMat[iter,7:8] <- c((temp$CLb_mle[,1]<Truebeta)
                    *(temp$CLb_mle[,2]>Truebeta))
    CMat[iter,9:10] <- c((temp$CLb_REML[,1]<Truebeta)
                    *(temp$CLb_REML[,2]>Truebeta))

    SMat[iter,5:6] <- c(temp$rb$sigmasq.u,temp$rb$sigmasq.e)
    SMat[iter,7:8] <- c(temp$sigmasqML[1],temp$sigmasqML[2])
    SMat[iter,9:10] <- c(temp$sigmasqREML[1],temp$sigmasqREML[2])

    SVMat[iter,5:6] <- c(temp$SEsigmasq)
    SVMat[iter,7:8] <- c(temp$sesigmasqML)
    SVMat[iter,9:10] <- c(temp$sesigmasqREML)

    KMat[iter,5:6] <- c((temp$CLVar[,1]<TrueVar)
                        *(temp$CLVar[,2]>TrueVar))
    KMat[iter,7:8] <- c((temp$CLVar_ML[,1]<TrueVar)
                        *(temp$CLVar_ML[,2]>TrueVar))
    KMat[iter,9:10] <- c((temp$CLVar_REML[,1]<TrueVar)
                        *(temp$CLVar_REML[,2]>TrueVar))

    FMat[iter,9:10] <- c(temp$CLVar[1,])
    FMat[iter,11:12] <- c(temp$CLVar[2,])
    FMat[iter,13:14] <- c(temp$CLVar_ML[1,])
    FMat[iter,15:16] <- c(temp$CLVar_ML[2,])
    FMat[iter,17:18] <- c(temp$CLVar_REML[1,])
    FMat[iter,19:20] <- c(temp$CLVar_REML[2,])

    print(c(iter,SMat[iter,c(1,3,5,7,9)]))
    print(c(iter,FMat[iter,c(1,2,5,6,9,10,13,14,17,18)]))
```

```
    print(c(iter,SMat[iter,c(2,4,6,8,10)]))
    print(c(iter,FMat[iter,c(3,4,7,8,11,12,15,16,19,20)]))

# End simulation loop
}

# Storage of simulation results
NewSim1 <- list(Truebeta=Truebeta,Truelambda=Truelambda,BMat=BMat,
            VMat=VMat,CMat=CMat,SMat=SMat,SVMat=SVMat,KMat=KMat,FMat=FMat)
done <- 1:norow

# Simulation Performance

# % Relative bias on intercept (true,b_Naive,b,b_mle,b_REML)
round(100*apply(NewSim1$BMat[done,c(1,3,5,7,9)]-
NewSim1$Truebeta[1],2,mean)/NewSim1$Truebeta[1],digits=2)

# % Relative bias on slope (true,b_Naive,b,b_mle,b_REML)
round(100*apply(NewSim1$BMat[done,c(2,4,6,8,10)]-
NewSim1$Truebeta[2],2,mean)/NewSim1$Truebeta[2],digits=2)

# % Relative RMSE on intercept (true,b_Naive,b,b_mle,b_REML)
round(100*sqrt(apply((NewSim1$BMat[done,c(1,3,5,7,9)]-
NewSim1$Truebeta[1])^2,2,mean)/NewSim1$Truebeta[1]),digits=2)

# % Relative RMSE on slope (true,b_Naive,b,b_mle,b_REML)
round(100*sqrt(apply((NewSim1$BMat[done,c(2,4,6,8,10)]-
NewSim1$Truebeta[2])^2,2,mean)/NewSim1$Truebeta[2]),digits=2)

# % Relative bias on between-group variance
round(100*apply(NewSim1$SMat[done,c(1,3,5,7,9)]-
sigma[1]^2,2,mean)/sigma[1]^2,digits=2)

# % Relative bias on within-group variance
round(100*apply(NewSim1$SMat[done,c(2,4,6,8,10)]-
sigma[2]^2,2,mean)/sigma[2]^2,digits=2)

# % Relative RMSE on between group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(1,3,5,7,9)]-
sigma[1]^2)^2,2,mean)/sigma[1]^2),digits=2)

# % Relative RMSE on within group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(2,4,6,8,10)]-
sigma[2]^2)^2,2,mean)/sigma[2]^2),digits=2)

# Actual % coverage of nominal 95% CI on intercept
round(100*apply(NewSim1$CMat[done,c(1,3,5,7,9)],2,mean),digits=1)
```

```
# Actual % coverage of nominal 95% CI on slope
round(100*apply(NewSim1$CMat[done,c(2,4,6,8,10)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on between group variance
round(100*apply(NewSim1$KMat[done,c(1,3,5,7,9)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on within group variance
round(100*apply(NewSim1$KMat[done,c(2,4,6,8,10)],2,mean),digits=1)

# Bias and RMSE on variance estimator of coefficient
Truese <- 0
Estse <- 0
bias <- 0
RMSE <- 0
for(i in 1:10){
    Truese[i] <- sqrt(sum((NewSim1$BMat[done,i]-
                mean(NewSim1$BMat[done,i]))^2)/norow)
    Estse[i] <- mean(NewSim1$VMat[done,i])
    bias[i] <- round(Estse[i]-Truese[i],digits=3)
    RMSE[i] <- round(sqrt(mean((NewSim1$VMat[done,i]-
                Truese[i])^2)),digits=3)
}
print(bias[c(1,3,5,7,9)])
print(bias[c(2,4,6,8,10)])
print(RMSE[c(1,3,5,7,9)])
print(RMSE[c(2,4,6,8,10)])

# Bias and RMSE on variance estimator of variance components
Truese2 <- 0
Estse2 <- 0
bias2 <- 0
RMSE2 <- 0
for(i in 1:10){
    Truese2[i] <- sqrt(sum((NewSim1$SMat[done,i]-
                mean(NewSim1$SMat[done,i]))^2)/norow)
    Estse2[i] <- mean(NewSim1$SVMat[done,i])
    bias2[i] <- round(Estse2[i]-Truese2[i],digits=3)
    RMSE2[i] <- round(sqrt(mean((NewSim1$SVMat[done,i]-
                Truese2[i])^2)),digits=3)
}
print(bias2[c(1,3,5,7,9)])
print(bias2[c(2,4,6,8,10)])
print(RMSE2[c(1,3,5,7,9)])
print(RMSE2[c(2,4,6,8,10)])

# Distribution of relative errors of intercept parameter estimates
par(mfrow = c(2, 1))
boxplot(list(
```

```
TR=100*(NewSim1$BMat[done,1]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
N=100*(NewSim1$BMat[done,3]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
AN=100*(NewSim1$BMat[done,5]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
ML=100*(NewSim1$BMat[done,7]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
RE=100*(NewSim1$BMat[done,9]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1]),
main="Linear Mixed Model: Scenario 1",sub="%RelError(Intercept)",
ylim=c(-40,40))
abline(h=0)

# Distribution of relative errors of slope parameter estimates
boxplot(list(
TR=100*(NewSim1$BMat[done,2]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
N=100*(NewSim1$BMat[done,4]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
AN=100*(NewSim1$BMat[done,6]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
ML=100*(NewSim1$BMat[done,8]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
RE=100*(NewSim1$BMat[done,10]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2]),
main="Linear Mixed Model: Scenario 1",sub="%RelError(Slope)",
ylim=c(-3,3))
abline(h=0)

# Distribution of relative errors of between-group variance parameter
estimates

par(mfrow = c(2, 1))
boxplot(list(
TR=100*(NewSim1$SMat[done,1]-
sigma[1]^2)/sigma[1]^2,
N=100*(NewSim1$SMat[done,3]-
sigma[1]^2)/sigma[1]^2,
AN=100*(NewSim1$SMat[done,5]-
sigma[1]^2)/sigma[1]^2,
ML=100*(NewSim1$SMat[done,7]-
sigma[1]^2)/sigma[1]^2,
RE=100*(NewSim1$SMat[done,9]-
sigma[1]^2)/sigma[1]^2),
main="Linear Mixed Model: Scenario 1",
sub="%RelError(Between group variance)",ylim=c(-60,60))
abline(h=0)
```

```
# Distribution of relative errors of within-group variance parameter
estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,2]-
sigma[2]^2)/sigma[2]^2,
N=100*(NewSim1$SMat[done,4]-
sigma[2]^2)/sigma[2]^2,
AN=100*(NewSim1$SMat[done,6]-
sigma[2]^2)/sigma[2]^2,
ML=100*(NewSim1$SMat[done,8]-
sigma[2]^2)/sigma[2]^2,
RE=100*(NewSim1$SMat[done,10]-
sigma[2]^2)/sigma[2]^2),
main="Linear Mixed Model: Scenario 1",
sub="%RelError(Within group variance)",ylim=c(-125,230))
abline(h=0)
```

D.2.2 Simulation of Estimated Lambda Case

```
# Set up initial values and simulation parameters

# Number of blocks
Q <- 3
R <- Q

# Number of population in each block
M <- c(130,120,100)
n <- sum(M)
T <- 4
N <- n*T
Block <- NULL
for (q in 1:Q){
    Block <- c(Block,rep(q,M[q]*T))
}

# Set up Grouping variable and time
Group <- c(rep(1:M[1],T),rep((M[1]+1):(M[1]+M[2]),T),
        rep((M[1]+M[2]+1):n,T))
G <- length(unique(Group))
Time <- NULL
m <- 25
for (q in 1:Q){
    Time <- c(Time,c(rep(1:T,each=M[q])))
}
```

```
Gr <- NULL
for (q in 1:Q){
    Gr <- c(Gr,c(rep(1:M[q],T)))
}

# Actual value of beta and lambda (yij=2+4xij+uj+eij)
a <- 2
b <- 4
Truebeta <- c(a,b)
p <- length(Truebeta)
Truelambda <- matrix(c(rep(1,3),rep(c(0.95,0.9,0.85),3)),nrow=3)

#Initialising
norow <- 800
BMat <- matrix(0,nrow=norow,ncol=10)
VMat <- matrix(0,nrow=norow,ncol=10)
CMat <- matrix(0,nrow=norow,ncol=10)
SMat <- matrix(0,nrow=norow,ncol=10)
SVMat <- matrix(0,nrow=norow,ncol=10)
KMat <- matrix(0,nrow=norow,ncol=10)
LMat <- matrix(0,nrow=norow,ncol=Q*T)
FMat <- matrix(0,nrow=norow,ncol=20)

for (iter in 1:norow) {

# Generating population values
    X2 <- NULL
    X2t1 <- runif(n)
    X2t2 <- X2t1+1
    X2t3 <- X2t2+1
    X2t4 <- X2t3+1
    Bl <- NULL
    for (q in 1:Q){
        Bl <- c(Bl,c(rep(q,M[q])))
    }
    for (q in 1:Q){
        X2 <- c(X2,c(X2t1[Bl==q],X2t2[Bl==q],X2t3[Bl==q],X2t4[Bl==q]))
    }
    X <- cbind(rep(1,N),X2)
    sigma <- c(3,1)
    TrueVar <- sigma^2
    Yfixed <- X%*%matrix(Truebeta,ncol=1)+rnorm(N,0,sigma[2])
    Y <- NULL
    for (q in 1:Q){
        r <- rnorm(M[q],0,sigma[1])
        for (t in 1:T){
            for (g in 1:M[q]){
                Y <- c(Y,c(Yfixed[Block==q & Time==t & Gr==g]+r[g]))
```

```
            }
        }
    }

# Estimates of beta and variance under perfect linkage
    fit <- lme(Y ~ X[,2], random=~1|Group, method="REML")
    TrueB <- fit$coefficients$fixed
    Truesigmasq <- c(as.numeric(VarCorr(fit)[1,1]),
                     as.numeric(VarCorr(fit)[2,1]))

# Estimates of standard errors and confidence limits under perfect linkage
    TrueSE <- sqrt(diag(fit$varFix))
    z <- qnorm(0.975)
    TrueLB <- TrueB-z*TrueSE
    TrueUB <- TrueB+z*TrueSE
    TrueLbe.var <- intervals(fit)$reStruct$Group[1]
    TrueUbe.var <- intervals(fit)$reStruct$Group[3]
    TrueLwithin.var <- intervals(fit)$sigma[1]
    TrueUwithin.var <- intervals(fit)$sigma[3]
    TrueLvar <- c(TrueLbe.var,TrueLwithin.var)
    TrueUvar <- c(TrueUbe.var,TrueUwithin.var)
    selog <- sqrt(diag(fit$apVar))
    senolog <- exp(attr(fit$apVar,'Pars')) * selog
    SEsigmasqTrue <- 2 * exp(attr(fit$apVar,'Pars')) * senolog

# Storing perfect linkage results
    BMat[iter,1:2] <- TrueB
    SMat[iter,1:2] <- Truesigmasq
    VMat[iter,1:2] <- TrueSE
    CMat[iter,1:2] <- c((TrueLB<Truebeta)*(TrueUB>Truebeta))
    SVMat[iter,1:2] <- c(SEsigmasqTrue)
    KMat[iter,1:2] <- c((TrueLvar<sigma)*(TrueUvar>sigma))
    FMat[iter,1:2] <- c(TrueLbe.var^2,TrueUbe.var^2)
    FMat[iter,3:4] <- c(TrueLwithin.var^2,TrueUwithin.var^2)

# Generating actual linked Y and X values

# Generating xstar and ystar
    Ystar <- NULL
    X2star <- NULL
    Aq.l <- list()
    lambdaHat <- matrix(1,nrow=Q,ncol=T)
    for (q in 1:Q){
        for (t in 1:T){
            Atq <- Afun(N=M[q],lambda=Truelambda[q,t])
            Ystar <- c(Ystar,c(Atq%*%matrix(Y[Block==q & Time==t])))
            X2star <- c(X2star,c(Atq%*%matrix(X2[Block==q & Time==t])))
            if(t>=2) Aq.l[[t]] <- Atq
```

```
        }
        Aq <- bdiag(Aq.l[2:T])
        lambdaq <- mean(sample(diag(Aq),size=m))
        lambdaHat[q,2:T] <- min((m-0.5)/m,max(1/M[q]*T,lambdaq))
    }
    LMat[iter,] <- c(lambdaHat)
    Xstar <- cbind(rep(1,N),X2star)
    Estlambda <- lambdaHat
    f4 <- lme(Ystar ~ Xstar[,2], random=~1|Group, method="REML")
    b_Naive <- f4$coefficients$fixed
    var.u <- as.numeric(VarCorr(f4)[1,1])
    var.e <- as.numeric(VarCorr(f4)[2,1])
    Naivesigmasq <- c(var.u,var.e)

# Estimates of standard errors and confidence limits under imperfect
linkage
    SEb_Naive <- sqrt(diag(f4$varFix))
    z <- qnorm(0.975)
    NaiveLB <- b_Naive-z*SEb_Naive
    NaiveUB <- b_Naive+z*SEb_Naive
    NaiveLbe.var <- intervals(f4)$reStruct$Group[1]
    NaiveUbe.var <- intervals(f4)$reStruct$Group[3]
    NaiveLwithin.var <- intervals(f4)$sigma[1]
    NaiveUwithin.var <- intervals(f4)$sigma[3]
    NaiveLvar <- c(NaiveLbe.var,NaiveLwithin.var)
    NaiveUvar <- c(NaiveUbe.var,NaiveUwithin.var)
    selogNaive <- sqrt(diag(f4$apVar))
    senologNaive <- exp(attr(f4$apVar,'Pars')) * selogNaive
    SEsigmasqNaive <- 2 * exp(attr(fit$apVar,'Pars')) * senologNaive

# Storing imperfect linkage results
    BMat[iter,3:4] <- b_Naive
    SMat[iter,3:4] <- c(var.u,var.e)
    VMat[iter,3:4] <- SEb_Naive
    CMat[iter,3:4] <- c((NaiveLB<Truebeta)*(NaiveUB>Truebeta))
    SVMat[iter,3:4] <- c(SEsigmasqNaive)
    KMat[iter,3:4] <- c((NaiveLvar<sigma)*(NaiveUvar>sigma))
    FMat[iter,5:6] <- c(NaiveLbe.var^2,NaiveUbe.var^2)
    FMat[iter,7:8] <- c(NaiveLwithin.var^2,NaiveUwithin.var^2)

# Calculation and storage of b, b_mle and b_REML estimates plus
associated standard errors and confidence limits given actual values
of lambda

    temp <- LinEstPlus(lambda=Truelambda,var.u=var.u,var.e=var.e,X=X,
            Xstar=Xstar,Ystar=Ystar,Block=Block,Group=Group,Time=Time)
    BMat[iter,5:6] <- c(temp$b)
    BMat[iter,7:8] <- c(temp$b_mle)
```

```
    BMat[iter,9:10] <- c(temp$b_REML)

    VMat[iter,5:6] <- c(temp$SEb)
    VMat[iter,7:8] <- c(temp$SEb_mle)
    VMat[iter,9:10] <- c(temp$SEb_REML)

    CMat[iter,5:6] <- c((temp$CLb[,1]<Truebeta)
                    *(temp$CLb[,2]>Truebeta))
    CMat[iter,7:8] <- c((temp$CLb_mle[,1]<Truebeta)
                    *(temp$CLb_mle[,2]>Truebeta))
    CMat[iter,9:10] <- c((temp$CLb_REML[,1]<Truebeta)
                    *(temp$CLb_REML[,2]>Truebeta))

    SMat[iter,5:6] <- c(temp$rb$sigmasq.u,temp$rb$sigmasq.e)
    SMat[iter,7:8] <- c(temp$sigmasqML[1],temp$sigmasqML[2])
    SMat[iter,9:10] <- c(temp$sigmasqREML[1],temp$sigmasqREML[2])

    SVMat[iter,5:6] <- c(temp$SEsigmasq)
    SVMat[iter,7:8] <- c(temp$sesigmasqML)
    SVMat[iter,9:10] <- c(temp$sesigmasqREML)

    KMat[iter,5:6] <- c((temp$CLVar[,1]<TrueVar)
                        *(temp$CLVar[,2]>TrueVar))
    KMat[iter,7:8] <- c((temp$CLVar_ML[,1]<TrueVar)
                        *(temp$CLVar_ML[,2]>TrueVar))
    KMat[iter,9:10] <- c((temp$CLVar_REML[,1]<TrueVar)
                        *(temp$CLVar_REML[,2]>TrueVar))

    FMat[iter,9:10] <- c(temp$CLVar[1,])
    FMat[iter,11:12] <- c(temp$CLVar[2,])
    FMat[iter,13:14] <- c(temp$CLVar_ML[1,])
    FMat[iter,15:16] <- c(temp$CLVar_ML[2,])
    FMat[iter,17:18] <- c(temp$CLVar_REML[1,])
    FMat[iter,19:20] <- c(temp$CLVar_REML[2,])

    print(c(iter,SMat[iter,c(1,3,5,7,9)]))
    print(c(iter,FMat[iter,c(1,2,5,6,9,10,13,14,17,18)]))
    print(c(iter,SMat[iter,c(2,4,6,8,10)]))
    print(c(iter,FMat[iter,c(3,4,7,8,11,12,15,16,19,20)]))

# End simulation loop
}

# Storage of simulation results
NewSim1 <- list(Truebeta=Truebeta,Truelambda=Truelambda,BMat=BMat,
            VMat=VMat,CMat=CMat,SMat=SMat,SVMat=SVMat,KMat=KMat,LMat=LMat,
            FMat=FMat)
```

```
done <- 1:norow

# Simulation Performance

# % Relative bias on intercept (true,b_Naive,b,b_mle,b_REML)
round(100*apply(NewSim1$BMat[done,c(1,3,5,7,9)]-
NewSim1$Truebeta[1],2,mean)/NewSim1$Truebeta[1],digits=2)

# % Relative bias on slope (true,b_Naive,b,b_mle,b_REML)
round(100*apply(NewSim1$BMat[done,c(2,4,6,8,10)]-
NewSim1$Truebeta[2],2,mean)/NewSim1$Truebeta[2],digits=2)

# % Relative RMSE on intercept (true,b_Naive,b,b_mle,b_REML)
round(100*sqrt(apply((NewSim1$BMat[done,c(1,3,5,7,9)]-
NewSim1$Truebeta[1])^2,2,mean)/NewSim1$Truebeta[1]),digits=2)

# % Relative RMSE on slope (true,b_Naive,b,b_mle,b_REML)
round(100*sqrt(apply((NewSim1$BMat[done,c(2,4,6,8,10)]-
NewSim1$Truebeta[2])^2,2,mean)/NewSim1$Truebeta[2]),digits=2)

# % Relative bias on between-group variance
round(100*apply(NewSim1$SMat[done,c(1,3,5,7,9)]-
sigma[1]^2,2,mean)/sigma[1]^2,digits=2)

# % Relative bias on within-group variance
round(100*apply(NewSim1$SMat[done,c(2,4,6,8,10)]-
sigma[2]^2,2,mean)/sigma[2]^2,digits=2)

# % Relative RMSE on between group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(1,3,5,7,9)]-
sigma[1]^2)^2,2,mean)/sigma[1]^2),digits=2)

# % Relative RMSE on within group variance
round(100*sqrt(apply((NewSim1$SMat[done,c(2,4,6,8,10)]-
sigma[2]^2)^2,2,mean)/sigma[2]^2),digits=2)

# Actual % coverage of nominal 95% CI on intercept
round(100*apply(NewSim1$CMat[done,c(1,3,5,7,9)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on slope
round(100*apply(NewSim1$CMat[done,c(2,4,6,8,10)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on between group variance
round(100*apply(NewSim1$KMat[done,c(1,3,5,7,9)],2,mean),digits=1)

# Actual % coverage of nominal 95% CI on within group variance
round(100*apply(NewSim1$KMat[done,c(2,4,6,8,10)],2,mean),digits=1)
```

```
# Bias and RMSE on variance estimator of coefficient
Truese <- 0
Estse <- 0
bias <- 0
RMSE <- 0
for(i in 1:10){
    Truese[i] <- sqrt(sum((NewSim1$BMat[done,i]-
                mean(NewSim1$BMat[done,i]))^2)/norow)
    Estse[i] <- mean(NewSim1$VMat[done,i])
    bias[i] <- round(Estse[i]-Truese[i],digits=3)
    RMSE[i] <- round(sqrt(mean((NewSim1$VMat[done,i]-
                Truese[i])^2)),digits=3)
}
print(bias[c(1,3,5,7,9)])
print(bias[c(2,4,6,8,10)])
print(RMSE[c(1,3,5,7,9)])
print(RMSE[c(2,4,6,8,10)])

# Bias and RMSE on variance estimator of variance components
Truese2 <- 0
Estse2 <- 0
bias2 <- 0
RMSE2 <- 0
for(i in 1:10){
    Truese2[i] <- sqrt(sum((NewSim1$SMat[done,i]-
                mean(NewSim1$SMat[done,i]))^2)/norow)
    Estse2[i] <- mean(NewSim1$SVMat[done,i])
    bias2[i] <- round(Estse2[i]-Truese2[i],digits=3)
    RMSE2[i] <- round(sqrt(mean((NewSim1$SVMat[done,i]-
                Truese2[i])^2)),digits=3)
}
print(bias2[c(1,3,5,7,9)])
print(bias2[c(2,4,6,8,10)])
print(RMSE2[c(1,3,5,7,9)])
print(RMSE2[c(2,4,6,8,10)])

# Distribution of relative errors of intercept parameter estimates
par(mfrow = c(2, 1))
boxplot(list(
TR=100*(NewSim1$BMat[done,1]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
N=100*(NewSim1$BMat[done,3]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
AN=100*(NewSim1$BMat[done,5]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
ML=100*(NewSim1$BMat[done,7]-
NewSim1$Truebeta[1])/NewSim1$Truebeta[1],
RE=100*(NewSim1$BMat[done,9]-
```

```
NewSim1$Truebeta[1])/NewSim1$Truebeta[1]),
main="Linear Mixed Model: Scenario 2",sub="%RelError(Intercept)",
ylim=c(-40,40))
abline(h=0)

# Distribution of relative errors of slope parameter estimates
boxplot(list(
TR=100*(NewSim1$BMat[done,2]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
N=100*(NewSim1$BMat[done,4]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
AN=100*(NewSim1$BMat[done,6]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
ML=100*(NewSim1$BMat[done,8]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2],
RE=100*(NewSim1$BMat[done,10]-
NewSim1$Truebeta[2])/NewSim1$Truebeta[2]),
main="Linear Mixed Model: Scenario 2",sub="%RelError(Slope)",
ylim=c(-3,3))
abline(h=0)

# Distribution of relative errors of between-group variance parameter
estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,1]-
sigma[1]^2)/sigma[1]^2,
N=100*(NewSim1$SMat[done,3]-
sigma[1]^2)/sigma[1]^2,
AN=100*(NewSim1$SMat[done,5]-
sigma[1]^2)/sigma[1]^2,
ML=100*(NewSim1$SMat[done,7]-
sigma[1]^2)/sigma[1]^2,
RE=100*(NewSim1$SMat[done,9]-
sigma[1]^2)/sigma[1]^2),
main="Linear Mixed Model: Scenario 2",
sub="%RelError(Between group variance)",ylim=c(-60,60))
abline(h=0)

# Distribution of relative errors of within-group variance parameter
estimates

boxplot(list(
TR=100*(NewSim1$SMat[done,2]-
sigma[2]^2)/sigma[2]^2,
N=100*(NewSim1$SMat[done,4]-
sigma[2]^2)/sigma[2]^2,
AN=100*(NewSim1$SMat[done,6]-
```

```
sigma[2]^2)/sigma[2]^2,
ML=100*(NewSim1$SMat[done,8]-
sigma[2]^2)/sigma[2]^2,
RE=100*(NewSim1$SMat[done,10]-
sigma[2]^2)/sigma[2]^2),
main="Linear Mixed Model: Scenario 2",
sub="%RelError(Within group variance)",ylim=c(-125,230))
abline(h=0)
```

Bibliography

[1] Abowd, J. M. & Woodcock, S. D. (2004). Multiply-Imputing Confidential Characteristics and File Links in Longitudinal Linked Data. In Domingo-Ferrer, J. & Torra, V. (Eds.), *Privacy in Statistical Databases* (pp. 290-297). Springer-Verlag.

[2] Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis.* New York: John Wiley & Sons.

[3] Bachrach, L. K., Hastie, T., Wang, M., Narasimhan, B. & Marcus, R. (1999). Bone Mineral Acquisition in Healthy Asian, Hispanic, Black, and Caucasian Youth: A Longitudinal Study. *Journal of Clinical Endocrinology & Metabolism, 84*, 4702-4712.

[4] Biemer, P. P., Groves, R. M., Lyberg, L. E., Mathiowetz, N. A. & Sudman, S. (1991). *Measurement errors in surveys.* New York: John Wiley & Sons.

[5] Boardman, T. J. (1974). Confidence Intervals for Variance components – A Comparative Monte Carlo Study. *Biomatrics, 30*(2), 251-262.

[6] Casella, G. & Berger, R. L. (2002). *Statistical Inference.* Pacific Grove, CA: Duxbury.

[7] Chambers, R. (2009). Regression Analysis Of Probability-Linked Data. *Official Statistics Research Series, 4.*

[8] Chesher, A. (1991). The effect of measurement error. *Biometrika, 78*(3), 451-462.

[9] Chesher, A. & Nesheim, L. (2006). Review of literature on the statistical properties of linked datasets. DTI Occasional Paper 3: http://www.dti.gov.uk/files/file24832.pdf

[10] Chua, T. C. & Fuller, W. A. (1987). A Model for Multinomial Response Error Applied to Labor Flows. *Journal of the American Statistical Association, 82*, 46-51.

[11] Copas, J. R. & Hilton, F. J. (1990). Record Linkage: Statistical Models for Matching Computer Records. *Journal of the Royal Statistical Society, A, 153*, 287-320.

[12] Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, B, 39*(1), 1-38.

[13] Diggle, P. J., Liang, K. & Zeger, S. L. (1994). *Analysis of Longitudinal Data*. New York: Oxford University Press.

[14] Dobson, A. J. (2002). *An Introduction to Generalized Linear Models*. Boca Raton, Florida: Chapman & Hall/CRC.

[15] Dwyer, J. H., Feinleib, M., Lippert, P. & Hoffmeister, H. (1992). *Statistical Models for Longitudinal Studies of Health*. New York: Oxford University Press.

[16] Elias, P. (2003). Do Employment Zones reduce unemployment? An analysis based on administrative microdata, University of Warwick Institute for Employment Research.

[17] Felligi, I. P. & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association, 64*, 1183-1210.

[18] Ferrante A. M., Rosman D. L. & Knuiman, M. (1993). The construction of a road injury database. *Accident Analysis and Prevention, 25*, 659-665.

[19] Fitzmaurice, G. M., Laird, N. M. & Ware, J. H. (2004). *Applied Longitudinal Analysis*. New York: John Wiley & Sons.

[20] Freund, J. E. & Walpole, R. E. (1987). *Mathematical Statistics*. New Jersey: Prentice-Hall.

[21] Fuller, W. A. (1987). *Measurement error models*. New York: John Wiley & Sons.

[22] Gill, L., Goldacre, M., Simmons, H., Bettley, G. & Griffith, M. (1993). Computerised linking of medical record: methodological guidelines. *Journal of Epidemiology and Community Health, 47*, 316-319.

[23] Gilmour, R., Thompson, R. & Cullis, B. R. (1995). Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. *Biometrics, 51*(4), 1440-1450.

[24] Godambe, V. P. (1976). Conditional Likelihood and Unconditional Optimum Estimating Equations. *Biometrika, 63*(2), 277-284.

[25] Godambe, V. P. (1991). *Estimating functions*. New York: Oxford University Press.

[26] Groves, R. M. (1989). *Survey errors and survey costs*. New York: John Wiley & Sons.

[27] Harville, D. A. (1977). Maximum-likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association, 72*(358), 320-340.

[28] Harville, D. A. (1997). *Matrix Algebra From A Statistician's Perspective*. New York: Springer-Verlag.

[29] Herzog, T. H., Scheuren, F. & Winkler, W. E. (2010). Record linkage. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(5), 535-543.

[30] Heyde, C. C. (1997). *Quasi-Likelihood And Its Application: A General Approach to Optimal Parameter Estimation*. New York: Springer-Verlag.

[31] Hogg, R. V. & Tanis, E. A. (1993). *Probability and Statistical Inference*. New York: Macmillan.

[32] Holman, C. D. J. (n.d.) The Analysis of Linked Health Data: Principles and Hands-On Applications. *A Short-Course with Professor C. D'Arcy J. Holman*.

[33] Holman, C. D. J., Bass, A. J., Rouse, I. L. & Hobbs, M. S. T. (1999). Population-based linkage of health records in Western Australia: development of a health services research linked database. *Australian and New Zealand Journal of Public Health, 23*(5), 453-459.

[34] Iron, K. S., Manuel, D. G. & Williams, J. (2003). Using a linked data set to determine the factors associated with utilization and costs of family physician services in Ontario: effects of seld-reported chronic conditions. *Chronic Diseases in Canada, 24*(4).

[35] Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association, 84*(406), 414-420.

[36] Kelly, N. (2003). Repairing EMS Employer Longitudinal Links. Retrieved from Statistics New Zealand website: www.stats.govt.nz/browse_for_stats/income-and-work/employment_and_unemployment/kelly-repairing-ems-links.aspx

[37] Kutner, M. H., Nachtsheim, C. J., Neter, J. & Li, W. (2005). *Applied Linear Statistical Models*. New York: McGraw-Hill/Irwin.

[38] Lahiri, P. & Larsen, M. D. (2005). Regression Analysis With Linked Data. *Journal of the American Statistical Association, 100*(469), 222-230.

[39] Lyberg, L., Biemer, P., Collins, M., Leeuw, E., Dippo, C., Schwarz, N. & Trewin, D. (1997). *Survey Measurement and Process Quality*. New York: John Wiley & Sons.

[40] Machado, C. J. (2004). A literature review of record linkage procedures focusing on infant health outcomes. *Cad. Saúde Pública, Rio de Janeiro, 20*(2), 362-371.

[41] Mackie, C. & Bradburn, N. (2000). *Improving Access to and Confidentiality of Research Data: Report of a Workshop*. Commitee on National Statistics (CNSTAT). Commission on Behavioral and Social Sciences and Education. Washington DC: National Academies Press.

[42] McCallum, J., Lonergan, J. & Raymond, C. (1993). *The NCEPH record linkage pilot study: a preliminary examination of individual Insurance Commission records with linked data sets*. Canberra: National Centre for Epidemiology and Public Health.

[43] McCullagh, P. & Tibshirani, R. (1990). A Simple Method for the Adjustment of Profile Likelihoods. *Journal of the Royal Statistical Society. Series B (Methodological), 52*(2), 325-344.

[44] McCulloch, C. E. & Searle, S. R. (2001). *Generalized, Linear, and Mixed Models*. New York: John Wiley & Sons.

[45] Moore, D. S. & McCabe, G. P. (2004). *Introduction to the practice of Statistics*. New York: W.H. Freeman and Company.

[46] National Statistical Service. (2011). *Australian Longitudinal Learning Database (ALLD)* (Concept Paper). Retrieved from http://www.nss.gov.au/nss/home.nsf/NSS/4FB1EC5C8DF5709BCA25784C000386C7?opendocument

[47] Neter, J. (1970). Measurement Errors in Reports of Consumer Expenditures. *Journal of Marketing Research, VII*, 11-25.

[48] Neter, J., Maynes, E. S. & Ramanathan, R. (1965). The effect of mismatching on the measurement of response error. *Journal of the American Statistical Association, 60*(312), 1005-1027.

[49] Newcombe, H. B. & Kennedy, J. M. (1962). Record linkage: Making maximum use of the discriminating power of identifying information. *Communications of the Association for Computing Machinary, 5*, 563-567.

[50] Newcombe, H. B., Kennedy, J. M., Axford, S. J. & James, A. P. (1959). Automatic linkage of vital records. *Science, 130*(3381), 954-959.

[51] Noble, M., Smith, G. A. N. & Cheung, S. Y. (1998). Origins and Destinations - Social Security Claimant Dynamics. *Journal of Social Policy, 27*, 351-369.

[52] Palca, J. (1990). Getting to the Heart of the Cholesterol Debate. *Science, 247*, 1170-1171.

[53] Peytchev, A., Peytcheva, E. & Groves, R. M. (2010). Measurement error, unit nonresponse, and self-reports of abortion experiences. *Public Opinion Quarterly, 74*(2), 319-327.

[54] Pinheiro, J. C. & Bates, D. M. (2004). *Mixed-Effects Models in S and S-Plus*. New York: Springer Science +Business Media.

[55] Rao, C. R. (2002). *Linear Statistical Inference and Its Applications*. New York: John Wiley & Sons.

[56] Rao, P. S. R. S. (1997). *Variance Components Estimation: Mixed models, methodologies and applications*. London: Chapman & Hall.

[57] Ruppert, D., Wand, M. P. & Carroll, R. J. (2003). *Semiparametric Regression*. New York: Cambridge University Press.

[58] Scheuren, F. & Winkler, W. E. (1993). Regression analysis of data files that are computer matched. *Survey Methodology, 19*, 39-58.

[59] Scheuren, F. & Winkler, W. E. (1997). Regression analysis of data files that are computer matched, II. *Survey Methodology, 23*, 157-165.

[60] Searle, S. R., Casella, G. & McCulloch, C. E. (2006). *Variance Components*. New York: John Wiley & Sons.

[61] Sibthorpe, B., Kliewer, E. & Smith, L. (1995). Record linkage in Australian epidemiological research: health benefits, privacy safeguards and future potential. *Australian Journal of Public Health, 19*, 250-256.

[62] Smith, M. E. & Newcombe, H. B. (1975). Methods for Computer Linkage of Hospital Admission-Separation Records into Cumulative Health Histories. *Methods of Information in Medicine, 14*(3), 118-125.

[63] Stanley, F. J., Croft, M., Gibbins, J. & Read, A. W. (1994). A population data base for maternal and child health research in Western Australia using record linkage. *Paediatric and Perinatal Epidemiology, 8*, 433-447.

[64] Tepping, B. J. (1968). A model for optimum linkage of records. *Journal of the American Statistical Association, 63*, 1321-1332.

[65] Trottini, M. (2005). *Statistical Confidentiality in Longitudinal Linked Data: Objectives and Attributes*. Paper presented at Joint UNECE/Eurostat work session on statistical data confidentiality, Geneva, Switzerland.

[66] Verbeke, G. & Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. New York: Springer-Verlag.

[67] Wackerly, D. D., Mendenhall III, W. & Scheaffer, R. L. (2008). *Mathematical Statistics with Applications*. Belmont, CA: Duxbury.

[68] Wand, M. P. & Ormerod, J. T. (2008). On Semiparametric Regression with O'Sullivan Penalized Splines. *Australian and New Zealand Journal of Statistics, 50*(2), 179-198.

[69] Weiss, R. E. (2005). *Modeling Longitudinal Data*. New York: Springer Science +Business Media.

[70] Winkler, W. E. (1988). Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. *Proceedings of the Conference on Uncertainty in Artificial Intelligence* (pp. 667-671). Alexandria, VA: American Statistical Association.

[71] Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. *Proceedings of the Section on Survey Research Methods* (pp. 354-359). Alexandria, VA: American Statistical Association.

[72] Winkler, W. E. (1993). Improve decision rules in the Fellegi-Sunter model of record linkage. *Proceedings of the Section on Survey Research Methods* (pp. 274-279). Alexandria, VA: American Statistical Association.