MISSING RESPONSES IN GENERALIZED LINEAR MIXED MODELS WHERE THE MISSINGNESS IS NONIGNORABLE

by

Ali Daher

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfilment of the requirements for the degree of

Master of Science

in

Probability and Statistics

Carleton University

Ottawa, Ontario

Abstract

In this thesis, we go through some practical issues of binary and Poisson regression models, which are particular cases of generalized linear models (GLMs) and are very useful for analyzing real datasets. We review methods for finding the maximum likelihood estimators in GLMs. The treatment of such methods provides a rigid foundation of GLMs.

This thesis also reviews generalized linear mixed models (GLMMs) and methods to find the maximum likelihood estimators of both fixed and random effects parameters, where GLMMs are of increasing importance to many practitioners. GLMMs are widely used in clustered and longitudinal data analyses, where random effects are used to model subject or cluster specific effects.

We study the influence of nonignorable missing observations on a binary mixed model, where the missingness occurs because of the treatment itself or certain drugs effects. Although highly efficient, the traditional maximum likelihood (ML) method becomes complex when the number of missing responses increases, requiring intensive computation. We review algorithms for finding the ML estimators in GLMMs with nonignorable missing responses.

We conduct a simulation study to assess the performance of the ML method in the presence of the nonignorable missing data in the response variable, where there is no missingness in the covariates. The simulation results indicate that when the percentage of missingness

is high, the empirical coverage probabilities of the parameter estimates are a bit apart from the nominal 95% level. But they tend to get closer to the nominal level when the number of missing responses decreases. Also, the average lengths of the confidence intervals for the regression parameters tend to be smaller for larger sample size, as expected. Under misspecified missing data models, we observe systemic bias in the regression estimators and also poor coverage probabilities from the confidence intervals.

We conclude that when analyzing incomplete data with missing responses, it is necessary to incorporate a suitable missing data model into the observed data likelihood function in order to obtain unbiased and efficient estimators of the model parameters. We also note that a misspecified missing data model can provide systematic bias in the maximum likelihood estimation. So it is important to assess the validity of a missing data model when performing a likelihood inference based on the given observed data.

Acknowledgements

I thank Dr. Sanjoy K. Sinha for his supervision, encouragement and support throughout the thesis work. He had guided me step by step through the whole content of my research. I appreciate all his encouragement, productive ideas, and support throughout this work. Completion of this work would have been impossible without his guidance. He was very patient, and he encouraged me in a professional way to defeat my fears until I reached the completion of this work. It has been an honour to get a chance of being his M.Sc. student.

I also thank many people in the School of Mathematics and Statistics at Carleton University for their help, continual support to continue my graduate study in Statistics. I would like to thank Nicole Gaertner, the Graduate Administrator of the school, and Kevin Crosby, the School Administrator for their valuable help and suggestions.

It is an honour to thank my wonderful wife Amal El harakeh who provided me strengths and support with unconditional love and care. My children Lana, Tala and Hussein motivate and support me with their smiles to face this challenge, and without my lovely family, I would have never made so far.

Contents

List of Abbreviation

CI	Confidence Interval
GLM	Generalized Linear Model
GLMM	Generalized Linear Mixed Model
IRLS	Iteratively Reweighted Least Squares
LM	Linear Model
MAR	Missing at Random
MCAR	Missing Completely at Random
MLE	Maximum Likelihood Estimator
MSE	Mean Squared Error
NI	Non-Ignorable
NMAR	Not Missing at Random
NR	Newton-Raphson
i.e	That is
MCEM	Monte Carlo EM
MCNR	Monte Carlo Newton-Raphson

RML	Robust maximum likelihood
w.r.t	With Respect To

List of Tables

Chapter 1

Introduction

1.1 Missing Responses in Generalized Linear Mixed Models

Missing responses are common occurrence for longitudinal studies because of treatment drop-out, mistimed measurements, and subjects being too sick to come to the clinic to be measured. A response can be missing at one follow-up time and then be measured at the next follow-up time, resulting in arbitrary non monotone missingness patterns. Missing values in the data could be either ignorable or nonignorable (Ibrahim, Lipsitz and Chen, 1999). An individual's unobserved response is called nonignorable if it is related to the missing values of the response variable (Little, 1982). If there are nonignorable missing responses, the distribution of the missing data needs to be correctly modelled to avoid any biases that may be resulted from the incomplete data analysis (Baker and Laird, 1988).

In this thesis, we concentrate on methods for estimating parameters in the generalized linear mixed model (GLMM) with nonignorable and nonmontone patterns of missing data in the response variable, where the reason of missingness often depends on the missing values themselves.

Our goal is to find a suitable method for estimating parameters in the generalized linear mixed model with nonignorable and nonmontone missing responses commonly encountered in longitudinal data analysis. Longitudinal studies consist of taking measurements repeatedly from an individual by following the participants at different times.

1.2 Missing Data Problem

As a practical example, consider the CARDIA (Coronary Artery Development in Young Adults) study (Hughes et al.1987) conducted in four urban centers (Birmingham, AL; Chicago, IL; Minneapolis, MN; and Oakland, CA) across the United States, where a total of 5,078 young adults of ages 18-30 years were followed prospectively and examined up to five times from 1986 to 1996. The goal of the statistical analysis is to draw inferences on the change in smoking prevalence by race/sex group in the presence of missing data.

Little and Rubin (1987) discuss missing data patterns in three different ways, which suggest the relationship between the missingness and observed values in the data. Suppose Y is the complete data that would occur in the absence of missing values. We can write $Y = (Y_{\text{obs}}, Y_{\text{mis}})$, where Y_{obs} denotes the observed values and Y_{mis} the missing values of Y. The density function of the joint distribution of Y_{obs} and Y_{mis} can be written as

$$f(Y|\theta) = f(Y_{\text{obs}}, Y_{\text{mis}}|\theta),$$

depending on some vector of unknown parameters θ.

Consider a binary variable that indicates whether each component of Y is observed or miss-

ing. A vector of response indicators $R = (R_{ij})$ is defined in such a way that

$$R_{ij} = \begin{cases} 1 & \text{if } y_{ij} \text{ is observed} \\ 0 & \text{if } y_{ij} \text{ is missing} \end{cases}.$$

Here R is treated as a random vector in the model and we assume a joint distribution of R and Y, which can be expressed as

$$f(Y, R|\theta, \phi) = f(Y|\theta) f(R|Y, \phi),$$

where $f(R|Y, \phi)$ depends on some parameters ϕ, which is the conditional distribution of R given Y for the missing data mechanism.

Data are considered to be missing completely at random (MCAR) when the missing data process does not depend on the observed values as well as the missing values of the data, which means that the missingness is independent of Y. It can be expressed as

$$f(R|Y_{\text{obs}}, Y_{\text{mis}}, \phi) = f(R|\phi).$$

The missing data mechanism is considered to be missing at random (MAR) when the missingness depends only on the observed values of the variables in the data, which means that the missingness only depends on observed components Y_{obs} of Y. It can be expressed as

$$f(R|Y_{\text{obs}}, Y_{\text{mis}}, \phi) = f(R|Y_{\text{obs}}, \phi).$$

An individual's non-response is considered to be not missing at random (NMAR) when the individual's response probability depends on the response variable itself, which means that the missingness depends on the unobserved values. It can be defined as

$$f(Y, R|X, \theta, \phi) = f(Y|X, \theta)f(R|Y, X, \phi).$$

Here X represents a full set of covariates. The first component classifies the distribution of Y given X in the population and the second component models the occurrence of response as a function of Y and X.

Rubin (1976) points out that when MAR occurs, the likelihood based inference doesn't depend on missing data mechanism. Baker and Laird (1988) point out that if missingness depends on individual's unobserved response, then it is considered to be nonignorable missing data. In such cases, the distribution needs to be modelled for the missing data mechanism. Several examples that describe the likelihood-based inference using the maximum likelihood and missing data mechanisms are discussed in Little (1995).

The likelihood function under certain modelling assumptions is mostly used as an estimation method for missing data. For the ML estimation, the EM algorithm is a well-known iterative algorithm. The M step of the expectation-maximization (EM) creates a function for the ML estimation by treating data as complete. Given the current estimates of the parameters, the conditional expectation of the missing data given the observed data is performed at the E step.

1.3 Statement of the Problems

In this thesis, we study the maximum likelihood method to make inferences for generalized linear mixed models for incomplete binary longitudinal data. Specifically, we generalize the

theories of the maximum likelihood method to make simultaneous inferences for generalized linear mixed models with missing responses occurring from a nonignorable missing data mechanism.

Under a given missing data mechanism, we study the finite sample properties of the estimators based on empirical biases and mean squared errors as well as coverage probabilities (CPs) and average lengths of the confidence intervals for regression and association parameters of GLMMs.

We also investigate the impact of various missingness patterns on the ML estimators in terms of biases and mean squared errors as well as coverage probabilities (CPs) and average lengths of the confidence intervals. Our interest also lies in studying the empirical properties of the parameter estimates in the generalized linear mixed model under both correctly specified and misspecified models.

1.4 Organization of the Thesis

The thesis is organized as follows. In Chapter 2, we introduce the generalized linear model (GLMs). In Chapter 3, we expand the notion of GLMs to GLMMs by incorporating random effects in the model. We then present a simulation study based on GLMMs for different sample sizes. In Chapter 4, we review the literature on missing responses for analyzing incomplete data and describe methods for estimating the parameters of GLMMs with nonignorable missing response data. Also, we present a simulation study and discuss the results of fitting GLMMs in the case of nonignorable missing data in the response variable. Chapter 5 concludes the thesis with some discussion and directions for future research.

Chapter 2

Generalized Linear Model(GLM)

2.1 Introduction

In this chapter, we present the notion of fitting GLMs to binary and count data. In Section 2.2, we discuss GLMs in the exponential form. In Section 2.3, we present the structure and link function of GLMs. In Section 2.4, we show the equation of the likelihood of canonical and non-canonical cases of GLMs. In Section 2.5, we describe in details the IRLS method to find the MLEs of regression parameters in GLMs. In Section 2.6, we describe the quasi-likelihood method for estimating the model parameters in GLMs. In Section 2.7, we analyze some real data sets using GLMs.

2.2 Exponential Family of Distribution

The GLM is an extension of the linear model $E(y_i) = x_i^t\beta$, where $x_i = (1, x_{i1}, \ldots . x_{ip})^t$ is the linear predictor and y_i is a response variable. The linear model consists of the following components:

Random Component: Components of the response vector $y = (y_1, \ldots . y_n)^t$ are independent and normally distributed.

The notion of the GLM has two important aspects as described below.

i) The distribution of the response.

ii) The model that relates the mean response to the regression variables.

These two issues are not independent of each other because certain types of models are more appropriate for some distributions than for others. Building a generalized linear model involves three decisions:

i) What is the distribution of the data?

ii) What function of the mean will be modeled as linear in the predictors ?

iii) What will be the predictors?

The strategy of the GLM is to apply a link function to the mean of the response and fit the resulting model by the method of maximum likelihood. An important unifying concept underlying the GLM is the exponential family of distributions. Members of the exponential family of distribution all have probability density functions for an observed response y that can be expressed in the form

$$f(y,\theta,\phi) = \exp\left[\frac{(y\theta - b(\theta))}{a(\phi)} + c(y,\phi)\right], \qquad (2.1)$$

where $a(.)$, $b(.)$ and $c(.)$ are some specific functions, θ is a natural location, ϕ is a dispersion parameter and $a(\phi)$=$\phi.w$, where w is a known constant. The binomial, Poisson, and normal distributions are members of this exponential family.

Example: The probability density function of a normal random variable y with parameters μ and σ is given by

$$f(y,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-(y-\mu)^2}{2\sigma^2}\right]$$
$$= \exp\left[\left(\frac{y\mu - \frac{\mu^2}{2}}{\sigma^2}\right) - \left(\frac{1}{2}\right)\left(\frac{y^2}{\sigma^2} + \ln(2\pi\sigma^2)\right)\right]. \qquad (2.2)$$

The density function is of the form (2.2), with

$$\theta = \mu,$$

$$b(\theta) = \frac{\mu^2}{2},$$

$$a(\phi) = \phi,$$

$$\phi = \sigma^2,$$

and

$$c(y,\phi) = -\left(\frac{1}{2}\right)\left[\frac{y^2}{\sigma^2} + \ln(2\pi\sigma^2)\right].$$

2.3 Formal Structure and the Link Function

Consider n independent observations, $(y_1, \ldots y_n)$, with corresponding means $(\mu_1, \ldots \mu_n)$. Assume that the i^{th} observation y_i has a distribution that is a member of the exponential family. The systematic portion of the model involves predictor variables $(x_1, \cdots x_p)$. The model is constructed around the linear predictor $x_i = (1, x_{i1}, \ldots . x_{ip})^t$ given by

$$\eta_i = {x_i}^t\beta = \beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}. \qquad (2.3)$$

The involvement of this linear predictor suggests the terminology generalized linear models. The model is found through the use of the link function $\eta_i = g(\mu_i),\ \ (i = 1, \ldots n)$. The term link is derived from the fact that the function is the link between the mean and the linear predictor. The expected response is

$$E(y_i) = g^{-1}(\eta_i) = g^{-1}(x_i^t \beta). \qquad (2.4)$$

The link function is a monotonic differentiable function and the variance σ_i^2, $(i = 1, \ldots n)$, is a function of the mean μ_i.

Link Function: It is a transformation on the population means, not the data, and it takes advantage of the natural distribution of the response. Not using an appropriate transformation can result in problems with a fitted linear model, and also can result in significant problems with a generalized model. There are many possible choices of the link function. If we choose $\eta_i = \theta_i$, then we say that η_i is the canonical link.

Table 2.1 Shows the canonical links for the most common choices of distributions employed with generalized linear model.

Table 2.1: Canonical links for the generalized linear model

Distribution	Canonical Link
Normal	$\eta_i = \mu_i$ (identity link)
Binomial	$\eta_i = \ln(\frac{\pi_i}{1-\pi_i})$(logistic link)
Poisson	$\eta_i = \ln(\mu_i)$ (log link)
Exponential	$\eta_i = \frac{1}{\mu_i}$(reciprocal link)
Gamma	$\eta_i = \frac{1}{\mu_i}$(reciprocal link)

2.4 Likelihood Equations for GLM

2.4.1 Non-Canonical Case $\eta_i \neq \theta_i$

The log-likelihood function is

$$l = \log L(\beta; y) = \sum_{i=1}^{n} \left[\frac{(y_i\theta_i - b(\theta_i))}{a(\phi)} + c(y_i, \phi) \right]. \tag{2.5}$$

The derivative of the log-likelihood with respect to β may be obtained as

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} \frac{\partial l}{\partial \theta_i} \frac{\partial \theta_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_i}. \tag{2.6}$$

The partial derivative of l with respect to θ_i is given by

$$\frac{\partial l}{\partial \theta_i} = \sum_{i=1}^{n} \frac{1}{a(\phi)} \left[y_i - \frac{\partial b(\theta_i)}{\partial \theta_i} \right]. \tag{2.7}$$

We can show that

$$E\left[\frac{\partial \log\ f(y_i)}{\partial \theta_i} \right] = 0. \tag{2.8}$$

Substituting Eq.(2.7) into Eq.(2.8), we have

$$\frac{\partial b(\theta_i)}{\partial \theta_i} = \mu_i.$$

Then Eq.(2.7) can be written in the form

$$\frac{\partial l}{\partial \theta_i} = \frac{1}{a(\phi)} \sum_{i=1}^{n} (y_i - \mu_i). \tag{2.9}$$

The partial derivative of the linear predictor with respect to β is given by

$$\frac{\partial \eta_i}{\partial \beta} = x_i. \tag{2.10}$$

Substituting the Eq.(2.9) and (2.10) into Eq.(2.6), we have

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi)} \frac{\partial \theta_i}{\partial \eta_i} x_i. \qquad (2.11)$$

The score equations are determined by $\frac{\partial l}{\partial \beta} = 0$, which can be written in the matrix form as

$$X^t \Delta (y - \mu) = 0, \qquad (2.12)$$

where

$$\Delta = \text{diag}\left\{ \frac{\partial \theta_i}{\partial \eta_i} \right\}.$$

2.4.2 Canonical Case $\eta_i = \theta_i$

For simplicity consider $a(\phi) = 1$. Then Eq.(2.6) has the form

$$\begin{aligned} \frac{\partial l}{\partial \beta} &= \frac{\partial l}{\partial \theta_i} \frac{\partial \theta_i}{\partial \beta}, \\ &= \sum_{i=1}^{n} \left[y_i - \frac{\partial b(\theta_i)}{\partial \theta_i} \right] x_i, \\ &= \sum_{i=1}^{n} (y_i - \mu_i) x_i. \qquad (2.13) \end{aligned}$$

To find the MLE of β we need to solve the equation $\sum_{i=1}^{n}(y_i - \mu_i)x_i = 0$. The matrix form of these equations is:

$$X^t(y - \mu) = 0. \qquad (2.14)$$

These are called the **maximum likelihood score equations** for $\mu^t = (\mu_1, \ldots \mu_n)$.

2.5 Iteratively Reweighted Least Squares (IRLS) Method

To solve the ML score equations, we can use the iteratively reweighted least squares (IRLS) algorithm. We know that Eq.(2.6) has the form

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} \frac{\partial l}{\partial \theta_i}\frac{\partial \theta_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta}.$$

As θ_i is a function of μ_i, and μ_i is a function of η_i, we can write

$$\frac{\partial \theta_i}{\partial \eta_i} = \frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}, \qquad (2.15)$$

$$\frac{\partial \theta_i}{\partial \mu_i} = \frac{1}{v(\mu_i)}, \qquad (2.16)$$

where $v(\mu_i) = \dfrac{\partial^2 b(\theta_i)}{\partial \theta_i^2}$ is a variance function.

Substituting Eq.(2.15) and (2.16) into Eq.(2.6), we have

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi)}\frac{1}{v(\mu_i)}\frac{\partial \mu_i}{\partial \eta_i}x_i. \qquad (2.17)$$

Eq.(2.17) can be written in the form

$$\frac{\partial l}{\partial \beta} = \frac{1}{a(\phi)} \sum_{i=1}^{n} \frac{y_i - \mu_i}{v(\mu_i) g_\mu(\mu_i)} x_i$$

$$= \frac{1}{a(\phi)} \sum_{i=1}^{n} (y_i - \mu_i) w_i g_\mu(\mu_i) x_i, \tag{2.18}$$

where $g_\mu(\mu_i) = \dfrac{\partial g(\mu_i)}{\partial \mu_i}$ and $w_i = \dfrac{1}{v(\mu_i) g_\mu^2(\mu_i)}$.

We can write Eq.(2.18) in matrix form as

$$\frac{\partial l}{\partial \beta} = \frac{1}{a(\phi)} X^t W \Delta^* (y - \mu), \tag{2.19}$$

where X is the design matrix, and W and Δ^* are diagonal matrices defined by

$$W = \text{diag}\,\{w_1, \ldots, w_n\},$$

and

$$\Delta^* = \text{diag}\,\{g_\mu(\mu_1), \ldots, g_\mu(\mu_n)\}.$$

The ML equations for β are then given by

$$X^t W \Delta^* y = X^t W \Delta^* \mu, \tag{2.20}$$

where W, Δ^* and μ are functions of the unknown parameter β. These are typically nonlinear functions of the regression parameter β and for this reason the above equations cannot be solved explicity.

Solutions of the ML estimating equations (2.20) for β are often obtained by the IRLS method. This can be derived from the use of the Newton-Raphson method with Fisher scoring:

$$\beta^{(k+1)} = \beta^{(k)} + I(\beta^{(k)})^{-1} \frac{\partial l}{\partial \beta}\bigg|_{\beta=\beta^{(k)}}, \tag{2.21}$$

where (k) represents the k^{th} iteration, and $I(\beta)$ is the information matrix

$$I(\beta) = -E\left(\frac{\partial^2 l}{\partial\beta\partial\beta^t}\right)$$

$$= \frac{1}{a(\phi)} X^t W \Delta^* \frac{\partial \mu}{\partial \beta^t} + 0$$

$$= \frac{1}{a(\phi)} X^t W \Delta^* \Delta^{*-1} X$$

$$= \frac{1}{a(\phi)} X^t W X.$$

Then Eq.(2.21) takes the form

$$\beta^{(k+1)} = \beta^{(k)} + (X^t W X)^{-1} X^t W \Delta^* (y - \mu), \tag{2.22}$$

where W, Δ^* and μ are evaluated at $\beta^{(k)}$.

Eq.(2.22) is related to the IRLS method, which uses a pesudo vector

$$z = X\beta + \Delta^*(y - \mu), \tag{2.23}$$

for which we have

$$\text{var}(z) = \text{var}[\Delta^*(y - \mu)]$$

$$= \Delta^* \text{var}(y) \Delta^*$$

$$= \text{diag}\{a(\phi)v(\mu_i)g_\mu^2(\mu_i);\ \ i = 1, \ldots, n\}$$

$$= a(\phi)W^{-1}. \qquad (2.24)$$

So a weighted regression of the pesudo vector z on X with weights equal to the inverse of $\text{var}(z)$ gives

$$\begin{aligned}\beta^{(k+1)} &= (X^tWX)^{-1}X^tWz \\ &= (X^tWX)^{-1}X^tW[X\beta^{(k)} + \Delta^*(y-\mu)] \\ &= \beta^{(k)} + (X^tWX)^{-1}X^tW\Delta^*(y-\mu), \qquad (2.25)\end{aligned}$$

which is exactly the same as Eq.(2.22).

The variance of the ML estimator $\hat{\beta}$ may be obtained as

$$\begin{aligned}\text{var}(\hat{\beta}) &= I^{-1}(\beta) \\ &= a(\phi)(X^tWX)^{-1}. \qquad (2.26)\end{aligned}$$

2.5.1 Example: Logistic regression

In the case of the binary logistic regression model, we consider

$$\eta_i = g(\mu_i) = x_i^t\beta,$$

with

$$\pi_i = \frac{\exp(x_i^t\beta)}{1+\exp(x_i^t\beta)}, \quad v(\mu_i) = \pi_i(1-\pi_i).$$

Here we have

$$g_\mu(\mu_i) = \{\pi_i(1-\pi_i)\}^{-1}$$

$$\begin{aligned} w_i &= \{v(\mu_i)g_\mu^2(\mu_i)\}^{-1} \\ &= \left\{\frac{\pi_i(1-\pi_i)}{(\pi_i(1-\pi_i))^2}\right\}^{-1} \\ &= \pi_i(1-\pi_i) = \{g_\mu(\mu_i)\}^{-1}. \end{aligned}$$

So Eq.(2.20) becomes

$$X^t y = X^t \mu. \tag{2.27}$$

The IRLS iterative equations (2.25) become

$$\beta^{(k+1)} = (X^t W X)^{-1} X^t W[X\beta^{(k)} + \Delta^*(y-\mu)], \tag{2.28}$$

where

$$W = \text{diag}\{\pi_1(1-\pi_1), \ldots, \pi_n(1-\pi_n)\}$$

and

$$\Delta^* = \text{diag}\left\{\frac{1}{\pi_1(1-\pi_1)}, \ldots, \frac{1}{\pi_n(1-\pi_n)}\right\}.$$

The variance of the ML estimators $\hat{\beta}$ may be approximated from

$$\text{var}(\hat{\beta}) = (X^t W X)^{-1}. \tag{2.29}$$

2.6 Quasi-Likelihood

The quasi-likelihood concept comes from the fact of weighted least squares, or more generally from GLS (generalized least squares) in the case where responses are correlated (without

specifying the underlying distribution). Suppose $\text{var}(y_i) = \sigma^2 v(\mu_i)$, where σ^2 may be unknown but $v(\mu_i)$ is a known variance function. This formulation allows over and under dispersion .

Let

$$E(y_i) = \mu_i \ , \ u_i = \frac{y_i - \mu_i}{\sigma^2 v(\mu_i)}. \tag{2.30}$$

Then u_i behaves like a score function, as the 1st-order asymptotic theory related to likelihood functions is based on the three properties:

i) $E(u_i) = 0$.

ii) $\text{var}(u_i) = \dfrac{1}{\text{var}(y_i)}$.

iii) $-E\left(\dfrac{\partial u_i}{\partial \mu_i}\right) = \dfrac{1}{\text{var}(y_i)}$.

The quasi-likelihood is defined by

$$Q(\mu, y) = \int_y^\mu \frac{y - t}{\sigma^2 v(t)} dt. \tag{2.31}$$

For a set of n observations $y = (y_1, \ldots y_n)^t$, the quasi-likelihood is given by

$$Q(\mu, y) = \sum_{i=1}^{n} Q_i(\mu_i, y_i), \tag{2.32}$$

where $\mu = (\mu_1, \ldots \mu_n)^t$ is the mean of the response vector y. The quasi-likelihood estimator of β is obtained by maximizing Eq.(2.32).

2.6.1 Estimation of Parameters

To estimate the parameters β and σ^2 we can use the Newton-Raphson method with Fisher scoring as an iterative algorithm. We'll start by finding the score equations

$$\frac{\partial Q}{\partial \beta} = 0,$$

or

$$\sum_{i=1}^{n} \frac{\partial Q_i}{\partial \beta_i} = \sum_{i=1}^{n} \frac{\partial Q_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_i},$$

or

$$\sum_{i=1}^{n} \frac{y_i - \mu_i}{\sigma^2 v(\mu_i)} \frac{\partial \mu_i}{\partial \eta_i} x_i = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\sigma^2 v(\mu_i)} D_i, \qquad (2.33)$$

where

$$D_i = \frac{\partial \mu_i}{\partial \eta_i} x_i.$$

Then the quasi-score equations for β may be written in the matrix form

$$\frac{1}{\sigma^2}\Big[D^t V^{-1}(y - \mu)\Big] = 0, \qquad (2.34)$$

where

$$V = \text{diag}\{v(\mu_1), \ldots, v(\mu_n)\}.$$

The second partial derivative of the score function is

$$\frac{\partial^2 Q}{\partial \beta \partial \beta^t} = \frac{1}{\sigma^2}\left[\frac{\partial}{\partial \beta^t}(D^t V^{-1})(y - \mu) + (D^t V^{-1})\frac{\partial}{\partial \beta^t}(y - \mu)\right]. \qquad (2.35)$$

The expected value of Eq.(2.35) is obtained as

$$E\bigg(-\frac{\partial^2 Q}{\partial \beta \partial \beta^t}\bigg) = D^t V^{-1} D. \qquad (2.36)$$

Using the 1^{st} order Taylor's series approximation, we have

$$\frac{\partial Q}{\partial \beta} \approx \frac{\partial Q}{\partial \beta}\bigg|_{\beta_0} + \frac{\partial^2 Q}{\partial \beta \partial \beta^t}\bigg|_{\beta_0} (\beta - \beta_0). \qquad (2.37)$$

Then using the Fisher scoring replacement,

$$E\left(\frac{\partial^2 Q}{\partial \beta \partial \beta^t}\right)\bigg|_{\beta_0} = \frac{\partial^2 Q}{\partial \beta \partial \beta^t}\bigg|_{\beta_0}, \qquad (2.38)$$

the quasi-likelihood estimator of β may be obtained from the iterative equations

$$\beta_{k+1} = \beta_k + (D_k^t V_k^{-1} D_k)^{-1} D_k^t V_k^{-1} (y - \mu_k). \qquad (2.39)$$

For $k = 0, 1, 2, ...$ we continue the iterations until a convergence is achieved. The initial estimator β_0 at $k = 0$ may be obtained using an ordinary least squares method.

2.7 Example: Logistic Regression Model

In 1986, the space shuttle challenger exploded during take off, killing the seven astronauts aboard. The explosion was the result of an O-ring failure, a splitting of a ring of rubber that seals the parts of the ship together. The accident was believed to be caused by the unusually cold weather (31 degrees F or 0 degrees C) at the time of launch, as there is a reason to believe that the O-ring failure probabilities increase as temperature decreases. Table 2.2 (Presidential disaster commission testimonial in ER Tufte, Visual Explanations) presents temperature and O-ring failure data obtained from the Challenger accident. Here our goal is to assess how the temperature is associated with the O-rings failure based on a logistic regression model.

Table 2.2: Temperature and O-Ring failure data from the challenger accident

Temp. at Launch $^\circ F$	O-Ring Failure	Temp. at Launch $^\circ F$	O-Ring Failure
53	1	70	1
56	1	70	1
57	1	72	0
63	0	73	0
66	0	75	0
67	0	75	1
67	0	76	0
67	0	76	0
68	0	78	0
69	0	79	0
70	0	80	0
70	1	81	0

The logistic model can be written as

$$\text{Log}\left(\frac{p_i}{1-p_i}\right) = x_i^t\beta,$$

where p_i is the probability at failure. The fitted logistic regression model is

$$\hat{p_i} = \frac{\exp(x_i^t b)}{1+\exp(x_i^t b)}$$

or

$$\hat{p_i} = \frac{\exp(10.8753 - 0.171321x_i)}{1+\exp(10.8753 - 0.171321x_i)},$$

where x_i represents temperature (in $^\circ F$). The model was fitted using the R-package "glm". The regression coefficient of x_i is estimated as $b_1 = -0.171321$ with a standard error of 0.08344.

To interpret the parameters, in the case of a single regressor, the fitted linear predictor is

i) $\hat{\eta}(x_i) = b_0 + b_1 x_i$,

ii) $\hat{\eta}(x_i + 1) = b_0 + b_1(x_i + 1)$.

The difference is

$$\hat{\eta}(x_{i+1}) - \hat{\eta}(x_i) = b_1,$$

or

$$\ln\left(\frac{\text{odds}_{x_i+1}}{\text{odds}_{x_i}}\right) = b_1.$$

We obtain the estimated odds ratio

$$\widehat{OR} = e^{b1} = e^{-0.17134} = 0.84.$$

This implies that every additional degree of temperature reduces the odds of failure by 16 percent. If the temperature increases by d=5 degrees, then the odds ratio becomes $e^{(db_1)} = e^{5(-0.171321)} = 0.42$. This indicates that the odds of failure are reduced by over 50% with a 5 degree increase in the temperature.

2.8 Example: Poisson Regression Models

In a real data set of mine fractures, counts were observed on the number of injuries or fractures that occur in the upper seam of mines in the coal fields of the Appalachian region in the western Virginia. A total of 44 observations were collected on mines in this area, as

described below. Variables were measured. They all were functions of the material in the land and the mining area.

x_1: Inner burden thickness in feet (INB).

x_2: percent extraction of the lower previously mined seam (Extrp).

x_3: Lower seam height (feet).

x_4: Time that the mine has been opened (years).

Sample Data: Table 2.3 presents data from Myers (1990) describing the number of injuries that occur in West Virginia coal fields. The data consist of $n = 44$ observations from different mines. A portion of the data is shown below. The dependent variable is Fracture, which tabulates the number of injuries in each mine. The other 4 columns are potential predictor variables that quantify various attributes of each mine.

Table 2.3: Mine fracture data

Y	$X_1(INB)$	$X_2(EXTRP)$	$X_3(feet)$	$X_4(years)$
2	50	70	52	1
1	230	65	42	6
0	125	70	45	1
4	75	65	68	0.5
1	70	65	53	0.5
2	65	70	46	3
0	65	60	62	1
0	350	60	54	0.5
4	350	90	54	0.5
4	160	80	38	0
...	...	...	...	...

To model the response y as a function of the predictors, we consider the Poisson regression

model

$$\log(\mu_i) = \beta_0 + \beta_1 x_1 + ... + \beta_4 x_4,$$

where $\mu_i = E(y_i)$.

The fitted model is obtained as

$$\hat{\mu} = \exp(-3.59 - 0.0014x_1 + 0.0623x_2 - 0.00208x_3 - 0.0308x_4).$$

In order to choose a best model, we consider all subset models and their corresponding deviances, as shown in Table 2.4 below. The deviance is defined by $D = -2\{l(\hat{(\beta)} - l(\underline{y})\}$, where l represents the log-likelihood, $l(\hat{\beta})$ is the value of the log-likelihood for the Ml estimator $\hat{\beta}$ for a given model and $l(\underline{y})$ is the value of the log-likelihood at the "saturated model" or full model for which β-coefficients are allowed to be different for each individual.

Table 2.4: Deviance of the fitted models of mine fracture data

Model	Deviance
x_1	71.84
x_2	48.62*
x_3	73.84
x_4	72.12
x_1, x_2	42.09
x_1, x_3	71.07
x_1, x_4	70.43
x_2, x_3	47.59
x_2, x_4	41.63*
x_3, x_4	71.28
x_1, x_2, x_3	41.75
x_1, x_2, x_4	38.03*
x_1, x_3, x_4	69.81
x_2, x_3, x_4	41.02
x_1, x_2, x_3, x_4	37.86

To find the best model, we assess the change in deviance for adding/removing a covariate to the regression model. If we compare a model with covariates (x_1, x_2, x_4) to the full model with covariates (x_1, x_2, x_3, x_4), the change in deviance is obtained as

$$\begin{aligned}\Delta &= D(x_1, x_2, x_4) - D(x_1, x_2, x_3, x_4) \\ &= 38.03 - 37.86 \\ &= 0.17,\end{aligned}$$

which is not significant at level $\alpha = 0.10$ for a χ^2_1 distribution. So covariate x_3 appears not significant here. All other covariates appear to be significant. So we can choose our best model as

$$\hat{\mu} = \exp(-3.72 - 0.00147x_1 + 0.0627x_2 - 0.03165x_4).$$

Summary:

In this chapter, we have discussed several methods for fitting the generalized linear model. Among them, the IRLS method is commonly used for finding the maximum likelihood estimators. The R package "glm" provides ML estimators in the generalized linear models for both discrete and continuous responses.

Chapter 3

Generalized Linear Mixed Model (GLMM)

3.1 Introduction

In this chapter, we review methods for fitting GLMMs. In section 3.2, we present the form of the linear mixed model. In Section 3.3, we show the structure of GLMMs and its exponential form, and specify some particular properties. In Section 3.4, we investigate two iterative methods, MCEM and MCNR, for finding the ML estimators in GLMMs. In Section 3.5, we investigate the empirical properties of the ML estimators in binary mixed models using a Monte Carlo simulation study. We also analyze a real dataset as an application of the binary mixed model.

3.2 General Linear Mixed Model

The linear mixed model is an extension of the linear model

$$y = X\beta + \epsilon,$$

where we incorporate a random effects into the model and we suppose that the vector of model errors now is a multivariate normal with $E(\epsilon) = 0$, $\mathrm{Var}(\epsilon) = S$, and $S = \sigma_\epsilon^2 I_n$. The linear mixed model has the form

$$y = X\beta + Zu + \epsilon, \qquad (3.1)$$

where y is a $(n \times 1)$ vector of responses with $n = \sum_{i=1}^{m} n_i$, X is a $(n \times p)$ design matrix for the fixed effects, Z is a $(n \times qm)$ design matrix for the random effects, β is the associated $(p \times 1)$ vector of model parameters, and u is the associated $(qm \times 1)$ vector containing the m levels of each q random effects. The random effect u is assumed multivariate normal with $E(u) = 0$ and $\mathrm{Cov}(u) = D$ with D being a $(qm \times qm)$ positive definite matrix and $\mathrm{Cov}(\epsilon, u) = 0$. The variance-covariance matrix of y is

$$\mathrm{Var}(y) = T = \mathrm{Var}(X\beta + Zu + \epsilon),$$

and since we have

$$\mathrm{Cov}(\epsilon, u) = 0,$$

we have

$$T = Z\mathrm{var}(u)Z^{'} + \mathrm{var}(\epsilon), \qquad (3.2)$$

with

$$\mathrm{var}(\epsilon) = \mathrm{var}(y|u) = S.$$

Then from Eq.(3.2), we have

$$T = ZDZ^{'} + S. \qquad (3.3)$$

Remarks:

i) For the i^{th} cluster, model (3.1) may be written in the form

$$y_i = X_i\beta + Z_i u_i + \epsilon_i,$$

where y_i is a $(n_i \times 1)$ response vector with a variance $\text{var}(y_i) = T_i \;\; (i = 1, \ldots, m)$.

ii) The variance-covariance matrix T is no longer diagonal but block-diagonal

$$\begin{bmatrix} T_1 & 0 & 0 & \ldots & 0 \\ 0 & T_2 & 0 & \ldots & 0 \\ 0 & 0 & \ddots & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \ldots & T_m \end{bmatrix},$$

where $T_i = \text{var}(y_i)$ for $y_i = (y_{i1}, \ldots, y_{in_i})^t$, the response vector for the i^{th} cluster.

iii) The diagonal elements of D tell us how the level of random factor varies from one subject to another.

iv) We suppose observations are taken at equally spaced points in time, assumption of autoregressive structure, the variance of error term ϵ_i has the form

$$\text{var}(\epsilon_i) = S_i = \sigma_\epsilon^2 M_i.$$

In particular if $n_i = 5$, we can write $\text{var}(\epsilon_i)$ in the form

$$\text{var}(\epsilon_i) = \sigma_\epsilon^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 \\ \rho & 1 & \rho & \rho^2 & \rho^3 \\ \rho^2 & \rho & 1 & \rho & \rho^2 \\ \rho^3 & \rho^2 & \rho & 1 & \rho \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{bmatrix},$$

where ρ denotes the correlation among any two adjacent model errors in the cluster.

v) M_i denotes the correlation matrix associated with observations within the i^{th} cluster.

vi) In the case of balanced blocks, $M_i = M$ are identical for all subjects.

vii) In the case of equally correlated observations (exchangeable correlation, $n_i = 5$) we have

$$M_i = \begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix}.$$

3.2.1 Estimation

1) Regression parameters β

The marginal distribution of the response vector y is multivariate normal, and is given by

$$y \sim \text{ MVN}(X\beta, T),$$

where T is the marginal variance-covariance matrix of y. Then the generalized least square (GLS) estimator of β is obtained as

$$\hat{\beta} = (XT^{-1}X)^{-1}X^tT^{-1}y. \qquad (3.4)$$

We consider two cases:

i) If T is known, then the ML estimator of β is obtained from Eq.(3.4).

ii) If T is unknown, we can use the restricted maximum likelihood (REML) method, an approach developed by Patterson and Thompson (1971), in order to estimate T. Once the variance parameters are estimated, the fixed effects model coefficients are estimated by

$$\hat{\beta} = (X\hat{T}^{-1}X)^{-1}X^t\hat{T}^{-1}y,$$

where the variance of $\hat{\beta}$ may be approximated by

$$\widehat{\text{var}(\hat{\beta})} = (X\hat{T}^{-1}X)^{-1}.$$

2) Random coefficients u:

The conditional mean of the response y given the random effect u is given by

$$E(y|u) = X\beta + Zu. \qquad (3.5)$$

If we consider a particular cluster (subject), then

$$E(y_i|u_i) = X_i\beta + Z_iu_i, \qquad (3.6)$$

where $y_i = (y_{i1}, \ldots, y_{in_i})$ and n_i is the number of observations for the i^{th} subject. The mean response for the i^{th} subject is given by

$$E(y_i) = E[E(y_i|u_i)] = X_i\beta. \qquad (3.7)$$

The conditional distribution of y given u is given by

$$y|u \sim N(X\beta + Zu, S). \qquad (3.8)$$

As the levels of the random effects are fixed by the conditioning, so we consider only the

model error. The estimators of the random effects are obtained from differentiating the log of the joint likelihoods (the product of conditional response likelihood and the random effects likelihood) with respect to u. The BLUP (best linear unbiased predictor) of the random effect is obtained as

$$\hat{u} = DZ^t V^{-1}(y - X\beta), \tag{3.9}$$

where $V = ZDZ^t + S$.

The conditional mean $E(y|u)$ may be estimated as

$$\hat{E}(y|u) = X\hat{\beta} + Z\hat{u}. \tag{3.10}$$

.

3.3 Generalized Linear Mixed Model

Generalized linear mixed models (GLMMs) are a powerful class of statistical models that combine the characteristics of generalized linear models and mixed models (models that include both fixed and random predictor variables). They handle a wide range of response distributions, and a wide range of scenarios where observations are sampled in some form of groups rather than completely independently.

GLMMs are widely used in the analysis of clustered data, including longitudinal data or repeated measurements. GLMMs are useful for accommodating the overdispersion often observed among nonnormally distributed responses and for modelling the dependence among responses inherent in longitudinal or repeated measures data by incorporating random effects (Stiratelli, Laird, and Ware 1984; Zeger, Liang, and Albert 1988).

A full maximum likelihood (ML) analysis based on the joint marginal likelihood of the responses can be used for estimating both fixed and random effects parameters in GLMMs. The ML estimation requires numerical integration techniques for calculating the log-likelihood, score equations, and information matrix.

GLMMs are used to deal with consequences for having random effects other than fixed effects, where the levels used in a study for random effects represent a random sample from a much larger population of possible levels. For example, suppose we wish to study factors affecting cost of hospitalization by taking a random sample of patient records from each of 15 teaching hospitals. The costs within a hospital may be regarded as correlated. They will be similar because of the general costs of running the hospital, billing practices, costs of nearby competing hospitals, and so on. Also, a goal is to make inferences on a larger population of research hospitals. Both of these could be accommodated by incorporating random hospital effects into the model. A potential benefit could be gained by using an efficient prediction tool to improve the predictions for individual hospitals.

The GLMM relates the conditional mean of y_i given random effects b_i to the predictors X_i and Z_i by the link function

$$E(y_i|b_i) = g^{-1}(\eta_i) = g^{-1}(X_i\beta + Z_i b_i), \qquad (3.11)$$

where $y_i = (y_{i1}, \ldots, y_{in_i})^t$ is the vector of responses for the i^{th} cluster (or subject), g is a differentiable monotonic link function, η_i is a linear predictor given by

$$\eta_i = X_i\beta + Z_i b_i, \qquad (3.12)$$

with X_i and Z_i being design matrices for the fixed and random effects, and β and b_i are $(p \times 1)$ and $(q \times 1)$ vectors of fixed and random effects, respectively. The total number of

observations is $n = \sum_{i=1}^{m} n_i$, where n_i is the number of observations for the i^{th} subject.

The random effects b_i are assumed to be independent and normally distributed with mean 0 and a variance-covariance matrix D.

3.3.1 Exponential form of GLMM

Conditional on b_i, elements of $y_i = (y_{i1}, \ldots, y_{in_i})^t$ are assumed to be independent and follow a distribution in the exponential family

$$P(y_{ij}|b_i, \beta, \tau) = \exp\left\{ \frac{y_{ij}\theta(\eta_{ij}) - b(\theta(\eta_{ij}))}{\tau} + c(y_{ij}, \tau) \right\}, \tag{3.13}$$

where τ is a scalar dispersion parameter, $\theta(.)$ is a link function, $\eta_{ij} = x_{ij}^t\beta + z_{ij}^t b_i$ is a linear predictor, and x_{ij}^t and z_{ij}^t are j^{th} rows of the covariate matrices X_i and Z_i, respectively. When $\theta(\eta_{ij}) = \eta_{ij}$, the link is a canonical link. In the case of a Gaussian response model, we assume $\tau = \sigma^2$. In the case of Poisson and binary mixed models , we assume $\tau = 1$. The random effects b_i are independent normal, $b_i \sim N_q(0, D)$, where D is an unknown covariance matrix.

3.3.2 Marginal versus Conditional Distributions

3.3.2.1 Marginal Mean

The marginal mean of the j^{th} response y_{ij} for the i^{th} subject is given by

$$E(y_{ij}) = E_{b_i}(E(y_{ij}|b_i)) = E_{b_i}[g^{-1}(x_{ij}^t\beta + z_{ij}^t b_i)], \tag{3.14}$$

which cannot be simplified further, in general.

Example: Count Data

Count data can be modelled by a conditional Poisson distribution. In this case, the link function is $g(\mu) = \log(\mu)$, and $g^{-1}(x) = \exp(x)$. The marginal mean of y_{ij} is

$$\begin{aligned} E(y_{ij}) &= E_{b_i}[g^{-1}(x_{ij}^t\beta + z_{ij}^t b_i)] \\ &= E_{b_i}[\exp(x_{ij}^t\beta + z_{ij}^t b_i)] \\ &= \exp(x_{ij}^t\beta)E_{b_i}[\exp(z_{ij}^t b_i)], \end{aligned} \tag{3.15}$$

which can be simplified further using the moment generating function.

The moment generating function

$$M_X(t) = E(e^{tx}) \implies E_{b_i}[\exp(z_{ij}^t b_i)] = M_{b_i}(z_{ij}). \tag{3.16}$$

Then

$$E(y_{ij}) = \exp(x_{ij}^t\beta)M_{b_i}(z_{ij}). \tag{3.17}$$

Furthermore, if b_i is a scalar quantity with $b_i \sim N(0, \sigma_b^2)$ and if $z_{ij} = 1$, then we can show that

$$E(y_{ij}) = \exp(x_{ij}^t\beta)\exp\left(\frac{\sigma_b^2}{2}\right),$$

or

$$\log E(y_{ij}) = x_{ij}^t\beta + \frac{\sigma_b^2}{2}. \tag{3.18}$$

3.3.2.2 Marginal Variance

The marginal variance of y_{ij} may be obtained as

$$\begin{aligned}\text{var}(y_{ij}) &= \text{var}[E(y_{ij}|b_i)] + E[\text{var}(y_{ij}|b_i)] \\ &= \text{var}(\mu_{ij}) + E[\tau v(\mu_{ij})] \\ &= \text{var}\{g^{-1}(x_{ij}^t\beta + z_{ij}^t b_i)\} + E\{\tau v(g^{-1}(x_{ij}^t\beta + z_{ij}^t b_i)\}, \qquad (3.19)\end{aligned}$$

which cannot be simplified further, in general. Now consider the log-link with normal random effects used earlier. If the distribution is Poisson, then $v(\mu_{ij}) = \mu_{ij}$ and $\tau = 1$. Also, suppose $z_{ij}^t b_i = b_i$. Then we have

$$\begin{aligned}\text{var}(y_{ij}) &= \text{var}\{\exp(x_{ij}^t\beta + b_i)\} + E\{\exp(x_{ij}^t\beta + b_i)\} \qquad (3.20) \\ &= \text{var}\{\exp(x_{ij}^t\beta + b_i)\} + E(\mu_{ij}) \\ &= E[\exp\{2(x_{ij}^t\beta + b_i)\}] - \{E[\exp(x_{ij}^t\beta + b_i)]\}^2 + E[\exp(x_{ij}^t\beta + b_i)]. \qquad (3.21)\end{aligned}$$

After some simplification, we can find the marginal variance of y_{ij} as a function of β and σ_b^2. We can further show that

$$\text{var}(y_{ij}) > E(y_{ij}). \qquad (3.22)$$

In this sense, random effects b_i may be considered as a way to model possible overdispersion in count data.

3.3.2.3 Covariance and Correlations

The random effect b_i introduces correlations among the outcomes. We have

$$\begin{aligned}\text{corr}(y_{ij}, y_{ik}) &= \text{cov}[E(y_{ij}|b_i), E(y_{ik}|b_i)] + E[\text{cov}(y_{ij}, y_{ik}|b_i)] \\ &= \text{cov}[g^{-1}(x_{ij}^t\beta + z_{ij}^t b_i), g^{-1}(x_{ik}^t\beta + z_{ik}^t b_i)], \end{aligned} \tag{3.23}$$

since $E[\text{cov}(y_{ij}, y_{ik}|b_i)] = 0$ by assumption.

3.3.2.4 Joint Density

Consider a set of N response vectors $y_1 \dots y_N$ from N clusters. Then the joint density of $y = (y_1^t, \dots, y_N^t)^t$ and $b = (b_1^t, \dots, b_N^t)^t$ may be obtained as

$$p(y, b|\beta, D) = \prod_{i=1}^{N} \prod_{j=1}^{n_i} p(y_{ij}|\beta, b_i) p(b_i|D), \tag{3.24}$$

where $p(b_i|D)$ is the density of b_i. The marginal density of the response vector y may be obtained by integrating Eq.(3.24) with respect to b_i, given by

$$p(y|\beta, D) = \prod_{i=1}^{N} \int \prod_{j=1}^{n_i} p(y_{ij}|\beta, b_i) p(b_i|D)\, db_i. \tag{3.25}$$

The calculation of the density function $p(y|\beta, D)$ involves high-dimensional integration. The marginal density $p(y|\beta, D)$ doesn't have a closed form, in general, except in some special cases with certain link functions. Therefore, maximizing the observed data likelihood directly is not at all feasible. To reduce the computational burden, some Monte Carlo methods are available in the literature for finding approximate maximum likelihood estimators in GLMMs.

3.4 Maximum Likelihood Algorithms for GLMMs

In this section, we discuss two algorithms for finding approximate maximum likelihood estimators in GLMMs. These are referred to as Monte Carlo EM and Monte Carlo Newton-Raphson, as described below.

3.4.1 Monte Carlo EM

Suppose conditional on b_i, the response variable y_i follows a distribution in the exponential family

$$P(y_i|b_i,\beta,\tau) = \exp\left\{\frac{y_i\eta_i - b(\eta_i)}{\tau} + c(y_i,\tau)\right\}. \tag{3.26}$$

For a set of N observations $y_1,\ldots,y_N$, the likelihood function is

$$L(\beta,\tau,D|y,b) = \int \prod_{i=1}^{N} p(y_i|b_i,\beta,\tau)p(b_i|D)db_i. \tag{3.27}$$

In the EM method, we consider joint distribution of $y = (y_1,\ldots,y_N)^t$ and $b = (b_1,\ldots,b_N)^t$, where b is treated as "missing". The joint density is given by

$$p(y,b|\beta,\tau) = p(y|b,\beta,\tau)p(b|D).$$

The EM algorithm was explained by Arthur, Laird, and Rubin (1977) is described below.

3.4.1.1 EM Steps

1. Choose starting values $\beta^{(0)}$, $\tau^{(0)}$, and $D^{(0)}$. Set $m = 0$.
2. Calculate (with expectations evaluated under $\beta^{(m)}$, $\tau^{(m)}$, and $D^{(m)}$).

   i) $\beta^{(m+1)}$ and $\tau^{(m+1)}$, which maximize $E[\ln p(y|b,\beta,\tau)|y]$.

   ii) $D^{(m+1)}$, which maximizes the $E[\ln p(b|D)|y]$.

iii) Set $m = m + 1$.

3. If a convergence is achieved, declare the current values to be the MLE's, otherwise return to step 2.

In general, neither of the expectations in 2(i) or 2(ii) can be computed in a closed form for the model (3.13). This is because the conditional distribution of $b|y$ involves $p(y)$, i.e the likelihood which we are trying to avoid calculating directly. However, it is possible to produce random draws from the conditional distribution of $b|y$ by using the Metropolis algorithm (Tanner,1993), which doesn't require specification of $p(y)$. One can then form Monte Carlo approximations to the required expectations.

3.4.1.2 Metropolis Algorithm

The Metropolis algorithm which generates a Markov Chain sequence of values that eventually stabilizes to draws from the candidate distribution. To specify a Metropolis algorithm, a candidate distribution, $h_B(b)$, must be selected, from which potential new values are drawn. The acceptance function, which gives the probability of accepting a new value (as opposed to keeping the previous value) is given by

$$A_k(b^*, b) = \min\left[1, \frac{p(b^*|y, \beta, \tau, D)h_B(b)}{p(b|y, \beta, \tau, D)h_B(b*)}\right], \tag{3.28}$$

where $b^* = (b_1, b_2, ..., b_{k-1}, b_k^*, b_{k+1}, ..., b_q)^t$ is the candidate new value and has all entries equal to the previous value except the k^{th}. If we choose $h_B(D) = p(b|D)$, then the ratio term simplifies to

$$\frac{p(b^*|y, \beta, \tau, D)h_B(b)}{p(b|y, \beta, \tau, D)h_B(b^*)} = \frac{\prod_{i=1}^{N} p(y_i|b^*, \beta, \tau)p(b^*|D)p(b|D)}{\prod_{i=1}^{N} p(y_i|b, \beta, \tau)p(b|D)p(b^*|D)}$$

$$= \frac{\prod_{i=1}^{N} p(y_i|b^*, \beta, \tau)}{\prod_{i=1}^{N} p(y_i|b, \beta, \tau)}. \qquad (3.29)$$

This calculation involves specifying only the GLM model portion of the model, i.e the conditional density of y given b.

The MCEM algorithm can be described as follows:

1. Choose starting values $\beta^{(0)}$,$\tau^{(0)}$,and $D^{(0)}$. Set m=0.

2. Generate M values $b^{(1)}, b^{(2)}, ..., b^{(M)}$, from the conditional distribution of b given y using the Metropolis algorithm described above. Then:

i) Calculate $\beta^{(m+1)}$ and $\tau^{(m+1)}$ to maximize a Monte Carlo estimate of $E[\ln p(y|b, \beta, \tau)|y]$, i.e choose them to maximize

$$\frac{1}{M} \sum_{k=1}^{M} \ln p(y|b^{(k)}, \beta, \tau). \qquad (3.30)$$

ii) Calculate $D^{(m+1)}$ to maximize $\frac{1}{M} \sum_{k=1}^{M} \ln p(b^{(k)}|D)$.

iii) Set $m = m + 1$.

3. If a convergence is achieved, declare the current values to be the MLE's, otherwise return to step 2.

3.4.2 Monte Carlo Newton-Raphson

EM is a standard technique for linear mixed models. GLMMs are usually fit with a Newton-Raphson or scoring algorithm. It thus makes sense to develop a simulation analog of the

Newton-Raphson approach for fitting GLMMs. We start by noting that whenever the marginal density of y is formed as a mixture with separate parameters for $p(y|b)$ and $p(b)$, then the ML equations for $\theta = (\beta, \tau)$ and D take the following form

$$E\left[\frac{\partial \ln p(y|b,\theta)}{\partial \theta}\middle| y\right] = 0, \tag{3.31}$$

$$E\left[\frac{\partial \ln p(b|D)}{\partial D}\middle| y\right] = 0. \tag{3.32}$$

Eq.(3.32), involves only the distribution of b and is often fairly easy to solve. On the other hand, Eq.(3.31), can be solved by a Newton- Raphson or scoring approach as used in a standard generalized linear model.

Expanding

$$\frac{\partial \ln p(y|b,\theta)}{\partial \theta},$$

as a function of β around a value of $\theta_0 = (\beta_0^t, \tau_0^t)^t$ gives

$$\frac{\partial \ln p(y|b,\theta)}{\partial \beta} = \frac{\partial \ln p(y|b,\theta)}{\partial \beta}\bigg|_{\theta=\theta_0} + \frac{\partial^2 \ln p(y|b,\theta)}{\partial \beta \partial \beta^t}\bigg|_{\theta=\theta_0} (\beta - \beta_0). \tag{3.33}$$

Specializing this to our model, and dropping the term with a conditional expected value of zero, the formula for a scoring type algorithm becomes

$$\frac{\partial \ln p(y|b,\theta)}{\partial \beta} \approx X^t W(\theta_0, b) \frac{\partial \eta}{\partial \mu}\bigg|_{\theta=\theta_0} (Y - \mu(\theta_0, b))/\tau - X^t W(\theta_0, b) X(\beta - \beta_0)/\tau, \tag{3.34}$$

where

$$\mu_i(\theta, b) = E(y_i|b),$$

$$W(\theta, b)^{-1} = \text{diag}\left\{ \left(\frac{\partial \eta_i}{\partial \mu_i}\right)^2 \text{var}(y_i|b) \right\},$$

and

$$\frac{\partial \eta}{\partial \mu} = \text{diag}\left\{\frac{\partial \eta_i}{\partial \mu_i}\right\}.$$

Substituting the values of Eq.(3.34) into in Eq.(3.31) leads to an iteration equation of the form

$$\beta^{(m+1)} = \beta^{(m)} + E\left[X^t W(\theta^{(m)}, b) X | y\right]^{-1} \times X^t \left(E\left[W(\theta^{(m)}, b) \frac{\partial \eta}{\partial \mu}\bigg|_{\theta=\theta^{(m)}} \times (y - \mu(\beta^{(m)}, b)) \bigg| y \right] \right). \tag{3.35}$$

This analog of scoring would proceed by iteratively solving Eq.(3.31) and (3.32). An advantage of the scoring approach over MCEM is that it makes automatic the maximization step in Eq.(3.35). Again, the expectations cannot typically be evaluated in closed form, which leads to our Monte Carlo Newton-Raphson (MCNR) approach, as described below

1. Choose starting values $\beta^{(0)}$, $\tau^{(0)}$, and $D^{(0)}$. Set $m = 0$.

2. Generate M values $(b^{(1)}, \ldots, b^{(M)})$ from $P(b|y, \beta^{(m)}, \tau^{(m)}, D(m))$ using Metropolis algorithm described above and use them to form Monte Carlo estimates of the expectations, denoted by $\tilde{E}(.)$ from the conditional distribution of b given y. Then

i) Calculate

$$\beta^{(m+1)} = \beta^{(m)} + \tilde{E}[X^t W(\theta^{(m)}, b) X | y]^{-1} \times X^t \left(\tilde{E}\left[(W(\theta^{(m)}, b) \frac{\partial \eta}{\partial \mu}\bigg|_{\theta=\theta^{(m)}} \times (y - \mu(\beta^{(m)}, b)) \bigg| y \right] \right).$$

ii) Calculate $\tau^{(m+1)}$ by solving $E\left[\dfrac{\partial \ln p(y|b,\theta)}{\partial \tau} \bigg| y\right] = 0$.

iii) Calculate $D^{(m+1)}$ by maximizing $\frac{1}{M}\sum_{k=1}^{M} \ln p(b^{(k)}|D)$ with respect to D.

iv) Set $m = m + 1$.

3. If convergence is achieved, declare the current values to be the MLE's, otherwise return to step 2.

Remark: A natural question concerns the convergence properties of these algorithms. For sufficiently large simulation sample sizes, MCEM or MCNR would inherit the properties of the exact versions. So MCEM would inherit the likelihood increasing properties of EM and would, under suitable regularity conditions (see, e.g., Wu 1983), converge to a local maximum. Newton-Raphson algorithms do not guarantee convergence properties when the surfaces to be maximized are not concave.

3.4.3 Binary Mixed Model

In generalized linear mixed models (GLMMs), the logistic regression analysis is often used to investigate the relationship between a binary response variable and a set of explanatory variables. A binary response consists, for example, of success or failure. To analyze this type of categorical variable which does not follow a normal distribution, the logistic regression model is commonly used in which the mean response is described as a function of the linear predictor by a logit link function.

Suppose that the dispersion parameter $\tau = 1$. The conditional distribution of $Y|b$ is assumed to follow the Bernoulli distribution:

$$Y_{ij}|b_i \sim \text{independent Bernoulli}(p_{ij}), \quad i = (1, \ldots N\ ; j = 1, \ldots, n_i),$$

$$b_i \sim \text{independent } N(0, \sigma_b^2),$$

$$\eta_{ij} = \log\left(\frac{p_{ij}}{1 - p_{ij}}\right) = \beta x_{ij} + b_i,$$

$$E[Y_{ij}|b_i] = p_{ij}(\beta, b_i) = \frac{e^{\eta_{ij}}}{1 + e^{\eta_{ij}}},$$

$$\text{var}(Y_{ij}|b_i) = \sigma^2_{ij}(\beta, b_i) = \frac{e^{\eta_{ij}}}{(1+e^{\eta_{ij}})^2},$$

$$\frac{\partial \eta_{ij}}{\partial p_{ij}} = \frac{1}{p_{ij}(1-p_{ij})} = \frac{1}{\sigma^2_{ij}},$$

$$\frac{\partial p_{ij}}{\partial \beta} = (p_{ij})(1-p_{ij})x_{ij},$$

where y_{ij} is the j^{th} response from the i^{th} cluster. Our goal is to estimate β and variance component σ^2_b by the ML method.

The log-likelihood of $\gamma = (\beta, \sigma^2_b)$ is given by

$$l(\beta, \sigma^2|y, b) = \log \int \prod_{i=1}^{N} p(y_i|b_i, \beta) p(b_i|\sigma^2_b) db_i,$$

$$l(\beta, \sigma^2|y, b) = \sum_{i=1}^{N} \log \int \prod_{j=1}^{n_i} P(y_{ij}|b_i, \beta) p(b_i|\sigma^2_b) db_i.$$

The ML estimator of γ may be obtained from iterative equations:

$$\gamma^{(m+1)} = \gamma^{(m)} + \left[-\frac{\partial^2 l}{\partial \gamma \partial \gamma^t} \right]^{-1}_{\gamma^{(m)}} \left[\frac{\partial l}{\partial \gamma} \right]_{\gamma^{(m)}},$$

for $m = 0, 1, 2, ...$ In the above equation, we have

$$\text{score function } \left[\frac{\partial l}{\partial \gamma} \right]_{\gamma^{(m)}} = \left[\frac{\partial l}{\partial \beta} \quad , \quad \frac{\partial l}{\partial \sigma^2_b} \right]^t_{\gamma^{(m)}},$$

and the

$$\text{Fisher information } \left[-\frac{\partial^2 l}{\partial \gamma \partial \gamma^t} \right]^{-1}_{\gamma^{(m)}}.$$

The partial log-likelihood w.r.t β is given by

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{N} E_b\Big[\sum_{j=1}^{n_i}(y_{ij} - p_{ij})x_{ij}\big|y\Big].$$

In this case, the Fisher Information matrix has the form

$$\frac{\partial^2 l}{\partial \gamma \partial \gamma^t} = \begin{bmatrix} \dfrac{\partial^2 l}{\partial \beta \partial \beta^t} & \dfrac{\partial^2 l}{\partial \beta \partial \sigma_b^2} \\ \dfrac{\partial^2 l}{\partial \sigma_b^2 \partial \beta} & \dfrac{\partial^2 l}{\partial \sigma_b^2 \partial \sigma_b^2} \end{bmatrix},$$

where

$$\frac{\partial^2 l}{\partial \beta \partial \beta^t} = -\sum_{i=1}^{N} X_i^t E_b(\Delta_i|y_i)X_i + \sum_{i=1}^{N} X_i^t E_b[(y_i - p_i)(y_i - p_i)^t|y_i]X_i$$

$$-\sum_{i=1}^{N} X_i^t E_b\left[(y_i - p_i)|y_i\right] E_b\left[(y_i - p_i)|y_i\right]^t X_i,$$

$$\Delta_i = \text{diag}\{\Delta_{ij}\},$$

with

$$\Delta_{ij} = p_{ij}(1 - p_{ij}).$$

The ML estimators of (β, σ_b^2) may be obtained numerically using a similar iterative algorithms as described earlier in Sections 3.4.1 and 3.4.2.

3.5 Simulation Study

To evaluate the performance of the ML method, two sets of small simulation studies were conducted. The set of simulations compared the biases and mean squared errors (MSE's) of the parameter estimates of the GLMM's .

The data were generated from the binary mixed model

$$Y_{ij}|b_i \sim \text{independent} \;\; \text{Bernoulli}(p_{ij}), \quad i = (1, \ldots N \; ; j = 1, \ldots, n_i),$$

$$\text{logit}(p_{ij}) = \beta_0 + \beta_1 x_{ij} + b_i,$$

$$b_i \sim \text{independent } N(0, \sigma_b^2).$$

We generated a series of 1000 replicates of datasets for each combination of the number of clusters $N = \{100, 400\}$ and an equal cluster size $n_i = n = 4$. with the values of the regression parameters fixed at $\beta_0 = -1$, $\beta_1 = 1$ and the variance component fixed at $\sigma_b^2 = 0.5$. Values of the covariates x_{ij} were generated from $N(0, 1)$. We used the R function "glmmML" to fit the binary mixed models. Tables 3.1 and 3.2 present empirical biases, mean squared errors, coverage probabilities and average lengths of 95% confidence intervals for the ML estimators of model parameters.

3.5.1 Investigation Methods

We studied the empirical properties of the maximum likelihood estimators in terms of biases and mean squared errors (MSEs) of the estimators, as well as coverage probabilities (CPs) and average lengths of the individual confidence intervals for the regression parameters.

Bias: The bias of an estimator $\hat{\theta}$ of a parameter θ is obtained as the difference between the expected value of $\hat{\theta}$ and the true value of the parameter θ, given by

$$\text{Bias}(\hat{\theta}) = E(\hat{\theta}) - \theta \approx \frac{1}{S}\sum_{s=1}^{S} \hat{\theta}^s - \theta,$$

where $\hat{\theta}^s$ is the estimate of θ obtained at the s_{th} simulation and S is the simulation size.

Mean Squared Error: The mean squared error (MSE) of an estimator $\hat{\theta}$ of a parameter θ can be obtained as

$$\text{MSE}(\hat{\theta}) = E(\hat{\theta} - \theta)^2 \approx \frac{1}{S}\sum_{s=1}^{S} (\hat{\theta}^s - \theta)^2,$$

where $\hat{\theta}^s$ is the estimate of θ obtained at the s_{th} simulation and S is the simulation size.

Coverage Probability: The coverage probability of an estimator $\hat{\theta}$ for a 95% confidence

interval on θ is obtained from

$$CP(\hat{\theta}) = \frac{1}{S}\sum_{s=1}^{S} I\{\mid \hat{\theta}^s - \theta \mid \leq 1.96 \times SE(\hat{\theta}^s)\},$$

where SE($\hat{\theta}$) is an estimate of the standard error of $\hat{\theta}$ and I is an indicator variable. The 95% confidence interval for θ is obtained as

$$CI(\hat{\theta}) = \hat{\theta} \pm 1.96 \times SE(\hat{\theta}).$$

Average Length: The length of a confidence interval for θ is obtained as the difference between two confidence limits:

$$L^c(\hat{\theta}) = \{\hat{\theta} + 1.96 \times SE(\hat{\theta})\} - \{\hat{\theta} - 1.96 \times SE(\hat{\theta})\}.$$

The average length is obtained by

$$\text{Ave.}L^c(\hat{\theta}) = \frac{1}{S}\sum_{s=1}^{S} L^c(\hat{\theta}^s),$$

where S is the simulation size.

3.5.2 Results

Table 3.1: Empirical biases, mean squared errors, coverage probabilities and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models for sample size ($N = 100, n = 4$).

Estimator	True values	Bias	SE	MSE	Coverage	Length
$\hat{\beta}_0$	-1	-0.022	0.147	0.0225	94.6	0.587
$\hat{\beta}_1$	1	0.017	0.152	0.0243	95.0	0.541
$\hat{\sigma}_b^2$	0.5	-0.050	0.407	0.0877	97.4	0.958

Table 3.2: Empirical biases, mean squared errors, coverage probabilities and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models for sample size ($N = 400, n = 4$).

Estimator	True values	Bias	SE	MSE	Coverage	Length
$\hat{\beta}_0$	-1	0.0003	0.073	0.0056	93.8	0.305
$\hat{\beta}_1$	1	0.0003	0.075	0.0056	94.5	0.305
$\hat{\sigma}_b^2$	0.5	-0.023	0.169	0.0241	97.0	0.568

Discussion:

It is clear from Tables 3.1 and 3.2 that the ML estimates of both regression and variance parameters are roughly unbiased. For the smaller sample size $N = 100$, we observe slight bias in the estimates, but the bias tends to decrease when the sample size N increases. As expected, the SEs and MSEs decrease as the sample size N increases. The empirical coverage probabilities are close to their nominal 95% confidence levels. We also observe that the empirical average lengths of the confidence intervals decrease when the sample size N increases.

3.5.3 Example: GUIDE Data

Preisser and Qaqish (1999) analyzed an interesting set of data from the Guidelines for Urinary Incontinence Discussion and Evaluation (GUIDE) study. The purpose of this study was to identify factors among urinary incontinent men and women of age 76 or above that are predictive of their responses to the question of whether an individual in that age group considers this accidental loss of urine a problem that interferes with his or her day-to-day activities or bothers him or her in other ways.

In this study, 137 patients from 38 medical practices were investigated. The binary response variable $y_{ij} = 1$ if the j^{th} patient from the i^{th} medical practice is bothered by the urinary incontinence, and 0 if not. The predictors considered are gender, age, weekacc = how many leaking accidents the patients have in an average week, severe = 1 if it just creates some moisture when they lose urine, 2 if it wets their underwear (or pad), 3 if it trickles down their thigh, 4 if it wets the floor, and toilet= how many times during the day they usually to go the toilet to urinate, dayacc = weekacc/7.

We analyzed the GUIDE data using a conditionally independent logistic model with the conditional means for the j_{th} patient from the i_{th} medical practice specified by

$\text{logit}(p_{ij}) = x_{ij}^t\beta + b_i$, where the random effects b_i are assumed to be iid $N(0, \sigma_b^2)$.

The logistic model is given by

$$\text{logit}(p_{ij}) = \beta_0 + \beta_1(\text{Female})_{ij} + \beta_2(\text{Age})_{ij} + \beta_3(\text{Dayacc})_{ij} + \beta_4(\text{Severe})_{ij} + \beta_5(\text{Toilet})_{ij} + b_i.$$

We use the Newton-Raphson iterative method for finding the ML estimates of the model parameters. The R code for computing the estimates is presented in Appendix B-2. The R function "glmmML" package can also be used to find the ML estimates, which is described

in Appendix B-3.

Table 3.3: Maximum likelihood estimates of the regression parameters and variance component in GLMM used for analyzing GUIDE data

Parameter	Estimate	SE	Z-value	P-value
Intercept	-3.5244	1.34487	-2.621	0.0087***
Female	-0.9120	0.73703	-1.237	0.2160
Age	-0.7828	0.71566	-1.094	0.2740
Dayacc	0.4662	0.12319	3.784	0.0001***
severe	0.9618	0.41849	2.298	0.0215*
Toilet	0.1159	0.09796	1.183	0.2370
σ_b^2	1.2179	1.38460	0.879	0.1895

Table 3.3 presents the maximum likelihood estimates of the model parameters together with their standard errors (SEs), Z-values and P-values from Wald tests for testing the significance of the predictors. It appears that both "dayacc" and "severe" have positive effects on the outcome of urinary incontinent of the patients. The P-value of the test $H_0 : \sigma_b^2 = 0$ vs $H_1 : \sigma_b^2 > 0$ indicates that the variance component σ_b^2 is significant at the 0.1895 level.

Summary:

GLMMs are useful for analyzing clustered and longitudinal data. We have discussed several algorithms for approximating the maximum likelihood estimators in GLMMs. The R package "glmmML" is available for fitting a generalized linear mixed model with a single random effect. For multiple random effects, however, we can use the approximate MCNR or MCEM methods discussed earlier.

Chapter 4

Generalized Linear Mixed Models with Missing Responses

4.1 Introduction

In this chapter, we present the methods to fit GLMMs to binary data sets with nonignorable misingness in the response. In section 4.2, we review some literature on missing data. In Section 4.3, we discuss some literature on missing responses. In Section 4.4, we investigate a method for estimating the parameters of a missing data model. In Section 4.5, we discuss the models for the missing data. In Section 4.6, we investigate the empirical properties of the MLE's based on a simulation study. In Section 4.7, we apply the MLE method to a real data set obtained from a clinical study.

4.2 Literature on Missing Data

Missing data are common in many experimental studies, including surveys and clinical trials. Little and Rubin (2002) review various missing data patterns or mechanisms, which concern the relationship between missingness and the values of the variables in the data. If missingness does not depend on the values of the data, missing or observed, the data are called

missing completely at random (MCAR). A less restrictive assumption is that missingness depends only on the observed values of the variables in the data, and not on the values that are missing. This missing data mechanism is referred to as missing at random (MAR). Rubin (1976) showed that if the data are MAR, then the likelihood-based inference does not depend on the missing data mechanism. If missingness is related to values of the missing variables, then the missing data are called nonignorable. In the case of nonignorable missing data, it is important to model the missing data mechanism. Little (1995) discusses techniques for modelling the data and the missing data mechanism simultaneously, and presents a number of examples to describe likelihood-based inferences via maximum likelihood or Bayesian methods.

The missing data problem had been formerly treated by Rubin (1976), where the missingness could be either in responses or in covariates. However, Rubin's primary focus was on continuous measurements. In biological experiments or social studies, responses are often discrete, in which case statistical analyses at multivariate level become difficult due to the lack of a proper multivariate distribution for discrete data

The missing data may lead to invalid inference if missingness is not properly addressed. An appropriate approach is needed to deal with the missing data. A wide range of analysis of missing data mechanism with incomplete data has been done by several authors in the literature (e.g., Brown 1990; Dantan et al. 2008; Diggle and Kenward 1994; Ibrahim et al. 1999, 2001; Molenberghs and Verbeke 2001; Preisser et al. 2000; Sinha 2008; Verbeke and Molenberghs 2005; Wu et al. 2009; Xie 2008; Yi and Cook 2002 and many others).

Greenlees, Reece, and Zieschang (1982) proposed a parametric model using survey data for maximum likelihood estimators for both missing data mechanism and data generating process to deal with nonignorable nonresponse.

Later on, Tang, Little, and Raghunathan (2003) suggested a pseudo-likelihood for nonignorable nonresponse, implementing a parametric model on the data generating process. Their work was slightly different from Greenlees, Reece, and Zieschang (1982) in the sense that the missing data mechanism could be unspecified. This approach estimates population parameters in the complete data case considering the missing data mechanism as a nuisance parameter.

Sinha (2008) used a generalized linear model for maximum likelihood estimation. He proposed a robust approach to downweight any influential observation in the presence of nonignorable missing covariates. He adopted a Metropolis-Hastings algorithm based on a Markov chain sampling method to carry out some simulations. He examined the behavior of the robust estimates and compared them to the classical maximum likelihood estimates. Finally, he applied his method to real life data of delirium patients.

4.3 Literature on Missing Responses

There is extensive research done with missing responses by many authors in the biological research area. The generalized linear mixed model was proposed by Ibrahim, Lipsitz and Chen (2001) for missing responses with nonignorable missing data mechanism. They developed a Monte Carlo EM algorithm for estimating the parameters in the model via a Gibbs sampler. Furthermore, for the normal random effect models model, they proposed a novel analytical form for the E and M steps, which was facilitated by integrating out the random effects. This form led to a computationally feasible and extremely efficient Monte Carlo EM algorithm for computing maximum likelihood estimates and standard errors. They illustrated two real

data sets from the International Breast Cancer Study Group and an environmental study involving dyspnoea in cotton workers.

Liang and Zeger (1986) proposed a unifying approach to such analysis for a variety of discrete and continuous outcomes. They considered a class of generalized estimating equations (GEEs) for the regression parameters. The authors extended equations of those used in quasi-likelihood, where GEEs have solutions which were consistent and asymptotically Gaussian. They illustrated the use of the GEE approach with longitudinal data from a study of the effect of mothers' stress on children's morbidity.

Baker and Laird (1988) used the EM algorithm to obtain ML estimates for categorical variables with outcome subject to nonignorable nonresponse. They proposed the Log-linear models which were particularly convenient for analyzing categorical data subject to nonresponse. They simplified the simultaneous fitting of both margin models for response and models for the response mechanism in the M step of the EM algorithm.

Sinha (2011) considered analyzing the CD4 count data from clinical trials of HIV-infected patients measured at only two time points. The baseline time 0 (TIME = 0) and the week 5 (TIME = 1) after the baseline period. All patients were observed at the baseline time 0, but only 285 patients (66%) were observed at week 5. The CD4 counts appeared to be generally lower at TIME = 1 than those measured at the baseline TIME = 0. His goal was to investigate the influence of these observations on the robust and classical estimates of the model parameters. The author performed his numerical study using Poisson mixed models with a single random effect and a simple nonignorable missing data model.
He proposed an efficient robust method in the framework of the maximum likelihood estimation by incorporating a missing data model into the observed data likelihood function. The RML method was almost as efficient as the ML method for uncontaminated data. But

when data were contaminated with outliers, the proposed RML method appeared to provide smaller biases and MSEs for estimating the model parameters, as compared to the classical ML method.

4.4 Estimation with Missing Data

Let $y_{\text{obs},i}$ denote the observed values and $y_{\text{mis},i}$ the missing values of the response vector y_i for individual i. Assuming arbitrary, nonmonotone patterns of missing data in y_i, we can write $y_i = (y_{\text{mis},i}, y_{\text{obs},i})$, which means that some permutation of the indices of y_i can be written as $(y_{\text{mis},i}, y_{\text{obs},i})$. The missing data mechanism is defined as the distribution of the $n_i \times 1$ random vector r_i, whose j^{th} components, r_{ij} equals 1 if y_{ij} is observed for subject i, and 0 if y_{ij} is missing. The conditional distribution of r_i given y_i, denoted by $[r_i|y_i, \tau]$, is indexed by the parameter vector τ. We will assume throughout that the distribution r_i depends only on y_i and not on b_i. Given y_i, the vector r_i is assumed to follow a distribution $p(r_i|y_i, \tau)$ depending on some parameters τ.

Treating (y_i, b_i, r_i) as complete data, the complete data likelihood of (β, θ, τ) for individual i is obtained from

$$p(y_i, b_i, r_i|\beta, \theta, \tau) = p(y_i|b_i, \beta)p(b_i|\theta)p(r_i|y_i, \tau). \tag{4.1}$$

Now, suppose $\{(y_{\text{mis},i}, b_i); i = 1, \ldots, N\}$ are the "missing" data. Then the full likelihood of (β, θ, τ) is any function of (β, θ, τ) proportional to the marginal density of the observed data $\{(y_{\text{obs},i}, r_i); i = 1, ..., N\}$:

$$L_{\text{full}}(\beta, \theta, \tau|y_{\text{obs}}, r) = \prod_{i=1}^{N} \int p(y_i|b_i, \beta)p(b_i|\theta)p(r_i|y_i, \tau)d(y_{\text{mis},i}, b_i), \tag{4.2}$$

where $y_{\text{obs}} = (y^t_{\text{obs},1}, \ldots y^t_{\text{obs},N})t$ and $r = (r^t_1, \ldots r^t_N)^t$.

In this setting, the linear predictor $\eta_{it} = x^t_{it}\beta + z^t_{it}b_i$. From Eq.(4.2), the ML estimating equations for β, θ and τ take the form:

$$\sum_{i=1}^{N} E\left\{ \frac{\partial \log p(y_i|b_i, \beta, \theta)}{\partial \beta} \middle| y_{\text{obs},i}, r_i \right\} = 0, \qquad (4.3)$$

$$\sum_{i=1}^{N} E\left\{ \frac{\partial \log p(b_i|\theta)}{\partial \theta} \middle| y_{\text{obs},i}, r_i \right\} = 0, \qquad (4.4)$$

and

$$\sum_{i=1}^{N} E\left\{ \frac{\partial \log p(r_i|y_i, \tau)}{\partial \tau} \middle| y_{\text{obs},i}, r_i \right\} = 0, \qquad (4.5)$$

where the conditional expectations are taken with respect to $(y_{\text{mis},i}, b_i)$, given the actual observed data $(y_{\text{obs},i}, r_i)$. These equations can be solved iteratively for the ML estimators of the parameters. Our main interest is in the estimation of the regression parameters β and the variance components θ, with τ being viewed as the nuisance parameters.

4.5 Models for the Missing Data

Nonignorable models are needed when the missing data mechanism depends on the missing observations or unobserved random effects. Little (1995) discusses these two kinds of non-ignorable dropout. In the "outcome-dependent" setting, dropout may depend on missing components of y_i for individual i . Symbolically:

$$p(r_i|y_i, b_i, \tau) = p(r_i|y_i, \tau). \qquad (4.6)$$

Examples of outcome-dependent dropout are given in Little and Rubin (2002). Diggle

and Kenward (1994) and Ibrahim et al. (2001) consider nonignorable models with outcome-dependent dropout. In the "random coefficient-dependent" setting, dropout may depend on the underlying random coefficients b_i .

Here we consider nonignorable models with outcome-dependent dropout. In this setting, a possible model for the missing data mechanism is the binomial model:

$$p(r_i|y_i,\tau) = \prod_{j=1}^{n_i} \pi_{ij}^{r_{ij}}(1-\pi_{ij})^{1-r_{ij}}, \qquad (4.7)$$

where $\pi_{ij} = E(r_{ij}) = p(r_{ij}=1|y_{ij},\tau)$, the probability of the response given y_{ij}, is modeled by a logistic regression involving (y_{ij}, x_{ij}). This model assumes "working" independence among the elements in the vector r_i conditional on the vector of responses, y_i. For a more general form of the missing data mechanism, see Little (1995) and Little and Rubin (2002). Ibrahim et al. (1999) consider modelling the missing data mechanism $p(r_i|y_i,\tau)$ as the product of a sequence of one dimensional conditional distributions:

$p(r_{i1},\ldots,r_{in}|y_i,\tau) = p(r_{in}|r_{i1},\ldots,r_{i,n-1},y_i,\tau_n)\times p(r_{i,n-1}|r_{i1},\ldots,r_{i,n-2},y_i,\tau_{n-1})\times\ldots\times p(r_{i2}|r_{i1},y_i,\tau_2)\times p(r_{i1}|y_i,\tau_1)$.

Note that in model (4.7), we consider the "working" independence assumption among the elements of r_i given the response vector y_i for computational simplicity. According to many references, dealing with incomplete longitudinal data suggests that the working independence assumption is typically not much sensitive to a misspecified missing data mechanism.

4.6 Simulation Study

To asses the performance of the ML estimators in GLMMs with nonignorable missing responses, we ran a series of simulations. Data were generated from the binary mixed model

$$y_{ij}|b_i \sim \text{independent Bernoulli}(p_{ij}); \quad i = (1,\ldots N\ ; j = 1,\ldots,n_i),$$

$$\text{logit}(p_{ij}) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_0 + \beta_1 x_{ij} + b_i, \tag{4.8}$$

$$b_i \sim \text{Independent } N(0, \sigma_b^2).$$

The values of the predictor x_{ij} were generated from $N(0,1)$.
The missing data indicator r_{ij} ($r_{ij} = 1$ if y_{ij} is observed, and 0 if y_{ij} is missing) is assumed to follow an independent Bernoulli distribution with

$$\pi_{ij} = p(r_{ij} = 1 | y_{ij}, x_{ij}, \tau) = \frac{\exp(\psi_{ij})}{1+\exp(\psi_{ij})}, \tag{4.9}$$

where $\psi_{ij} = \tau_0 + \tau_1 x_{ij} + \tau_2 y_{ij}$. In this case, the joint density of $r = (r_1^t, \ldots, r_N^t)$, with $r_i = (r_{i1}, \ldots, r_{in})^t$, may be defined as

$$\begin{aligned} p(r|x,y) &= \prod_{i=1}^{N}\prod_{j=1}^{n} p(r_{ij}|x_{ij}, y_{ij}, \tau) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{n} \pi_{ij}^{r_{ij}}(1-\pi_{ij})^{1-r_{ij}}. \end{aligned} \tag{4.10}$$

We generated a series of 500 replicates of datasets for each combination of the missing data parameters $\tau_1 = \{0, 0.5\}$, $\tau_2 = \{-3, -1\}$, and sample sizes $N = \{100, 200, 300\}$ and $n = \{2, 3\}$. The other parameters were fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0) = (-2, 3, 1, 2)$. Note that the non-zero values of $\tau_2 = \{-3, -1\}$ indicates that the response y is not missing at random (NMAR). The choice of $\tau_2 = -1$ resulted in a roughly 20% missing values in the response variable, whereas $\tau_2 = -3$ resulted in higher proportion (roughly 45%) of missing values in the response.

Table 4.1- 4.8 present empirical biases, mean squared errors, coverage probabilities, and average length of 95% confidence intervals for maximum likelihood estimators in logistic regression models under different sample sizes $N = \{100, 200, 300\}$ and with $n = \{2, 3\}$. The

regression models were fitted using two methods: a) correctly specified NMAR, where we maximize the likelihood under non-ignorable missing data models, and b) misspecified MAR, where we ignore the missing data model when finding the maximum likelihood estimators of regression parameters. We use the R function "glmm.like.miss" as shown in the appendix.

It is clear from the results in Table 4.1- 4.8 that under correctly specified NMAR models, the ML estimators of the regression parameters $(\beta_0, \beta_1, \beta_2)$, variance components σ_b^2 and nuisance parameters (τ_0, τ_1, τ_2) of the missing data model are approximately unbiased and also provide coverage probabilities that are generally close to the nominal 95% confidence level.

On the other hand, under the misspecified MAR model (i.e, under the assumption of ignorable missing responses), the ML estimators appear to be generally biased under all scenarios. For example, when $N = 300$ and $n = 2$, we observe from Table 4.1 that under the MAR model the ML estimator of β_0 provides an empirical bias of -1.419 (71% relative bias) and a small coverage probability of 62%. On the other hand, under the NMAR model, the ML estimator of β_0 provides a much smaller bias of -0.206 (10% relative bias) and a better coverage probability of 86%. As expected, the mean squared errors of the estimators generally decrease for increased sample sizes. The average length of the confidence intervals also decrease with the increased sample size under both NMAR and MAR models.

When comparing the estimators under different proportions of missing responses (20%) missing with $\tau_2 = -1$ and 45% missing with $(\tau_2 = -3)$, we observe that the ML estimators under the misspecified MAR models provide longer systematic bias for higher proportions of missing data. For example, we observe from Table 4.5 that at $\tau_2 = -1$ with 20% missing values, the ML estimates of β_0 under the MAR model provide an empirical bias of -0.397(20% relative bias) and a coverage probability of 95% for the sample size $(N = 100, n = 2)$, whereas at $\tau_2 = -3$ with 45% missing values, Table 4.1 shows that the corresponding ML estimates of

β_0 provides a much larger empirical bias of -1.380 (69% relative bias) and a lower coverage probability of 85%. As the data were generated from NMAR assumption then we expect the systematic bias and poor coverage probability from the misspecified MAR model do not improve when the sample size increased.

It is also important to note that for larger proportions of missing data, the correctly specified NMAR model also provides some bias and low coverage probabilities, but the extent of the bias and poor coverage is much less by the NMAR model, as compared to the misspecified MAR model.

Table 4.1: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0, -3)$. Cluster size $n = 2$.

		NMAR				MAR			
(N=100,n=2)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	-1.043	0.4328	88.09	4.730	-1.380	0.4860	85.58	4.861
β_1	3	0.891	0.3337	90.57	5.033	0.382	0.2508	96.88	6.504
σ_b^2	1	2.292	0.3977	94.53	13.20	1.119	0.3838	79.56	8.578
τ_0	2	0.023	0.2873	85.18	3.471	-	-	-	-
τ_1	0	-0.051	0.2434	97.20	1.312	-	-	-	-
τ_2	-3	0.210	0.3358	96.08	3.072	-	-	-	-
(N=200,n=2)									
β_0	-2	-0.311	0.2309	88.88	2.872	-1.391	0.5376	76.96	3.594
β_1	3	0.449	0.1978	90.85	3.103	0.279	0.2286	93.58	4.282
σ_b^2	1	1.210	0.2704	91.30	7.231	1.106	0.3499	81.37	5.440
τ_0	2	0.154	0.2032	91.13	12.72	-	-	-	-
τ_1	0	-0.021	0.0553	93.89	0.897	-	-	-	-
τ_2	-3	-0.072	0.2889	93.05	2.300	-	-	-	-
(N=300,n=2)									
β_0	-2	-0.206	0.4397	85.78	2.22	-1.419	0.6086	62.09	3.102
β_1	3	0.245	0.1908	85.36	2.23	0.281	0.2052	94.16	3.568
σ_b^2	1	0.680	0.6592	86.63	5.015	1.196	0.3304	74.92	4.526
τ_0	2	0.282	0.1767	86.58	2.46	-	-	-	-
τ_1	0	-0.044	0.2622	88.18	0.749	-	-	-	-
τ_2	-3	-0.030	0.0105	84.63	2.051	-	-	-	-

Table 4.2: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0, -3)$. Cluster size $n = 3$.

		NMAR				MAR			
(N=100,n=3)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	0.134	0.2585	85.06	2.361	-1.355	0.5443	77.01	3.485
β_1	3	0.074	0.2342	88.47	2.523	0.348	0.2181	95.26	4.072
σ_b^2	1	0.236	0.4778	94.74	4.885	1.107	0.3159	75.58	4.609
τ_0	2	0.353	0.2059	96.55	4.395	-	-	-	-
τ_1	0	0.103	0.0594	90.05	0.881	-	-	-	-
τ_2	-3	-0.721	0.2664	97.34	2.878	-	-	-	-
(N=200,n=3)									
β_0	-2	0.263	0.2103	80.45	1.492	-1.424	0.6256	24.46	2.535
β_1	3	-0.112	0.1664	85.56	1.573	0.303	0.1840	96.93	2.663
σ_b^2	1	0.080	0.2258	88.60	2.664	0.558	0.2883	75.06	3.193
τ_0	2	0.629	0.2396	92.82	3.029	-	-	-	-
τ_1	0	0.114	0.0392	86.39	0.602	-	-	-	-
τ_2	-3	-0.842	0.2812	88.68	1.981	-	-	-	-
(N=300,n=3)									
β_0	-2	0.328	0.2116	69.32	1.171	-1.432	0.6962	3.31	2.082
β_1	3	-0.187	0.1425	80.24	1.192	0.226	0.1574	94.81	2.021
σ_b^2	1	-0.009	0.2199	81.49	1.976	0.507	0.2999	72.60	2.479
τ_0	2	0.806	0.2742	90.35	2.604	-	-	-	-
τ_1	0	0.108	0.0276	84.09	0.484	-	-	-	-
τ_2	-3	-0.9065	0.3409	81.42	1.687	-	-	-	-

Table 4.3: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0.5, -3)$. Cluster size $n = 2$.

		NMAR				MAR			
(N=100,n=2)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	-0.291	0.2557	90.82	3.723	-1.242	0.4755	85.55	4.160
β_1	3	0.591	0.2633	92.23	4.304	0.579	0.1894	97.80	5.950
σ_b^2	1	1.734	0.3172	92.83	10.71	1.253	0.3960	86.08	6.908
τ_0	2	0.013	0.2204	89.18	2.881	-	-	-	-
τ_1	0.5	0.493	0.0895	91.22	1.106	-	-	-	-
τ_2	-3	0.158	0.2836	94.27	2.525	-	-	-	-
(N=200,n=2)									
β_0	-2	-0.159	0.2053	87.43	2.403	-1.388	0.5490	72.23	3.342
β_1	3	0.312	0.2123	90.27	2.647	0.671	0.2074	98.95	4.270
σ_b^2	1	0.867	0.2870	89.33	5.907	1.163	0.3376	73.71	4.785
τ_0	2	0.115	0.2083	92.99	2.410	-	-	-	-
τ_1	0.5	0.492	0.0445	94.47	0.804	-	-	-	-
τ_2	-3	-0.062	0.2728	92.83	2.068	-	-	-	-
(N=300,n=2)									
β_0	-2	-0.146	0.1774	86.58	1.890	-1.425	0.6064	48.47	2.843
β_1	3	0.203	0.1760	88.91	2.046	0.630	0.1798	98.40	3.480
σ_b^2	1	0.533	0.2571	91.35	4.421	0.787	0.2973	77.51	3.706
τ_0	2	0.112	0.1466	93.80	1.808	-	-	-	-
τ_1	0.5	0.507	0.0327	90.63	0.650	-	-	-	-
τ_2	-3	-0.108	0.2244	90.78	1.521	-	-	-	-

Table 4.4: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0.5, -3)$. Cluster size $n = 3$.

		NMAR				MAR			
(N=100,n=3)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	0.049	0.2361	87.00	2.371	-1.422	0.5259	70.68	3.252
β_1	3	0.107	0.2227	88.83	4.923	0.652	0.2016	98.97	3.886
σ_b^2	1	0.457	0.2810	94.42	4.923	0.971	0.3033	75.67	4.175
τ_0	2	0.320	0.2013	96.17	2.856	-	-	-	-
τ_1	0.5	0.560	0.0549	95.44	0.849	-	-	-	-
τ_2	-3	-0.377	0.2813	94.85	2.217	-	-	-	-
(N=200,n=3)									
β_0	-2	0.167	0.1794	82.61	1.486	-1.421	0.6215	14.22	2.325
β_1	3	-0.072	0.1725	83.64	1.591	0.490	0.2142	98.99	2.402
σ_b^2	1	0.377	0.2407	89.63	2.860	0.580	0.3067	75.95	2.546
τ_0	2	0.430	0.2095	96.11	1.912	-	-	-	-
τ_1	0.5	0.572	0.0295	92.63	0.595	-	-	-	-
τ_2	-3	-0.564	0.2242	90.09	1.506	-	-	-	-
(N=300,n=3)									
β_0	-2	0.195	0.1540	76.66	1.145	-1.567	0.7333	1.405	1.874
β_1	3	-0.114	0.1426	81.42	1.183	0.416	0.1982	97.59	1.821
σ_b^2	1	0.051	0.2173	88.10	2.061	0.293	0.3021	78.66	2.062
τ_0	2	0.365	0.1912	91.36	1.466	-	-	-	-
τ_1	0.5	0.573	0.0198	92.63	0.476	-	-	-	-
τ_2	-3	-0.503	0.2573	83.66	1.167	-	-	-	-

Table 4.5: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0, -1)$. Cluster size $n = 2$.

		NMAR				MAR			
(N=100,n=2)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	-0.345	0.2615	95.75	3.285	-0.397	0.2203	95.49	2.991
β_1	3	0.535	0.2276	94.41	3.932	0.377	0.2300	92.08	3.911
σ_b^2	1	0.703	0.5777	96.42	8.691	0.600	0.3004	81.18	4.242
τ_0	2	0.241	0.2136	95.70	2.723	-	-	-	-
τ_1	0	-0.033	0.1210	88.13	1.451	-	-	-	-
τ_2	-1	0.113	0.3502	92.81	3.196	-	-	-	-
(N=200,n=2)									
β_0	-2	-0.183	0.1879	91.68	2.121	-0.361	0.1779	98.19	2.180
β_1	3	0.257	0.1837	92.81	2.440	0.221	0.1875	96.57	2.715
σ_b^2	1	0.434	0.3029	91.47	5.421	0.565	0.3182	75.58	3.221
τ_0	2	0.119	0.1815	90.78	2.302	-	-	-	-
τ_1	0	-0.0182	0.0900	89.21	1.087	-	-	-	-
τ_2	-1	-0.063	0.4171	89.76	2.560	-	-	-	-
(N=300,n=2)									
β_0	-2	-0.137	0.1611	91.21	1.704	-0.305	0.1678	97.19	1.629
β_1	3	0.182	0.1648	92.58	1.931	0.126	0.1639	94.78	1.981
σ_b^2	1	0.457	0.3001	83.68	4.501	0.281	0.2941	75.77	2.422
τ_0	2	0.151	0.1215	92.79	1.614	-	-	-	-
τ_1	0	-0.023	0.0644	92.41	0.902	-	-	-	-
τ_2	-1	-0.023	0.4920	92.55	1.980	-	-	-	-

Table 4.6: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0, -1)$. Cluster size $n = 3$.

		NMAR				MAR			
(N=100,n=3)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	0.006	0.2082	90.640	2.094	-0.302	0.2499	97.79	2.041
β_1	3	0.138	0.1950	92.36	2.316	0.131	0.1793	97.76	2.375
σ_b^2	1	0.522	0.2592	98.93	4.861	0.309	0.2690	81.86	2.411
τ_0	2	0.358	0.1978	97.21	2.831	-	-	-	-
τ_1	0	0.113	0.0953	90.84	1.080	-	-	-	-
τ_2	-1	-0.547	0.3470	96.93	2.691	-	-	-	-
(N=200,n=3)									
β_0	-2	0.092	0.1361	85.51	1.381	-0.263	0.1339	94.97	1.376
β_1	3	0.029	0.1400	86.75	1.460	0.064	0.1223	96.57	1.598
σ_b^2	1	0.207	0.2446	90.49	2.850	0.199	0.2739	77.70	1.709
τ_0	2	0.364	0.1731	98.65	1.958	-	-	-	-
τ_1	0	0.123	0.0605	92.60	0.744	-	-	-	-
τ_2	-1	-0.611	0.4346	91.23	1.831	-	-	-	-
(N=300,n=3)									
β_0	-2	0.149	0.1088	82.36	1.051	-0.238	0.1213	92.18	1.098
β_1	3	-0.052	0.1001	86.20	1.111	0.036	0.0997	96.19	1.271
σ_b^2	1	0.074	0.1856	86.13	2.035	0.067	0.2299	80.98	1.351
τ_0	2	0.330	0.1676	92.44	1.431	-	-	-	-
τ_1	0	0.130	0.0398	83.87	0.591	-	-	-	-
τ_2	-1	-0.582	0.4834	83.51	1.391	-	-	-	-

Table 4.7: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0.5, -1)$. Cluster size $n = 2$.

		NMAR				MAR			
(N=100,n=2)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	-0.323	0.2612	91.48	3.208	-0.404	0.2027	98.31	3.031
β_1	3	0.565	0.2579	93.97	3.904	0.386	0.2251	96.44	3.951
σ_b^2	1	0.294	0.4829	90.35	7.831	1.103	0.3132	74.28	4.291
τ_0	2	0.075	0.2027	92.50	2.852	-	-	-	-
τ_1	0.5	0.541	0.2034	86.29	1.786	-	-	-	-
τ_2	-1	0.066	0.3811	91.87	3.641	-	-	-	-
(N=200,n=2)									
β_0	-2	-0.215	0.1899	91.46	2.071	-0.668	0.1963	96.78	1.931
β_1	3	0.278	0.1865	91.05	2.357	0.330	0.1881	96.56	2.411
σ_b^2	1	0.811	0.2566	90.86	5.391	0.341	0.3234	80.94	2.907
τ_0	2	0.081	0.1597	89.54	21.94	-	-	-	-
τ_1	0.5	0.431	0.1487	87.62	1.410	-	-	-	-
τ_2	-1	0.137	0.3447	88.80	2.851	-	-	-	-
(N=300,n=2)									
β_0	-2	-0.121	0.1387	92.28	1.571	-0.312	0.1488	97.59	1.572
β_1	3	0.142	0.1326	91.32	1.772	0.257	0.1419	97.77	1.961
σ_b^2	1	0.425	0.2571	90.36	3.911	0.407	0.2836	77.82	2.331
τ_0	2	0.095	0.1376	91.79	1.611	-	-	-	-
τ_1	0.5	0.458	0.1038	89.06	1.151	-	-	-	-
τ_2	-1	0.044	0.4379	91.95	2.318	-	-	-	-

.

Table 4.8: Empirical biases, mean squared errors, coverage probabilities, and average lengths of 95% confidence intervals for maximum likelihood estimators in binary logistic regression models with missing responses under different sample sizes. Data were generated from the NMAR model and fitted by two models, correctly specified NMAR and misspecified MAR. Parameters are fixed at $(\beta_0, \beta_1, \sigma_b^2, \tau_0, \tau_1, \tau_2) = (-2, 3, 1, 2, 0.5, -1)$. Cluster size $n = 3$.

		NMAR				MAR			
(N=100,n=3)	TRUE	BIAS	MSE	CP	AL	BIAS	MSE	CP	AL
β_0	-2	0.001	0.2048	90.12	2.092	-0.294	0.1707	96.79	1.951
β_1	3	0.144	0.2108	91.13	2.311	0.213	0.1710	97.79	2.306
σ_b^2	1	0.603	0.2545	93.75	4.771	0.449	0.3099	76.92	2.542
τ_0	2	0.280	0.1831	96.05	2.531	-	-	-	-
τ_1	0.5	0.614	0.1699	89.87	1.418	-	-	-	-
τ_2	-1	-0.431	0.3593	91.64	2.961	-	-	-	-
(N=200,n=3)									
β_0	-2	0.065	0.1372	87.05	1.391	-0.251	0.1405	94.58	1.335
β_1	3	0.021	0.1483	90.33	1.468	0.132	0.1243	96.59	1.541
σ_b^2	1	0.214	0.2378	89.04	2.702	0.121	0.2495	82.71	1.691
τ_0	2	0.261	0.1543	94.74	1.678	-	-	-	-
τ_1	0.5	0.631	0.0922	87.66	1.027	-	-	-	-
τ_2	-1	-0.458	0.4411	86.93	2.052	-	-	-	-
(N=300, n=3)									
β_0	-2	0.084	0.1080	83.87	1.081	-0.221	0.1091	92.18	1.071
β_1	3	-0.014	0.1066	88.83	1.121	0.105	0.0909	96.39	1.231
σ_b^2	1	0.132	0.1979	85.06	2.045	0.054	0.2069	81.21	1.251
τ_0	2	0.243	0.1416	92.39	1.314	-	-	-	-
τ_1	0.5	0.636	0.0701	84.16	0.816	-	-	-	-
τ_2	-1	-0.471	0.4864	82.68	1.611	-	-	-	-

4.7 Application: Analysis of Smoking Data

In the CARDIA (Coronary Artery Development in Young Adults) study conducted in four urban centers (Birmingham, AL; Chicago, IL; Minneapolis, MN; and Oakland, CA) across the United States, a total of 5,115 young adults ages 18-30 years were followed prospectively and examined up to five times from 1986 to 1996. Recruitment, restricted to blacks and whites, was carried out to achieve approximate balance in sample size with respect to age, race, gender, and education. Within targeted subgroups, participants were generally recruited by telephone, from selected census tracts, or from a health plan membership (Hughes et al. 1987).

Study participants were scheduled for visits at years 0, 2, 5, 7, and 10. We consider the first four visits and 5,078 (99.3%) young adults with self- reported smoking status (yes/no) known at baseline (year 0). Table 4.9 shows that, on average, young adults in the four groups were similar with respect to age, and that whites were more likely to have a higher level of attained education (at study end) than blacks.

Furthermore, CARDIA participants are considerably more educated than young adults in the general U.S. population. Of CARDIA participants born 1963-1967, 54.8% of whites and 19.4% of blacks attained at least a bachelor's degree. The corresponding 1992 U.S. Census Population survey rates for white and black young adults ages 25-29 are 25.0% and 11.3% (U.S. Bureau of the Census, 1998 a).

Table 4.9 shows that 7-year changes in smoking prevalence for the sample are -.1, -1.8, -4.7, and -6.8 in percentage points for black males, black females, white males, and white females, respectively, so that the observed smoking rates decreased over time for each group, although only slightly for blacks.

Table 4.9: Characteristics of study sample

	Race/Sex Group			
Variable	Black Males	Black Females	White Males	White Females
Number of participants	1,145	1,473	1,160	1,300
Age at year 0(mean years)	24.2	24.4	25.4	25.4
High school or less(%)	39.5	28.8	17.5	15.9
Some college (%)	42.1	51.0	29.5	27.6
College degree(%)	18.4	20.2	53.0	56.5
Year0(% smoke)	36.9	31.3	26.5	27.3
Year2(% smoke)	38.4	30.6	25.7	24.0
Year5(% smoke)	37.2	31.9	24.3	21.9
Year7(% smoke)	36.8	29.5	21.8	20.5

Table 4.10 shows that participants who were smokers at baseline were more likely to have subsequent missing visits than participants who were nonsmokers at baseline. Also note that blacks are more likely to have missing visits than whites.

Table 4.10: Occasion-specific sample sizes by baseline smoking status

Baseline smoker	Year exam	Black males	Black females	White males	White females
No	0	722(100%)	1,012(100%)	853(100%)	945(100%)
	2	616(85%)	906(90%)	806(95%)	901(95%)
	5	581(81%)	835(83%)	783(92%)	868(92%)
	7	535(74%)	790(78%)	752(88%)	816(86%)
Yes	0	423(100%)	461(100 %)	307(100%)	355(100%)
	2	356(84%)	374(81%)	285(93%)	320(90%)
	5	315(75%)	370(80%)	260(85%)	299(84%)
	7	285(67%)	338(73%)	246(80%)	280(79%)

4.7.1 Fitting Models

To find the association between the outcome of smoking status and other covariates, we use the logistic regression model

$$\text{logit}\{E(y_{ij}|b_i)\} = \beta_0 + \beta_1(\text{Age}|10)_i + \beta_2(\text{Time})_{ij}$$

$$+\beta_3(\text{Black female})_i + \beta_4(\text{White male})_i + \beta_5(\text{White female})_i + b_i,$$

for $i = 1, \ldots 5078$, $j = 1, 2, 3, 4$ and $\text{Time}_{ij} = 0, 2, 5, 7$ ($j = 1, 2, 3, 4$). The group "Black male" was considered a reference group in the model. The random effect b_i are assumed independent $N(0, \sigma_b^2)$.

The missing data model was chosen as

$$\text{logit}(\pi_{ij}) = \tau_0 + \tau_1(\text{Time})_{ij} + \tau_2 y_{ij} \ .$$

The R code for fitting the data is presented in the Appendix.

Table 4.11: Smoking data fitted by NMAR and MAR models

N=5078	NMAR				MAR			
n=4	Estimate	SE	Z-value	P-value	Estimate	SE	Z-value	P-value
$\hat{\beta}_0$	-1.924	0.0875	-21.98	3.847e-107	-3.6030	0.9680	-3.719	2.000e-04
$\hat{\beta}_1$	-0.144	0.0152	-9.487	2.366e-21	0.5071	0.9687	1.331	1.830e-01
$\hat{\beta}_2$	0.013	0.0124	1.112	2.659e-01	-0.0553	0.3811	-4.136	3.540e-05
$\hat{\beta}_3$	-0.592	0.1258	-4.708	2.494e-06	-1.5670	0.0133	-4.108	4.000e-05
$\hat{\beta}_4$	-0.750	0.2635	-2.849	4.382e-03	-3.1473	0.3814	-7.519	5.510e-14
$\hat{\beta}_5$	-1.244	0.1106	-11.24	2.521e-2	-3.1978	0.4185	-7.888	3.110e-15
$\hat{\sigma_b}$	5.537	0.1316	42.07	0.000e-0	7.786	0.1733	44.92	0.000e-0
$\hat{\tau_0}$	2.970	0.0800	37.11	1.864e-301	-	-	-	-
$\hat{\tau_1}$	-0.172	0.0118	-14.47	1.774e-47	-	-	-	-
$\hat{\tau_2}$	-1.031	0.0833	-12.37	3.519e-35	-	-	-	-

Table 4.11 presents the maximum likelihood estimates of regression coefficients and the variance component obtained under both NMAR and MAR models. The groups black female, white male, and white fmale are found to be highly significant by both NMAR and MAR methods. But the predictor age is significant at the .05 level only by the NMAR method, whereas the predictor time is is significant at the .05 level only by the MAR method. The p values of the test $H_0 : \sigma_b = 0$ vs. $H_1 : \sigma_b > 0$, indicate that the standard deviation component σ_b is significant at a level close to 0 by both methods.

Chapter 5

Conclusion

In this thesis, we have studied the generalized linear mixed model with missing responses for analyzing binary longitudinal data. We particularly considered joint estimation of the regression parameters and the association parameters by using the maximum likelihood method. We studied a set of maximum likelihood estimating equations for fitting regression models to binary data. We used the Newton-Raphson algorithm to estimate the parameters for logistics regression. Our simulation study demonstrates that the missing data mechanism generally provides unbiased and efficient estimators when it follows correctly specified models under the NMAR method for binary data.

The purpose of the thesis was to study the empirical properties of the maximum likelihood estimators for assessing the significance of regression and nuisance parameters at a given level of significance. We study empirical biases, MSEs, average lengths and empirical coverage probabilities of the estimators of model parameters under both correctly specified and misspecified models, when the true distribution is the binomial distribution.

It is apparent that when we use the NMAR model to fit the data, the models provide unbiased and efficient estimators for binary regression. From the numerical study, we conclude

that when analyzing incomplete data with missing responses, it is necessary to incorporate a suitable missing data model into the observed data likelihood function in order to obtain unbiased and efficient estimators of the model parameters. We also note that a misspecified missing data model can provide systematic bias in the maximum likelihood estimation. So it is important to assess the validity of a missing data model when performing a likelihood inference based on the given observed data.

5.1 Future Research

There are many current and future research considerations for missing data problems when the mechanism is nonignorable. In this thesis, we worked with binary mixed models as the missingness were only occurred in the responses. Our future research will be concentrated on studying Poisson data with nonignorable missingness in responses and covariates. However, it is well-known that the ordinary ML estimators are sensitive to extreme observations or outliers, so we will make more researches on robust methods, which are useful for down-weighting any influential observations in the data when estimating the model parameters, Sinha (2012).

Bibliography

[1] Ibrahim, J. G., Chen, M. H. and Lipsitz, S. R. (1999). Monte Carlo EM for missing covariates in parametric regression models. *Journal of the Royal Statistical Society*, 61, 173-190.

[2] Little, R. J. (1982). Models for non-response in sample surveys. *Journal of the American Statistical Association*, 77, 237-249.

[3] Baker, S. G. and Laird, N. M. (1982). Regression analysis for categorical variables with outcome subject to nonignorable nonresponse. *Journal of the American Statistical Association*, 83, 62-69.

[4] Preisser, J. N., Galecki, A. T., Lohman, K. K. and Wagenknecht, L. E. (2000). Analysis of smoking trends with incomplete longitudinal binary responses. *Journal of the American Statistical Association*, 95, 1021-1031.

[5] Little, R. J. and Rubin, D. B. (1991). Statistical analysis with missing data. *Journal of Educational Statistics*, 16, 150-155.

[6] Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63, 581-592.

[7] Little, R. J. (1995). Modeling the drop-out mechanism in repeated-measures studies. *Journal of the American Statistical Association*, 90, 1112-1121.

[8] Patterson, H. D. and Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58, 545-554.

[9] Stiratelli, R., Laird, N. and Ware, J. H. (1984). Random effects models for serial observations with binary response . *Biometrics*, 40, 961-971.

[10] Zeger, S. L., Liang, K. Y. and Albert, P.S. (1988). Models for longitudinal data: A generalized estimating equation approach . *Biometrics*, 44, 1049-1060.

[11] Chib, S. and Albert, E. G. (1995). Understanding the Metropolis-Hastings algorithm . *The American Statistician*, 49, 327-335.

[12] Levine, R. A. and Casella, G. (2001). Implementations of the Monte Carlo EM algorithm . *Journal of Computational and Graphical Statistics*, 10, 422-439.

[13] Preisser, J. S. and Qaqish, B. F. (1999). Robust regression for clustered data with application to binary responses. *Biometrics*, 55, 574-579.

[14] Greenlees, W. S., Reece, J. S. and Zieschang, K. D. (1982). Imputation of missing values when the probability of response depends on the variable being imputed. *Journal of the American Statistical Association*, 77, 251-261.

[15] Ibrahim, J. G. and Lipsitz, S. R. (1999). Missing covariates in generalized linear models when the missing data mechanism is nonignorable. *Journal of the Royal Statistical Society*, 61, 173-190.

[16] Sinha, S. K. (2008). Robust methods for generalized linear models with nonignorable missing covariates. *The Canadian Journal of Statistics*, 36, 277-299.

[17] Tang, G., Little, R. J. and Raghunathan, T. E. (2003). Analysis of multivariate missing data with nonignorable nonresponse. *Biometrika*, 90, 747-764.

[18] Sinha, S. K. (2004). Robust analysis of generalized linear mixed . *Journal of the American Statistical Association*, 466, 451-460.

[19] Ibrahim, J. G., Hui, M. C. and Lipsitz, S. R. (2001). Missing response in generalized linear mixed models when the missing data mechanism is non ignorable. *Biometrika*, 61, 173-190.

[20] McCulloch, C. E., Searle, S. R. and Neuhaus, J. M. (2008). Generalized linear and mixed models (2nd edition). *Wiley and Sons, Inc*, New Jersey.

[21] Casella, G. and Berger, R. L. (1990). Statistical inference . *Duxbury Press Belmont*, CA.

[22] Wei, G. C. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor mans data augmentation algorithms. *Journal of the American Statistical Association,* 85, 699-704.

[23] Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika,* 73, 13-22.

[24] Greenlees, W. S., Reece, J. S. and Zieschang, K. D. (1982). Imputation of missing values when the probability of response depends on the variable being imputed. *Journal of the American Statistical Association,* 77, 251-261.

[25] Baker, S. G. and Laird, N. M. (1988). Regression analysis for categorical variables with outcome subject to nonresponse. Journal of the American Statistical Association, 83, 62-49.

[26] Sinha, S. K. (2011). Robust analysis of longitudinal data with nonignorable missing responses, *Metrika*, 75, 913-938.

[27] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, 39, 1-38.

[28] Little, R. J. and Rubin, D. B. (2002). Statistical analysis with missing data, *Second Edition*, New York.

[29] Diggle, P. and Kenward, M. G. (1994). Informative drop-out in longitudinal data analysis, *Journal of the Royal Statistical Society*, 43, 49-93.

APPENDIX

A: R Codes for Chapter 2

A-1\\

R-code:\\

```
 o_ring <-read.table("F:\\challenger_data.txt",header=T)
View(o_ring)
o_ring.model<-glm(fail~temp,family=binomial,data=o_ring)
Call:
glm(formula = fail ~ temp, family = binomial, data = o_ring)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2125  -0.8253  -0.4706   0.5907   2.0512

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) 10.87535    5.70291   1.907   0.0565 .
temp        -0.17132    0.08344  -2.053   0.0400 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28.975  on 23  degrees of freedom
Residual deviance: 23.030  on 22  degrees of freedom
AIC: 27.03

Number of Fisher Scoring iterations: 4




# Logistic Regression
# where F is a binary factor and
# x1 are continuous predictors

summary(o_ring.model) # display results.
-------------------------------------------------
----------------------#------------------------
A-2:\\

R-Code:\\

logitfit1<-function(iter)
```

```
# FITTING OF LOGISTIC MODEL (FULL MODEL)
# MODEL: theta = exp(beta0+beta1*x)/(1+exp(beta0+beta1*x))
{
x<-c(53,56,57,63,66,67,67,67,68,69
,70,70,70,70,72,73,75,75,76,76,78,79,80,81)

y<-c(1,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,1,0
,0,0,0,0,0)

xx<-cbind(rep(1,length(x)),x)

n=1

thetabar<-c(0.29,0.29,0.29,0.29,0.29,
0.29,0.29,0.29,0.29,0.29,0.29,0.29,0.29,0.29,
 0.29,0.29,0.29,0.29,0.29,0.29,0.29,0.29,
            0.29,0.29)

    # Transform the model to a linear one
# so as to get starting values:
logits<-log(thetabar/(1-thetabar))
weights<-24*thetabar*(1-thetabar)
ls.fit<-lm(logits~x,weights=weights)
beta0<-ls.fit$coef
beta0
# Now beta0 is the vector of starting values
# The iterative procedure is described below
```

```
beta<-NULL
for(i in 1:iter)
{
eta0<-xx%*% beta0
theta0<-exp(eta0)/(1+exp(eta0))
logits<-log(theta0/(1-theta0))
logits
z<-logits+(y-n*theta0)/(n*theta0*(1-theta0))
z
weights<-c(n*theta0*(1-theta0))
weights
ls.fit<-lm(z~x,weights=weights)
ls.fit
beta0<-ls.fit$coef
beta<-cbind(beta,beta0)
beta
}
 dev<-2*sum(y*log(y/(n*theta0))+(n-y)*log((n-y)/(n-n*theta0)))
  list(beta=beta,deviance=dev)
  }



plot(x,y,xlab="temp",ylab="failure")
g=glm(y~x,family=binomial,dat)

curve(predict(g,data.frame(x=x),type="resp"),add=TRUE)
```

```
points(x,fitted(g),pch=20)
```

```
  (Intercept)             x
-8.953840e-01 -3.784234e-18
```

```
           beta beta0      beta0      beta0      beta0      beta0      beta0
(Intercept) ?    9.533426   10.70831   10.87331   10.87535   10.87535   10.87535
x           ?    -0.1490448 -0.1686703 -0.1712888 -0.1713205 -0.1713205 -0.1713205
            beta0
(Intercept) 10.87535
x           -0.1713205
```

A-3:\\

R-Code:\\

```
x<-c(53,56,57,63,66,67,67,67,68,69
     ,70,70,70,70,72,73,75,75,76,76,78,79,80,81)

y<-c(1,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,1,0
     ,0,0,0,0,0)
logits<-log(thetabar/(1-thetabar))
weights<-24*thetabar*(1-thetabar)
ls.fit<-lm(logits~x,weights=weights)
beta0<-ls.fit$coef
```

```
beta0

out<-glm(y ~x, family = quasi(variance = "mu(1-mu)", link = "logit"),
start = beta0)



summary(out)
Call:
glm(formula = y ~ x, family = quasi(variance = "mu(1-mu)", link = "logit"),
    start = beta0)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2125  -0.8253  -0.4706   0.5907   2.0512

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.87535    5.79981   1.875   0.0741 .
x           -0.17132    0.08486  -2.019   0.0558 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasi family taken to be 1.034419)

    Null deviance: 28.975  on 23  degrees of freedom
Residual deviance: 23.030  on 22  degrees of freedom
AIC: NA
```

```
Number of Fisher Scoring iterations: 4



-----------------------------------------------

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


A-4:\\

R code:\\

\begin{verbatim}



x1<-c(50,230,125,75,70,65,65,350,350,160,145,145,180,
43,42,42,45,83,300,190,145,510,65,470,300,275,420,65,
      40,900,95,40,140,150,80,80,145,100,150,150,210,11,100,50)
x2<-c(70,65,70,65,65,70,60,60,90,80,65,85,70,
80,85,85,85,85,65,90,90,80,75,90,80,90,50,
      80,75,90,88,85,90,50,60,85,65,65,80,80,75,75,65,88)
x3<-c(52,42,45,68,53,46,62,54,54,38,38,38,42,40,51,51,42,48,68,84,54,57,68,90,
      165,40,44,48,51,48,36,57,38,44,96,96,72,72,48,48,42,42,60,60)
x4<-c(1,6,1,0.5,0.5,3,1,0.5,0.5,0,10,0,2,0,12,0,0,10,10,6,12,
      10,5,9,9,4,17,15,15,35,20,10,7,5,5,5,9,9,3,0,2,0,25,20)
y<-c(2,1,0,4,1,2,0,0,4,4,1,4,1,5,2,5,5,5,0,5,
     1,1,3,3,2,2,0,1,5,2,3,3,3,0,0,2,0,0,3,2,3,5,0,3)
```

```
mine<-glm(y~x1+x2+x3+x4,family=poisson)
mine
summary(mine)

Call:
glm(formula = y ~ x1 + x2 + x3 + x4, family = poisson)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.78962  -0.85988  -0.04893   0.37313   2.16201

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.5930896  1.0256803  -3.503  0.00046 ***
x1          -0.0014066  0.0008358  -1.683  0.09240 .
x2           0.0623458  0.0122862   5.074 3.89e-07 ***
x3          -0.0020803  0.0050661  -0.411  0.68134
x4          -0.0308135  0.0162648  -1.894  0.05816 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 74.984  on 43  degrees of freedom
Residual deviance: 37.856  on 39  degrees of freedom
AIC: 144.13
```

```
Number of Fisher Scoring iterations: 5
-----------------------------------------------
----------------------#------------------------


A-5:\\
R code:\\


loglink<-function(iter)
  # FITTING OF LOGISTIC MODEL
{


  x1<-c(50,230,125,75,70,65
  ,65,350,350,160,145,
  145,180,43,42,42,45,83,
  300,190,145,510,65,470,300,275,420,65,
        40,900,95,40,140,150,80,80,145,100,150,150,210,11,100,50)
  x2<-c(70,65,70,65,65,70,60,60,90,80,65,85,70 ,80,85,85,85,85,65,90,90,80,75
 ,90 ,80,90,50,80,75,90,88,85,90,50 ,
  60,85,65,65,80,
 80,75,75,65,88)
  x3<-c(52,42,45,68,53,46,62,54,54,38,38,38,42,40,51,51,42,48,68,84,54,57,68,90,
        165,40,44,48,51,48,36,57,38,44,96,96,72,72,48,48,42,42,60,60)
  x4<-c(1,6,1,0.5,0.5,3,1,0.5,0.5,0,10,0,2,0,12,0,0,10,10,6,12,
        10,5,9,9,4,17,15,15,35,20,10,7,5,5,5,9,9,3,0,2,0,25,20)
  y<-c(2,1,0,4,1,2,0,0,4,4,1,4,1,5,2,5,5,5,0,5,
       1,1,3,3,2,2,0,1,5,2,3,3,3,0,0,2,0,0,3,2,3,5,0,3)
```

```
xx<-cbind(rep(1,length(x1)),x1,x2,x3,x4)
xx
mean(y)
thetabar<-rep(mean(y),44)
thetabar
# Transform the model to a linear one
# so as to get starting values:
log<-log(thetabar)
weights<-thetabar
ls.fit<-lm(log~x1+x2+x3+x4,weights=weights)
beta0<-ls.fit$coef
beta0

# Now beta0 is the vector of starting values
# The iterative procedure is described below
beta<-NULL
for(i in 1:iter)
{
  eta0<-xx%*%beta0
  theta0<-exp(eta0)
  log<-log(theta0)
  z<-log+(y-theta0)/theta0
  weights<-c(theta0)
  ls.fit<-lm(z~x1+x2+x3+x4,weights=weights)
  beta0<-ls.fit$coef
```

```
    beta<-cbind(beta,beta0)
    beta
  }

}
fit<-loglink(iter=7)
```

R-outpu:

```
   beta beta0         beta0        beta0        beta0        beta0
(Intercept)    -2.255497     -3.329721    -3.583492    -3.593075    -3.59309
x1            -0.0008513658 -0.00126572  -0.00139955  -0.001406573 -0.001406588
x2             0.04695147    0.05912732   0.06222515   0.06234558   0.06234576
x3             -0.003588549  -0.002394832 -0.002093906 -0.00208037  -0.002080342
x4            -0.02256136   -0.02883411  -0.03072087  -0.0308133   -0.03081349
            beta0        beta0        beta0        beta0        beta0
(Intercept) -3.59309     -3.59309     -3.59309     -3.59309     -3.59309
x1          -0.001406588 -0.001406588 -0.001406588 -0.001406588 -0.001406588
x2          0.06234576   0.06234576   0.06234576   0.06234576   0.06234576
x3          -0.002080342 -0.002080342 -0.002080342 -0.002080342 -0.002080342
x4          -0.03081349  -0.03081349  -0.03081349  -0.03081349  -0.03081349
```

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

A-6:\\

R code:\\

```
x1<-c(50,230,125,75,70,65,65,350,350,160,145,145,
180,43,42,42,45,83,300,190,145,510,65,470,300,275,420,65,
      40,900,95,40,140,150,80,80,145,100,150,150,210,11,100,50)
x2<-c(70,65,70,65,65,70,60,60,90,80,65,85,70,80,
85,85,85,85,65,90,90,80,75,90,80,90,50,
      80,75,90,88,85,90,50,60,85,65,65,80,80,75,75,65,88)
x3<-c(52,42,45,68,53,46,62,54,54,38,38,38,42,40,51,51,42,48,68,84,54,57,68,90,
      165,40,44,48,51,48,36,57,38,44,96,96,72,72,48,48,42,42,60,60)
x4<-c(1,6,1,0.5,0.5,3,1,0.5,0.5,0,10,0,2,0,12,0,0,10,10,6,12,
      10,5,9,9,4,17,15,15,35,20,10,7,5,5,5,9,9,3,0,2,0,25,20)
y<-c(2,1,0,4,1,2,0,0,4,4,1,4,1,5,2,5,5,5,0,5,
     1,1,3,3,2,2,0,1,5,2,3,3,3,0,0,2,0,0,3,2,3,5,0,3)
x<-cbind(rep(1,length(x1)),x1,x2,x3,x4)

mean(y)
thetabar<-rep(mean(y),44)
thetabar
# Transform the model to a linear one
# so as to get starting values:
log<-log(thetabar)
weights<-thetabar
ls.fit<-lm(log~x1+x2+x3+x4,weights=weights)
beta0<-ls.fit$coef
beta0
```

```
mine<-glm(y ~x1+x2+x3+x4, family = quasi(variance = "mu", link = "log")

mine
summary(mine)

Call:
glm(formula = y ~ x1 + x2 + x3 + x4, family = quasi(variance = "mu",
    link = "log"), start = beta0)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.78962  -0.85988  -0.04893   0.37313   2.16201

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.5930896  0.9837105  -3.653 0.000762 ***
x1          -0.0014066  0.0008016  -1.755 0.087165 .
x2           0.0623458  0.0117835   5.291 4.98e-06 ***
x3          -0.0020803  0.0048588  -0.428 0.670892
x4          -0.0308135  0.0155992  -1.975 0.055339 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasi family taken to be 0.9198261)

    Null deviance: 74.984  on 43  degrees of freedom
```

```
Residual deviance: 37.856  on 39  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
-----------------------------------------------
----------------------#-----------------------

Appendix:A-7:
x1<-c(50,230,125,75,70,65,65,350,350,160,145,145,180,
43,42,42,45,83,300,190,145,510,65,470,300,275,420,65,
      40,900,95,40,140,150,80,80,145,100,150,150,210,11,100,50)
x2<-c(70,65,70,65,65,70,60,60,90,80,65,85,70,
80,85,85,85,85,65,90,90,80,75,90,80,90,50,
      80,75,90,88,85,90,50,60,85,65,65,80,80,75,75,65,88)
x3<-c(52,42,45,68,53,46,62,54,54,38,38,38,42,40,51,51,42,48,68,84,54,57,68,90,
      165,40,44,48,51,48,36,57,38,44,96,96,72,72,48,48,42,42,60,60)
x4<-c(1,6,1,0.5,0.5,3,1,0.5,0.5,0,10,0,2,0,12,0,0,10,10,6,12,
      10,5,9,9,4,17,15,15,35,20,10,7,5,5,5,9,9,3,0,2,0,25,20)
y<-c(2,1,0,4,1,2,0,0,4,4,1,4,1,5,2,5,5,5,0,5,
     1,1,3,3,2,2,0,1,5,2,3,3,3,0,0,2,0,0,3,2,3,5,0,3)

mine<-glm(y~x1+x2+x3+x4,family=poisson)
mine
summary(mine)
R-outpu:\\\\

> summary(mine)
```

```
Call:
glm(formula = y ~ x1 + x2 + x4, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7727  -0.9073  -0.0107   0.2716   2.1783

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.7206821  0.9788770  -3.801 0.000144 ***
x1          -0.0014793  0.0008244  -1.794 0.072757 .
x2           0.0627011  0.0122711   5.110 3.23e-07 ***
x4          -0.0316514  0.0163095  -1.941 0.052298 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 74.984  on 43  degrees of freedom
Residual deviance: 38.031  on 40  degrees of freedom
AIC: 142.3

Number of Fisher Scoring iterations: 5.
-----------------------------------------------
----------------------#------------------------
```

B: R Codes for Chapter 3

```
B-1:\\
R code\\

library (glmmML)

beta<-NULL
se<-NULL
sigma<-NULL
std.sigma<-NULL
for(i in 1:1000)
{

  binary.dat <- function(k=100, n=4, beta0=-1, beta1=1, sigmasq=0.25)
  {
    u1 <- rnorm(k, 0, sqrt(sigmasq))
    u1

    u <- rep(u1, rep(n, k))
    id <- rep(c(1:k), rep(n, k))
```

```
    k1 <- n * k
    x <- round(rnorm(k1, 0, 1),digits=2)
    x

    eta <- beta0 + beta1 * x + u
    p <- round(exp(eta)/(1+exp(eta)), digits=3)

    y <- rbinom(k1, 1, p)

    dat <- cbind(id, x, y, p, u)
    dat
  }
  dat<-data.frame(binary.dat(100))
  dat

  fit <- glmmML(y ~ x, data = dat, family=binomial, cluster=id,
                method="ghq", n.points = 25)
  beta<-c(beta,coef(fit))
  se<-c(se,fit$coef.sd)
  sigma0<-fit$sigma
  sigma0
  sigma<-c(sigma,sigma0)
  std.sigma<-c(std.sigma,fit$sigma.sd)
  std.sigma


}
#thelast (summary fit) between 1000 summary fit's.
summary(fit)
```

```
#calculate the estimators(beta0 ,beta1) for 1000 datasets.
beta<-data.frame(beta)
beta<-as.matrix(beta)

#calculate standarad errors for 1000 datasets.
se<-data.frame(se)
se<-as.matrix(se)

#calculate the mean of standarad error of beta0.and beta1
even_indexes<-seq(0,2000,2)
odd_indexes<-seq(1,1999,2)
std.beta0 <- se[odd_indexes,1]

mean.std.beta0<-c(mean(std.beta0,na.rm=TRUE))

std.beta1<- se[even_indexes,1]

mean.std.beta1<-c(mean(std.beta1))

#calculate the estimation of sigma.sq.
sigma<-data.frame(sigma)
sigma<-as.matrix(sigma)

write.csv(sigma)
mean.sigma<-c(mean(sigma))


#calculate the bias of the estimation of sigma.sq.
bias.sigma<-mean(sigma)-0.5
```

```
#calculate the average standarad errors of the estimate of the sigma square .
mean.std.sigma<-c(mean(std.sigma,na.rm=TRUE))


#calculate the mean square error of the estimate of the sigma square.
MSE.sigma=mean((sigma-0.5)^2)

#calculate the bias of estimators.
even_indexes<-seq(0,2000,2)
odd_indexes<-seq(1,1999,2)
beta0 <- beta[odd_indexes,1]
write.csv(beta0)
beta1<- beta[even_indexes,1]
write.csv(beta1)
beta<-cbind.data.frame(beta0,beta1)
mean.beta0<-c(mean(beta0))
mean.beta1<-c(mean(beta1))
bias.beta0<-mean(beta0)+1
bias.beta1<-mean(beta1)-1
#calculate the mean squared errors of estimators.
MSE.beta0=mean((beta0+1)^2)
MSE.beta1=mean((beta1-1)^2)
#calculate the C.I,95% of (beta0,beta1)
up.conf=beta0+qt(1-0.05/2, 338)*se
lo.conf=beta0-qt(1-0.05/2, 338)*se
conf.in.beta0<-cbind(lo.conf,up.conf)
write.csv(conf.in.beta0)
up.conf=beta1+qt(1-0.05/2, 338)*se
lo.conf=beta1-qt(1-0.05/2, 338)*se
```

```
conf.in.beta1<-cbind(lo.conf,up.conf)
write.csv(conf.in.beta1)

#calculate the C.I,95% of (sigmasq)

up.conf.sigma=sigma+1.96*std.sigma
lo.conf.sigma=sigma-1.96*std.sigma
conf.in.sigma<-cbind(lo.conf.sigma,up.conf.sigma)

write.csv(conf.in.sigma)

#from excell file:
coverage.prob.beta0<-c(0.946)
coverage.prob.beta1<-c(0.95)
coverage.prob.sigma<-c(0.974)
av.ci.length.beta0<-c(0.58782334)
av.ci.length.beta1<-c(0.541923609)
av.ci.length.sigma<-c(0.958709778)

table<-cbind(mean.beta0,mean.beta1,mean.
sigma,bias.beta0,bias.beta1,
bias.sigma,mean.std.beta0,
  mean.std.beta1,mean.std.sigma,
  MSE.beta0,MSE.beta1,MSE.sigma,
  coverage.prob.beta0,
  coverage.prob.beta1,coverage.prob.sigma,  av.ci.length.beta0,
  av.ci.length.beta1,
  av.ci.length.sigma)
t(table)
```

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

B-2:\\
R code:\\

guidedata <- read.table("C:\\Users\\r\\Desktop\\Thesis\\GuideData.txt",header=T)
summary(guidedata)

simpson <- function(vec, a, b)
{
  # used for numerical integreation
  # uses Simpson's rule; function values are elements of vec
  # length of vec should be odd
  n <- length(vec)
  m <- n - 1      # m is the number of intervals
  h <- (b - a)/m  # h is the length of each interval
  d <- c(1, rep(c(4, 2), m/2 - 1), 4, 1)
  integral <- (h/3) * sum(d * vec)
  list(integral = integral)
}



guidefit.mle <- function(dat=guidedata, beta0=c(-3.52,-.91,-.78,.47,.96,.115),
                         sigmasq0 = 1.218, iter=10)
{
  cat("Fitting GLMM to the GUIDE data.\n")
  cat("Here guidedata is a data.frame.\n")
```

```
yy <- dat[, "bothered"]
xx <- dat[, c("female", "age", "dayacc", "severe", "toilet")]

k <- 38
ID <- dat[, "pract_id"]
id0 <- c(1:k)
id <- rep(id0, table(ID))

data0 <- cbind(id, yy, xx)
a1 <- -10 # lower limit of u
a2 <-  10 # upper limit of u

u <- seq(a1, a2, length=201)

beta <- NULL
sigmasq <- NULL

for(m in 1:iter)
{
  cat(".")

  q <- 0
  M <- 0
  Q <- 0
  QQ <- 0

  q1 <- 0

  dbeta.dsig <- 0
```

```
db.sig <- 0

f.u <- dnorm(u, 0, sqrt(sigmasq0))

Eusq <- NULL
Eu4 <- NULL

for ( j in 1:k)
{
  y <- data0[data0[,"id"]==j, "yy"]
  n <- length(y)

  x <- data0[data0[,"id"]==j, c("female", "age", "dayacc", "severe",
   "toilet")]
  intercept <- rep(1,n)
  X <- as.matrix(cbind(intercept, x))

  # ..............................................
  # calculating conditional means w.r.t u|y
  # ..............................................

  f.y <- 1

  for (i in 1:n)
  {
    eta <- c(t(X[i,]) %*% beta0) + u
    mu.u <- exp(eta)/(1+exp(eta))
    vmu.u <- mu.u * (1-mu.u)
```

```
  f.y0 <- mu.u^y[i] * (1-mu.u)^(1-y[i])
  f.y <- f.y * f.y0
}

mu <- NULL
vmu <- NULL

vvmu <- matrix(c(0),n,n)

delbeta.sig <- NULL

for (i in 1:n)
{
  eta <- c(t(X[i,]) %*% beta0) + u
  mu.u <- exp(eta)/(1+exp(eta))
  vmu.u <- mu.u * (1-mu.u)

  fy.fu <- simpson(f.y * f.u, a1, a2)$integral
  mu.fy.fu <- simpson(mu.u * f.y * f.u, a1, a2)$integral
  vmu.fy.fu <- simpson(vmu.u * f.y * f.u, a1, a2)$integral

  mu0 <- c(mu.fy.fu/fy.fu)
  mu <- c(mu,mu0)

  vmu0 <- c(vmu.fy.fu/fy.fu)
  vmu <- c(vmu,vmu0)

  for (i1 in 1:n)
  {
```

```
    eta1 <- c(t(X[i1,]) %*% beta0) + u
    mu.u1 <- exp(eta1)/(1+exp(eta1))

    ymusq.fy.fu <- simpson((y[i]-mu.u)*(y[i1]-mu.u1)*f.y*f.u,a1,a2)$integral
    vvmu0 <- c(ymusq.fy.fu/fy.fu)
    vvmu[i,i1] <- vvmu0

  }

  delbeta.sig00 <- (y[i]-mu.u) * (-1/(2*sigmasq0) + 1/(2*sigmasq0^2) * u^2)
  delbeta.sig10 <- simpson(delbeta.sig00 * f.y * f.u, a1, a2)$integral
  delbeta.sig0 <- c(delbeta.sig10/fy.fu)
  delbeta.sig <- c(delbeta.sig, delbeta.sig0)
}

usq.fy.fu <- simpson(c(u^2) * f.y * f.u, a1, a2)$integral

db.sig0 <- t(X) %*% delbeta.sig
db.sig <- db.sig + db.sig0


Eusq0 <- c(usq.fy.fu/fy.fu)
Eusq <- c(Eusq, Eusq0)

u4.fy.fu <- simpson(c(u^4) * f.y * f.u, a1, a2)$integral

Eu40 <- c(u4.fy.fu/fy.fu)
Eu4 <- c(Eu4, Eu40)
```

```
  #.............................................

  M0 <- t(X) %*% diag(vmu,n,n) %*% X
  q0 <- t(X) %*% (y-mu)

  M <- M + M0
  q <- q + q0

  Q0 <- t(X) %*% ((y-mu)%*%t(y-mu)) %*% X
  Q <- Q + Q0

  QQ0 <- t(X) %*% vvmu %*% X
  QQ <- QQ + QQ0

  q10 <- -1/(2*sigmasq0) + 1/(2*sigmasq0^2) * Eusq0
  dbeta.dsig0 <- q0 * q10
  dbeta.dsig <- dbeta.dsig + dbeta.dsig0

  q1 <- q1 + q10
}

EEusq <- Eusq %*% t(Eusq)
EEusq <- EEusq - diag(diag(EEusq))

const <- k/(2*sigmasq0^2)
s1 <- const - (1/sigmasq0^3) * sum(Eusq)
s2 <- (const^2)*( (1/k^2) * sum(Eu4) + (1/k^2)* sum(c(EEusq)) - sigmasq0^2)

diag1 <- M - QQ + Q
```

```
    diag2 <- -(s1+s2)
    off.diag <- - (db.sig - dbeta.dsig)
    fisher <- rbind(cbind(diag1, off.diag), c(off.diag, diag2))

    est <- c(beta0, sigmasq0)
    est <- est + solve(fisher) %*% c(q,q1)

    beta0 <- est[1:6]
    beta <- cbind(beta, beta0)

    sigmasq0 <- est[7]
    sigmasq <- c(sigmasq, sigmasq0)

  }

  std <- (sqrt(diag(solve(fisher))))
  cat("\n")

  list(beta=beta, sigmasq=sigmasq, beta0=beta0,
       sigmasq0=sigmasq0, std=std)
}
guidefit.mle(iter=10)

R-output:
> data0
    id yy female age dayacc severe toilet
1    1  1      1 0.1  7.000      3      8
2    1  0      1 0.6  1.000      1      3
3    1  1      1 0.2  7.000      3      6
```

```
4    2  0      1 1.1  0.286      2      6
5    2  0      1 0.2  2.000      2      4
6    3  0      1 0.3  1.000      2      4
7    3  1      1 1.4 15.000      4     20
8    3  0      0 0.1  9.286      1     10
9    3  0      1 0.8  3.000      2      4
10   3  1      1 0.1 14.857      2     15
11   4  0      0 1.3  6.000      2      5
12   4  0      1 1.2  6.000      2      3
13   4  0      1 0.7  2.000      3     10
14   4  1      1 0.3  0.714      2      6
15   5  0      0 0.3  0.143      1      5
16   5  0      1 0.1  1.000      2      5
17   5  1      1 0.1  1.429      3      7
18   5  0      1 0.3  0.571      2      7
19   6  1      1 0.7  0.143      2      3
20   6  0      1 0.3  0.571      2      4
21   7  0      1 0.2  3.000      2      4
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
$beta
          beta0      beta0      beta0      beta0      beta0      beta0      beta0
[1,] -3.5241638 -3.5243614 -3.5243585 -3.5243585 -3.5243585 -3.5243585 -3.5243585
[2,] -0.9119856 -0.9121288 -0.9121256 -0.9121256 -0.9121256 -0.9121256 -0.9121256
[3,] -0.7827754 -0.7829036 -0.7829010 -0.7829010 -0.7829010 -0.7829010 -0.7829010
[4,]  0.4661583  0.4662138  0.4662130  0.4662130  0.4662130  0.4662130  0.4662130
[5,]  0.9617036  0.9617877  0.9617861  0.9617861  0.9617861  0.9617861  0.9617861
```

```
[6,]  0.1159137  0.1159186  0.1159185  0.1159185  0.1159185  0.1159185  0.1159185
          beta0      beta0      beta0
[1,] -3.5243585 -3.5243585 -3.5243585
[2,] -0.9121256 -0.9121256 -0.9121256
[3,] -0.7829010 -0.7829010 -0.7829010
[4,]  0.4662130  0.4662130  0.4662130
[5,]  0.9617861  0.9617861  0.9617861
[6,]  0.1159185  0.1159185  0.1159185

$sigmasq
 [1] 1.217078 1.217950 1.217931
  1.217931 1.217931 1.217931
  1.217931 1.217931 1.217931
[10] 1.217931

$beta0
[1] -3.5243585 -0.9121256 -0.7829010  0.4662130  0.9617861  0.1159185

$sigmasq0
[1] 1.217931

$std
 intercept     female        age     dayacc     severe     toilet
1.34491351 0.73706969 0.71569903 0.12319751 0.41850927 0.09796587 1.38463173

> guidedata <- read.table("C:\\Users\\r\\Desktop\\Thesis\\GuideData.txt",header=T)

> summary(guidedata)
    pract_id        doct_id          pat_id       bothered          female
```

```
Min.   :  8.0   Min.   :  2.0   Min.   :  1   Min.   :0.0000   Min.   :0.000
1st Qu.: 89.0   1st Qu.:129.0   1st Qu.: 35   1st Qu.:0.0000   1st Qu.:1.000
Median :125.0   Median :289.0   Median : 69   Median :0.0000   Median :1.000
Mean   :133.9   Mean   :290.2   Mean   : 69   Mean   :0.3942   Mean   :0.854
3rd Qu.:201.0   3rd Qu.:390.0   3rd Qu.:103   3rd Qu.:1.0000   3rd Qu.:1.000
Max.   :235.0   Max.   :815.0   Max.   :137   Max.   :1.0000   Max.   :1.000

    ageyrs           age            weekacc           dayacc           severe
Min.   :76.00   Min.   :0.0000   Min.   :  1.00   Min.   : 0.143   Min.   :1.000
1st Qu.:78.00   1st Qu.:0.2000   1st Qu.:  4.00   1st Qu.: 0.571   1st Qu.:2.000
Median :80.00   Median :0.4000   Median : 14.00   Median : 2.000   Median :2.000
Mean   :81.25   Mean   :0.5248   Mean   : 22.76   Mean   : 3.251   Mean   :2.117
3rd Qu.:84.00   3rd Qu.:0.8000   3rd Qu.: 35.00   3rd Qu.: 5.000   3rd Qu.:2.000
Max.   :91.00   Max.   :1.5000   Max.   :117.00   Max.   :16.714   Max.   :4.000

    toilet           mdage         dobs               rit
Min.   : 2.000   45     :18   Min.   :0.000010   Min.   :-3.32900
1st Qu.: 4.000   41     :12   1st Qu.:0.000490   1st Qu.:-0.54100
Median : 6.000   42     :11   Median :0.001510   Median :-0.35000
Mean   : 6.372   40     :10   Mean   :0.009681   Mean   :-0.01625
3rd Qu.: 8.000   50     : 8   3rd Qu.:0.008410   3rd Qu.: 0.48300
Max.   :20.000   34     : 7   Max.   :0.262560   Max.   : 3.86100
                 (Other):71

     hit
Min.   :0.00400
1st Qu.:0.02000
Median :0.03500
Mean   :0.04377
3rd Qu.:0.05500
```

```
 Max.   :0.26500
```

```
se <-c(1.34491351, 0.73706969 ,0.71569903, 0.12319751 ,0.41850927 ,
0.09796587 ,1.38463173 )

beta0<-c(-3.5243585, -0.9121256, -0.7829010 , 0.4662130 , 0.9617861 ,
 0.1159185,1.217931)
zvalue <- (beta0)/(s.e)

pvalue <- 2 * pnorm(-abs(zvalue),0,1)

table<-cbind(beta0,s.e,zvalue,pvalue)
```

```
             beta0        s.e       zvalue       pvalue
(Intercept) -3.5243585 1.34491351 -2.6205094 0.0087798509***
female      -0.9121256 0.73706969 -1.2375025 0.2159005788
age         -0.7829010 0.71569903 -1.0938970 0.2740001680
dayacc       0.4662130 0.12319751  3.7842729 0.0001541586***
severe       0.9617861 0.41850927  2.2981238 0.0215547459*
toilet       0.1159185 0.09796587  1.1832539 0.2367085280
sigmasq      1.2179310 1.38463173  0.8796064 0.3790725437
```

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

B-3:\\

```
R code:\\

data0 <- read.table("C:\\Users\\r\\Desktop\\data0.txt",header=T)
data0
library(glmmML)
fit1 <- glmmML(yy ~ female + age + dayacc + severe +
                     toilet, data = data0, family=binomial, cluster=id,
                   method="ghq", n.points = 8)

summary(fit1)
Call:  glmmML(formula = yy ~ female + age + dayacc + severe + toilet,
 family = binomial, data = data0,
 cluster = id, method = "ghq",
   n.points = 8)

table(2.6)( Guide Data Summary GlmmMl.)

               coef se(coef)      z Pr(>|z|)
(Intercept) -3.5244  1.34487 -2.621 0.008780***
female      -0.9120  0.73703 -1.237 0.216000
age         -0.7828  0.71566 -1.094 0.274000
dayacc       0.4662  0.12319  3.784 0.000154***
severe       0.9618  0.41849  2.298 0.021500*
toilet       0.1159  0.09796  1.183 0.237000

Scale parameter in mixing distribution:  1.103 gaussian
Std. Error:                              0.6271

        LR p-value for H_0: sigma = 0:  0.1076
```

```
Residual deviance: 125.6 on 130 degrees of freedom  AIC: 139.6


-----------------------------------------------

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

C: R Codes for Chapter 4

```
 C-1:\\
 R code\\
binary.dat <- function(k=200, n=3, beta=c(-2, 3), sigmasq=1, tau=c(2, 0, -3))
  # generates data from binary mixed model with nonignorable missing responses
  # k = number of clusters
  # n = number of observations in each cluster
  # the code is written for n = 3
{
  id <- rep(c(1:k), rep(n, k))

  u0 <- rnorm(k, 0, sqrt(sigmasq))
  u <- rep(u0, rep(n, k))

  N <- k * n
  one <- rep(1, N)

  x <- round(rnorm(N, mean=1, sd=1), digits=2)
  xx <- cbind(one, x)
```

```
  eta <- c(xx %*% beta) + u
  p <- round(exp(eta)/(1+exp(eta)), digits=3)

  y <- rbinom(N, 1, p)

  z <- cbind(one, x, y)
  zeta <- c(z %*% tau)

  pi0 <- round(exp(zeta)/(1+exp(zeta)), digits=3)
  v <- rbinom(N, 1, pi0)

  # ====================================
  # v = 1 for observed; v = 0 for missing
  # ====================================

  data <- data.frame(id, y, x, p, u, v, pi0)
  data
}




# data0 <- binary.dat(k=200, n=3, beta=c(-2, 3), sigmasq=1, tau=c(2, 0, -3))

glmm.like.miss <- function(dat=data0, beta0=c(-2, 3), sigmasq0=1,
                           tau0=c(2, 0, -3))
  # maximizes likelihood with missing data
  # "optim" function is used to minimize -log.likelihood
{
```

```
id <- dat$id

y <- dat$y
x <- dat$x
v <- dat$v

u.id <- unique(id)
k <- length(u.id)

initial <- c(beta0, sigmasq0, tau0)

obj.fn <- function(theta)
{

  beta0 <- theta[1:2]
  sigmasq0 <- theta[3]
  tau0 <- theta[4:6]

  neg.log.like <- 0

  for(i in 1:k)
  {

    yi <- y[id==u.id[i]]
    xi <- x[id==u.id[i]]
    vi <- v[id==u.id[i]]

    n <- length(yi)
```

```
like.fn <- function(u)
{

  if(all(vi==c(1, 1, 1)))
  {
    f.yi <- 1
    f.vi <- 1

    for(j in 1:n)
    {

      vij <- vi[j]

      yij <- yi[j]
      xij <- xi[j]

      etaij <- c(t(c(1, xij)) %*% beta0) + u
      muij <- exp(etaij)/(1+exp(etaij))

      f.yij <- muij^yij * (1-muij)^(1-yij)
      f.yi <- f.yi * f.yij

      zetaij <- c(t(c(1, xij, yij)) %*% tau0)
      piij <- exp(zetaij)/(1+exp(zetaij))

      f.vij <- piij^vij * (1-piij)^(1-vij)
      f.vi <- f.vi * f.vij
    }
```

```
  f.yi.vi <- f.yi * f.vi
}

else

{

  if(all(vi==c(1,1,0)))
  yi.pop <- cbind(c(yi[1],yi[2], 0), c(yi[1],yi[2], 1))
  if(all(vi==c(1,0,1)))
   yi.pop <- cbind(c(yi[1],0, yi[3]), c(yi[1],1, yi[3]))
  if(all(vi==c(0,1,1)))
   yi.pop <- cbind(c(0, yi[2],yi[3]), c(1,yi[2], yi[3]))
  if(all(vi==c(1,0,0)))
   yi.pop <- cbind(c(yi[1],1,1),  c(yi[1],1,0),  c(yi[1],0,1),
     c(yi[1],1,1))
  if(all(vi==c(0,1,0)))
  yi.pop <- cbind(c(1,yi[2],1),  c(1,yi[2],0),  c(0,yi[2],1),
   c(0,yi[2],0))
  if(all(vi==c(0,0,1)))
   yi.pop <- cbind(c(1,1,yi[3]),  c(1,0,yi[3]),  c(0,1,yi[3]),
   c(0,0,yi[3]))
  if(all(vi==c(0,0,0)))
  yi.pop <- cbind(c(rep(1,3)),   c(rep(1,2),0), c(0,rep(1,2)),
   c(1,rep(0,1),1),
   c(rep(0,2),1),
     c(1,rep(0,2)), c(0,rep(1,1),0), c(rep(0,3)))
```

```
m0 <- ncol(yi.pop)

f.yi.vi.sum <- 0

for(m in 1:m0)
{
  yi.m <- yi.pop[, m]

  f.yi.m <- 1
  f.vi.m <- 1

  for(j in 1:n)

  {
    vij <- vi[j]

    yij.m <- yi.m[j]
    xij <- xi[j]

    etaij <- c(t(c(1, xij)) %*% beta0) + u
    muij <- exp(etaij)/(1+exp(etaij))

    f.yij.m <- muij^yij.m * (1-muij)^(1-yij.m)
    f.yi.m <- f.yi.m * f.yij.m

    zetaij.m <- c(t(c(1, xij, yij.m)) %*% tau0)
    piij.m <- exp(zetaij.m)/(1+exp(zetaij.m))

    f.vij.m <- piij.m^vij * (1-piij.m)^(1-vij)
```

```
            f.vi.m <- f.vi.m * f.vij.m
          }

          fyi.vi.m <- f.yi.m * f.vi.m
          f.yi.vi.sum <- f.yi.vi.sum + fyi.vi.m
        }

        f.yi.vi <- f.yi.vi.sum
      }

      f.ui <- dnorm(u, mean=0, sd=sqrt(sigmasq0))

      f.yi.vi * f.ui
    }

    like.i <- integrate(like.fn, lower=-10, upper=10)$value

    neg.log.like.i <- -log(like.i)
    neg.log.like <- neg.log.like + neg.log.like.i
  }

  neg.log.like
}

fit <- optim(par=initial, fn=obj.fn, hessian=TRUE, method = c("L-BFGS-B"),
  lower=c(-Inf, -Inf, 0, -Inf, -Inf, -Inf),
upper=c(Inf, Inf, Inf, Inf, Inf, Inf))

cat(fit$message, "\n")
```

```
  estimate <- fit$par
  std.err <- sqrt(diag(solve(fit$hessian)))
  objective <- fit$value

  list(estimate=estimate, std.err=std.err, objective=objective)

}



# ml.fit <- glmm.like.miss(dat=data0, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))




rglmm.sim <- function(sim=2, k=200, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
{

  estimate <- NULL
  std.err <- NULL



  for(i in 1:sim)
  {
    cat(i,"\n")

    data0 <- binary.dat(k=k, n=n, beta=beta0, sigmasq=sigmasq0, tau=tau0)
```

```
    cat("Computing the Exact ML Estimates\n")
    ml.fit <- glmm.like.miss(dat=data0, beta0=beta0, sigmasq0=sigmasq0, tau0=tau0)

    estimate0 <- ml.fit$estimate
    std.err0 <- ml.fit$std.err

    cat("\n")



    estimate <- rbind(estimate, estimate0)
    std.err <- rbind(std.err, std.err0)

  }

  cat("\n")

  list(estimate=round(estimate, digits=4), std.err=round(std.err, digits=4))
}



# x1 <- rglmm.sim(sim=2, k=200, n=3, beta0=c(-2, 3), sigmasq0=1, tau0=c(2, 0, -3))


rglmm.sim.try <- function(sim=2, k=200, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))

{
  doit1 <- function(s=1)
```

```
{
  cat("Simulation no:", s, "\n")

  data0 <- binary.dat(k=k, n=n, beta=beta0, sigmasq=sigmasq0, tau=tau0)

  cat("Computing the Exact ML Estimates\n")
  ml.fit <- glmm.like.miss(dat=data0, beta0=beta0, sigmasq0=sigmasq0, tau0=tau0)

  coef <- ml.fit$estimate
  std.err <- ml.fit$std.err

  list(coef=coef, std.err=std.err)
}

cat("\n")

res1 <- lapply(1:sim, function(i) try(doit1(s=i), TRUE))
res.run1 <- res1[sapply(res1, function(x) !inherits(x, "try-error"))]

length.res.run1 <- length(res.run1)
cat("Total samples:", length.res.run1, "\n")

estimate <- NULL

for(i in 1:length.res.run1)
{

  coef <- res.run1[[i]]$coef
  std.err <- res.run1[[i]]$std.err
```

```
    estimate0 <- c(coef, std.err)
    estimate <- rbind(estimate, estimate0)
  }

  dimnames(estimate) <- list(NULL, c("beta0", "beta1", "sigmasq", "tau0", "tau1",
 "tau2", "std.beta0", "std.beta1",
  "std.sigmasq", "std.tau0",
   "std.tau1", "std.tau2"))
  estimate
}



#glmm.sim1 <- rglmm.sim.try(sim=500, k=100, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
#glmm.sim2 <- rglmm.sim.try(sim=500, k=200, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
#glmm.sim3 <- rglmm.sim.try(sim=500, k=300, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))



#glmm.sim4 <- rglmm.sim.try(sim=500, k=100, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0.5, -3))
glmm.sim5 <- rglmm.sim.try(sim=500, k=200, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0.5, -3))
#glmm.sim6 <- rglmm.sim.try(sim=500, k=300, n=3, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0.5, -3))
```

```
data0
```

```
> data0
     id y     x     p            u v   pi0
1     1 0 -1.30 0.014 -1.937551207 1 0.934
2     1 1  3.28 0.585 -1.937551207 0 0.345
3     2 0 -0.59 0.749  2.684338473 1 0.908
4     2 1  1.49 0.960  2.684338473 1 0.563
5     3 0  0.59 0.094 -1.858367995 1 0.846
6     3 0 -0.54 0.032 -1.858367995 1 0.906
7     4 0  0.47 0.412  0.175672798 1 0.854
8     4 1  0.59 0.442  0.175672798 1 0.669
9     5 1  1.06 0.603  0.357349772 0 0.615
10    5 1  1.72 0.746  0.357349772 0 0.535
11    6 0  0.33 0.055 -2.176650508 1 0.862
12    6 0  0.23 0.050 -2.176650508 0 0.868
```

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

C-2:\\

```
R-code:\\


library(glmmML)
beta0.hat<-NULL
beta1.hat<-NULL
sigmasq.hat<-NULL

stdr.beta0<-NULL
stdr.beta1<-NULL
stdr.sigmasq<-NULL




for(s in 1:500)
{ binary.dat <- function(k=300, n=2, beta=c(-2, 3), sigmasq=1, tau=c(2, 0.5, -3))
  # generates data from binary mixed model with nonignorable missing responses
  # k = number of clusters
  # n = number of observations in each cluster
  # the code is written for n = 3
{
  id <- rep(c(1:k), rep(n, k))

  u0 <- rnorm(k, 0, sqrt(sigmasq))
  u <- rep(u0, rep(n, k))

  N <- k * n
```

```
  one <- rep(1, N)

  x <- round(rnorm(N, mean=1, sd=1), digits=2)
  xx <- cbind(one, x)

  eta <- c(xx %*% beta) + u
  p <- round(exp(eta)/(1+exp(eta)), digits=3)

  y <- rbinom(N, 1, p)

  z <- cbind(one, x, y)
  zeta <- c(z %*% tau)

  pi0 <- round(exp(zeta)/(1+exp(zeta)), digits=3)
  v <- rbinom(N, 1, pi0)

  # ====================================
  # v = 1 for observed; v = 0 for missing
  # ====================================

  data <- data.frame(id, y, x, p, u, v, pi0)
  data
}



data0 <- binary.dat(k=300, n=2, beta=c(-2, 3))

data1<-subset(data0, v == "1")
data1
```

```
fit <- glmmML(y ~ x, data = data1, family=binomial, cluster=id,
              method="ghq", n.points = 25)




fit
beta0<-cbind.data.frame(coef(fit))[1,1]
beta1<-cbind.data.frame(coef(fit))[2,1]
sigmasq<-(fit$sigma)^2
std.beta0<-cbind.data.frame(fit$coef.sd)[1,1]
std.beta1<-cbind.data.frame(fit$coef.sd)[2,1]
std.sigmasq<-fit$sigma.sd

beta0.hat<-c(beta0.hat,beta0)
beta1.hat<-c(beta1.hat,beta1)
sigmasq.hat<-c(sigmasq.hat,sigmasq)

stdr.beta0<-c(stdr.beta0,std.beta0)
stdr.beta1<-c(stdr.beta1,std.beta1)
stdr.sigmasq<-c( stdr.sigmasq,std.sigmasq)




}

beta0.h<-sum(beta0.hat)/500
beta1.h<-round(mean(beta1.hat,na.rm=TRUE),digits=3)
sigmasq.h<-round(mean(sigmasq.hat,na.rm=TRUE),digits=3)
```

```
bias.beta0<-round(mean(beta0.hat)+2,digits=3)
bias.beta1<-round(mean(beta1.hat)-3,digits=3)
bias.sigmasq<-round(mean(sigmasq.hat,na.rm=TRUE)-1,digits=3)
beta0.hat[complete.cases(beta0.hat) ]
beta1.hat[complete.cases(beta1.hat) ]
sigmasq.hat[complete.cases(sigmasq.hat) ]

length(beta0.hat)

length(beta1.hat)

length(sigmasq.hat)

MSE.beta0=sum((beta0.hat+2)^2)/500
MSE.beta1=sum((beta1.hat-3)^2)/500
MSE.sigmasq=sum((sigmasq.hat-1)^2)/500

lb.beta0<-beta0.hat-qt(1-0.05/2,598)*stdr.beta0
ub.beta0<-beta0.hat+qt(1-0.05/2,598)*stdr.beta0



lb.beta1<-beta1.hat-qt(1-0.05/2,598)*stdr.beta1
ub.beta1<-beta1.hat+qt(1-0.05/2,598)*stdr.beta1



lb.sigmasq<-sigmasq.hat-1.96*stdr.sigmasq
```

```
ub.sigmasq<-sigmasq.hat+1.96*stdr.sigmasq




CI.beta0<-cbind(lb.beta0,ub.beta0)
CI.beta00<-as.matrix(CI.beta0[complete.cases(CI.beta0), ])

CI.beta1<-cbind(lb.beta1,ub.beta1)
CI.beta11<-as.matrix(CI.beta1[complete.cases(CI.beta1), ])

CI.sigmasq<-cbind(lb.sigmasq,ub.sigmasq)
CI.sigmasq<-as.matrix(CI.sigmasq[complete.cases(CI.sigmasq), ])



beta0.hat
length(CI.beta00)
CP0<-0

for (i in 1:500) {

  if ( (-2 > CI.beta00[i,1] )&& (-2< CI.beta00[i,2]) )
    CP0<-CP0+1

}
CP0<-(CP0/500)*100

length(CI.beta11)
```

```
CP1<-0

for (i in 1:500) {

  if ( (3 > CI.beta11[i,1] )&& (3< CI.beta11[i,2]) )
    CP1<-CP1+1

}
CP1<-(CP1/500)*100

length(CI.sigmasq)

CP2<-0

for (i in 1:500) {

  if ( (1 > CI.sigmasq[i,1] )&& (1< CI.sigmasq[i,2]) )
    CP2<-CP2+1

}
CP2<-(CP2/500)*100

cp.beta0.per<-round((cp.beta0/500)*100,digits=3)
cp.beta1.per<-round((cp.beta1/500)*100,digits=3)
cp.sigmasq.per<-round((cp.sigmasq/500)*100,digits=3)


av.ci.length.beta0<-round(mean((ub.beta0-lb.beta0),na.rm=TRUE),digits=3)
```

```
av.ci.length.beta1<-round(mean((ub.beta1-lb.beta1),na.rm=TRUE),digits=3)
av.ci.length.sigmasq<-round(mean((ub.sigmasq-lb.sigmasq),na.rm=TRUE),digits=3)

beta0.hat<-round(beta0.hat,digits = 3)
beta1.hat<-round(beta1.hat,digits = 3)
sigmasq.hat<-round(sigmasq.hat,digits = 3)
stdr.beta0<-round(stdr.beta0,digits = 3)
stdr.beta1<-round(stdr.beta1,digits = 3)
stdr.sigmasq<-round(stdr.sigmasq,digits = 3)
table<-cbind(beta0.hat,beta1.hat,sigmasq.hat,stdr.beta0,stdr.beta1,stdr.sigmasq)
write.table(table,"foo.txt")
```

C-3:\\
R-code:

```
binary.dat <- function(k=200, n=2, beta=c(-2, 3), sigmasq=1, tau=c(2, 0, -3))
  # generates data from binary mixed model with nonignorable missing responses
  # k = number of clusters
  # n = number of observations in each cluster
  # the code is written for n = 2
```

```
{
  id <- rep(c(1:k), rep(n, k))

  u0 <- rnorm(k, 0, sqrt(sigmasq))
  u <- rep(u0, rep(n, k))

  N <- k * n
  one <- rep(1, N)

  x <- round(rnorm(N, mean=1, sd=1), digits=2)
  xx <- cbind(one, x)

  eta <- c(xx %*% beta) + u
  p <- round(exp(eta)/(1+exp(eta)), digits=3)

  y <- rbinom(N, 1, p)

  z <- cbind(one, x, y)
  zeta <- c(z %*% tau)

  pi0 <- round(exp(zeta)/(1+exp(zeta)), digits=3)
  v <- rbinom(N, 1, pi0)

  # ====================================
  # v = 1 for observed; v = 0 for missing
  # ====================================

  data <- data.frame(id, y, x, p, u, v, pi0)
  data
```

```
}



# data0 <- binary.dat(k=200, n=2, beta=c(-2, 3), sigmasq=1, tau=c(2, 0, -3))

glmm.like.miss <- function(dat=data0, beta0=c(-2, 3), sigmasq0=1,
                           tau0=c(2, 0, -3))
  # maximizes likelihood with missing data
  # "optim" function is used to minimize -log.likelihood
{

  id <- dat$id

  y <- dat$y
  x <- dat$x
  v <- dat$v

  u.id <- unique(id)
  k <- length(u.id)

  initial <- c(beta0, sigmasq0, tau0)

  obj.fn <- function(theta)
  {

    beta0 <- theta[1:2]
    sigmasq0 <- theta[3]
    tau0 <- theta[4:6]
```

```
neg.log.like <- 0

for(i in 1:k)
{

  yi <- y[id==u.id[i]]
  xi <- x[id==u.id[i]]
  vi <- v[id==u.id[i]]

  n <- length(yi)

  like.fn <- function(u)
  {

    if(all(vi==c(1, 1)))
    {
      f.yi <- 1
      f.vi <- 1

      for(j in 1:n)
      {

        vij <- vi[j]

        yij <- yi[j]
        xij <- xi[j]

        etaij <- c(t(c(1, xij)) %*% beta0) + u
```

```
    muij <- exp(etaij)/(1+exp(etaij))

    f.yij <- muij^yij * (1-muij)^(1-yij)
    f.yi <- f.yi * f.yij

    zetaij <- c(t(c(1, xij, yij)) %*% tau0)
    piij <- exp(zetaij)/(1+exp(zetaij))

    f.vij <- piij^vij * (1-piij)^(1-vij)
    f.vi <- f.vi * f.vij
  }

  f.yi.vi <- f.yi * f.vi
}

else

{

  if(all(vi==c(1, 0))) yi.pop <- cbind(c(yi[1], 0), c(yi[1], 1))
  if(all(vi==c(0, 1))) yi.pop <- cbind(c(0, yi[2]), c(1, yi[2]))
  if(all(vi==c(0, 0))) yi.pop <- cbind(c(0, 0), c(1, 0), c(0, 1), c(1, 1))

  m0 <- ncol(yi.pop)

  f.yi.vi.sum <- 0

  for(m in 1:m0)
  {
```

```
  yi.m <- yi.pop[, m]

  f.yi.m <- 1
  f.vi.m <- 1

  for(j in 1:n)

  {
    vij <- vi[j]

    yij.m <- yi.m[j]
    xij <- xi[j]

    etaij <- c(t(c(1, xij)) %*% beta0) + u
    muij <- exp(etaij)/(1+exp(etaij))

    f.yij.m <- muij^yij.m * (1-muij)^(1-yij.m)
    f.yi.m <- f.yi.m * f.yij.m

    zetaij.m <- c(t(c(1, xij, yij.m)) %*% tau0)
    piij.m <- exp(zetaij.m)/(1+exp(zetaij.m))

    f.vij.m <- piij.m^vij * (1-piij.m)^(1-vij)
    f.vi.m <- f.vi.m * f.vij.m
  }

  fyi.vi.m <- f.yi.m * f.vi.m
  f.yi.vi.sum <- f.yi.vi.sum + fyi.vi.m
}
```

```
        f.yi.vi <- f.yi.vi.sum
      }

      f.ui <- dnorm(u, mean=0, sd=sqrt(sigmasq0))

      f.yi.vi * f.ui
    }

    like.i <- integrate(like.fn, lower=-10, upper=10)$value

    neg.log.like.i <- -log(like.i)
    neg.log.like <- neg.log.like + neg.log.like.i
  }

  neg.log.like
}

fit <- optim(par=initial, fn=obj.fn, hessian=TRUE, method = c("L-BFGS-B"),
             lower=c(-Inf, -Inf, 0, -Inf, -Inf, -Inf),
upper=c(Inf, Inf, Inf, Inf, Inf, Inf))

cat(fit$message, "\n")

estimate <- fit$par
std.err <- sqrt(diag(solve(fit$hessian)))
objective <- fit$value

list(estimate=estimate, std.err=std.err, objective=objective)
```

```
}


# ml.fit <- glmm.like.miss(dat=data0, beta0=c(-2, 3), sigmasq0=1,
tau0=c(2, 0, -3))



rglmm.sim <- function(sim=2, k=200, n=2, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
{

  estimate <- NULL
  std.err <- NULL


  for(i in 1:sim)
  {
    cat(i,"\n")

    data0 <- binary.dat(k=k, n=n, beta=beta0, sigmasq=sigmasq0, tau=tau0)

    cat("Computing the Exact ML Estimates\n")
    ml.fit <- glmm.like.miss(dat=data0, beta0=beta0, sigmasq0=sigmasq0, tau0=tau0)

    estimate0 <- ml.fit$estimate
    std.err0 <- ml.fit$std.err
```

```
    cat("\n")


    estimate <- rbind(estimate, estimate0)
    std.err <- rbind(std.err, std.err0)

  }

  cat("\n")

  list(estimate=round(estimate, digits=4), std.err=round(std.err, digits=4))
}



# x1 <- rglmm.sim(sim=2, k=200, n=3, beta0=c(-2, 3), sigmasq0=1, tau0=c(2, 0, -3))


rglmm.sim.try <- function(sim=2, k=200, n=2, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))

{
  doit1 <- function(s=1)
  {
    cat("Simulation no:", s, "\n")

    data0 <- binary.dat(k=k, n=n, beta=beta0, sigmasq=sigmasq0, tau=tau0)

    cat("Computing the Exact ML Estimates\n")
```

```
  ml.fit <- glmm.like.miss(dat=data0, beta0=beta0, sigmasq0=sigmasq0, tau0=tau0)

  coef <- ml.fit$estimate
  std.err <- ml.fit$std.err

  list(coef=coef, std.err=std.err)
}

cat("\n")

res1 <- lapply(1:sim, function(i) try(doit1(s=i), TRUE))
res.run1 <- res1[sapply(res1, function(x) !inherits(x, "try-error"))]

length.res.run1 <- length(res.run1)
cat("Total samples:", length.res.run1, "\n")

estimate <- NULL

for(i in 1:length.res.run1)
{

  coef <- res.run1[[i]]$coef
  std.err <- res.run1[[i]]$std.err



  estimate0 <- c(coef, std.err)
  estimate <- rbind(estimate, estimate0)
}
```

```
  dimnames(estimate) <- list(NULL, c("beta0", "beta1", "sigmasq", "tau0", "tau1",
   "tau2", "std.beta0", "std.beta1", "std.sigmasq",
"std.tau0", "std.tau1", "std.tau2"))
  estimate
}



#glmm.sim1 <- rglmm.sim.try(sim=500, k=100, n=2, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
#glmm.sim2 <- rglmm.sim.try(sim=500, k=200, n=2, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0, -3))
glmm.sim3 <- rglmm.sim.try(sim=500, k=300, n=2, beta0=c(-2, 3), sigmasq0=1,
tau0=c(2, 0, -3))



#glmm.sim4 <- rglmm.sim.try(sim=500, k=100, n=2, beta0=c(-2, 3), sigmasq0=1,
 tau0=c(2, 0.5, -3))
#glmm.sim5 <- rglmm.sim.try(sim=500, k=200, n=2, beta0=c(-2, 3), sigmasq0=1,
tau0=c(2, 0.5, -3))
#glmm.sim6 <- rglmm.sim.try(sim=500, k=300, n=2, beta0=c(-2, 3), sigmasq0=1,
tau0=c(2, 0.5, -3))



}



write.table(glmm.sim2, "foo.txt", sep=",")
```

```
C-4:\\
R-code:
 \begin{verbatim}
 library(glmmML)
data<-read.csv("C:\\Users\\r\\Desktop\\Thesis\\smoke.data.mis.csv",header = T)
head(data)
y<-data$y
x0<-data$age
x1<-x0/10
x3<-data$B.F
x4<-data$W.M
x5<-data$W.F
x2<-data$yr_fu
id<-data$id-10000
v<-data$v



data0<-cbind.data.frame(id,y,x1,x2,x4,x3,x5,v)
head(data0)
data0<-data0[y!=9,]
fit<-glmmML(y~x1+x2+x3+x4+x5,family=binomial,
 data = data0,cluster = id,
       method = "ghq", n.points = 25)
summary(fit)
 > summary(fit)

Call:  glmmML(formula = y ~ x1 + x2 + x3 + x4 + x5, family = binomial,
data = data0, cluster = id, method = "ghq", n.points = 25)
```

```
               coef se(coef)      z Pr(>|z|)
(Intercept) -3.6030  0.96874 -3.719 2.00e-04**
x1           0.5071  0.38113  1.331 1.83e-01
x2          -0.0553  0.01337 -4.136 3.54e-05**
x3          -1.5670  0.38149 -4.108 4.00e-05**
x4          -3.1473  0.41857 -7.519 5.51e-14***
x5          -3.1978  0.40540 -7.888 3.11e-15***

Scale parameter in mixing distribution:  7.786 gaussian
Std. Error:                              0.1733

        LR p-value for H_0: sigma = 0:  0

Residual deviance: 12680 on 17988 degrees of freedom  AIC: 12700
```

C-5:\\

R-code:

```
data0 <- read.csv("C:\\Users\\r\\Desktop\\Thesis\\smoke.data.mis.csv",header = T)

head(data0)

glmm.like.miss<-  function(dat=data0, beta0=c(-1, -0.1, 0.1, -1, -2, -2),
sigma.u0=5, tau0=c(4, -0.3, -1.5))
```

```
  # maximizes likelihood with missing data
  # "optim" function is used to minimize -log.likelihood
{

  id <- dat$id
  y <- dat$y

  x1 <- dat$age/10
  x2 <- dat$yr_fu
  x3 <- dat$B.F
  x4 <- dat$W.M
  x5 <- dat$W.F

  s <- x2
  v <- dat$v

  u.id <- unique(id)
  k <- length(u.id)

  x <- cbind(1, x1, x2, x3, x4, x5)

  initial <- c(beta0, sigma.u0, tau0)

  obj.fn <- function(theta)
  {

    beta0 <- theta[1:6]
    sigma.u0 <- theta[7]
    tau0 <- theta[8:10]
```

```
neg.log.like <- 0

for(i in 1:k)
{


  vi <- v[id==u.id[i]]
  yi <- y[id==u.id[i]]

  ni <- length(yi)

  xi <- x[id==u.id[i], ]
  si <- s[id==u.id[i]]

  like.fn <- function(u)
  {

    if(all(vi==c(1, 1, 1, 1)))
    {
      f.yi <- 1
      f.vi <- 1

      for(j in 1:ni)
      {

        yij <- yi[j]
        xij <- xi[j, ]
```

```
  sij <- si[j]

  vij <- vi[j]


  etaij <- c(t(xij) %*% beta0) + u
  muij <- exp(etaij)/(1+exp(etaij))

  f.yij <- muij^yij * (1-muij)^(1-yij)
  f.yi <- f.yi * f.yij


  zetaij <- c(t(c(1, sij, yij)) %*% tau0)

  piij <- exp(zetaij)/(1+exp(zetaij))

  if (j==1)
  {
    f.vij <- 1
  }
  else
  {
    f.vij <- piij^vij * (1-piij)^(1-vij)
  }

  f.vi <- f.vi * f.vij

}
```

```
  f.yi.vi <- f.yi * f.vi

}

else

{
  if(all(vi==c(1,1,1,0))) yi.pop <- cbind(c(yi[1], yi[2], yi[3],     0),
   c(yi[1], yi[2], yi[3],     1))
  if(all(vi==c(1,1,0,1))) yi.pop <- cbind(c(yi[1], yi[2],     0, yi[4]),
   c(yi[1], yi[2],     1, yi[4]))
  if(all(vi==c(1,0,1,1))) yi.pop <- cbind(c(yi[1],     0, yi[3], yi[4]),
   c(yi[1],     1, yi[3], yi[4]))

  if(all(vi==c(1,1,0,0))) yi.pop <- cbind(c(yi[1], yi[2],     1,     1),
  c(yi[1], yi[2],     1,     0),  c(yi[1], yi[2],     0,     1),
   c(yi[1], yi[2],     0,     0))
  if(all(vi==c(1,0,1,0))) yi.pop <- cbind(c(yi[1],     1, yi[3],     1),
  c(yi[1],     1, yi[3],     0),  c(yi[1],     0, yi[3],     1),
   c(yi[1],     0, yi[3],     0))
  if(all(vi==c(1,0,0,1))) yi.pop <- cbind(c(yi[1],     1,     1, yi[4]),
   c(yi[1],     1,     0, yi[4]),  c(yi[1],     0,     1, yi[4]),
   c(yi[1],     0,     0, yi[4]))
  if(all(vi==c(1,0,0,0))) yi.pop <- cbind(c(yi[1],     1,     1,     1),
c(yi[1],     1,     1,     0),
c(yi[1],     1,     0,     1),
c(yi[1],     0,     1,     1),
c(yi[1],     1,     0,     0),
c(yi[1],     0,     1,     0),
```

```
c(yi[1],     0,     0,     1),
c(yi[1],     0,     0,     0))

        m0 <- ncol(yi.pop)

        f.yi.vi.sum <- 0

        for(m in 1:m0)

        {

          yi.m <- yi.pop[, m]
          f.yi.m <- 1
          f.vi.m <- 1

          for(j in 1:ni)

          {

            yij.m <- yi.m[j]
            xij <- xi[j, ]



            sij <- si[j]
            vij <- vi[j]

            etaij <- c(t(xij) %*% beta0) + u
            muij <- exp(etaij)/(1+exp(etaij))
```

```
        f.yij.m <- muij^yij.m * (1-muij)^(1-yij.m)
        f.yi.m <- f.yi.m * f.yij.m

        zetaij.m <- c(t(c(1, sij, yij.m)) %*% tau0)
        piij.m <- exp(zetaij.m)/(1+exp(zetaij.m))

        if (j==1)
        {
          f.vij.m <- 1
        }
        else
        {
          f.vij.m <- piij.m^vij * (1-piij.m)^(1-vij)
        }

        f.vi.m <- f.vi.m * f.vij.m
      }

      fyi.vi.m <- f.yi.m * f.vi.m
      f.yi.vi.sum <- f.yi.vi.sum + fyi.vi.m
    }

    f.yi.vi <- f.yi.vi.sum
  }

  f.ui <- dnorm(u, mean=0, sd=sigma.u0)
  f.yi.vi * f.ui
}
```

```
      like.i <- integrate(like.fn, lower=-20, upper=20)$value

      neg.log.like.i <- -log(like.i)
      neg.log.like <- neg.log.like + neg.log.like.i
    }

    print(c(neg.log.like, theta))

    neg.log.like
  }

  fit <- optim(par=initial, fn=obj.fn, hessian=TRUE, method = c("L-BFGS-B"),
               lower=c(-5, -1, -1, -4, -4, -4,   0.2,   1, -2, -2),
               upper=c( 0,  1,  1,  0,  0,  0,    10,   5,  0,  0))
  cat(fit$message, "\n")

  estimate <- fit$par
  std.err <- sqrt(diag(solve(fit$hessian)))
  objective <- fit$value

  list(estimate=estimate, std.err=std.err, objective=objective)
}


smoke.fit <- glmm.like.miss(dat=data0, beta0=c(-1, -0.1, 0.1, -1, -2, -2),
 sigma.u0=25, tau0=c(4, -0.3, -1.5))

#smoke.fit1 <- glmm.like.miss(dat=data0, beta0=c(-3, 0.5, 0.01, -2, -2.5, -2.5),
 sigma.u0=25, tau0=c(3, -0.3, -0.8))
```

```
#smoke.fit2 <- glmm.like.miss(dat=data0, beta0=c(-3, -0.5, -0.1, -1, -1, -2),
 sigma.u0=2, tau0=c(3, -0.5, -0.5))

> se <-c( 0.005547911 , 0.006185823 ,0.006960497 ,0.209727595, 0.112595729
+         ,0.272347649 ,0.149250328, 0.070171577 ,0.011802995 ,0.013125852)
>
> beta0<-c( -3.119317689 , 0.727839216 , 0.002582718 ,-2.240311985 ,-2.901255793
+            ,-3.016871372,6.372554352 , 2.912942081, -0.183718726 ,-0.797422839)
>
> zvalue <- (beta0)/(se)
>
> pvalue <- 2 * pnorm(-abs(zvalue),0,1)
>
> table<-cbind(beta0,se,zvalue,pvalue)
> table
             beta0          se       zvalue        pvalue
 [1,] -3.119317689 0.005547911 -562.2508524  0.000000e+00
 [2,]  0.727839216 0.006185823  117.6624705  0.000000e+00
 [3,]  0.002582718 0.006960497    0.3710537  7.105976e-01
 [4,] -2.240311985 0.209727595  -10.6820087  1.235686e-26
 [5,] -2.901255793 0.112595729  -25.7670146 2.078228e-146
 [6,] -3.016871372 0.272347649  -11.0772808  1.617094e-28
 [7,]  6.372554352 0.149250328   42.6970878  0.000000e+00
 [8,]  2.912942081 0.070171577   41.5117090  0.000000e+00
 [9,] -0.183718726 0.011802995  -15.5654328  1.250215e-54
 [10,] -0.797422839 0.013125852  -60.7520822  0.000000e+00
-----------------------------------------------
-----------------------------------------------
```