



University of  
Stavanger

Faculty of Science and Technology

## MASTER'S THESIS

Study program/Specialization:  
Mathematics and  
Physics/Mathematics

Spring semester, 2016

Open / Restricted access

Writer: Eze Sunday

.....

(Writer's signature)

Faculty supervisor: Assoc. prof. Bjorn Henrik Auestad

External supervisor(s): Not applicable

Thesis title: Computational Statistics and Data Analysis: Robustness of the Linear Mixed Effects Model to Non-normality of Longitudinal Data.

Credits (ECTS): 60

Key words:

Fixed effects

Random effects

Covariates

Response variable

Growth curve model

Covariance structure

Random intercept

Random intercept and random slope

Robustness

Pages: 82

+ enclosure: 1 CD

Stavanger, 14/06/2016

Date/year

UNIVERSITY OF STAVANGER

MASTER THESIS

---

**Computational Statistics and  
Data Analysis: Robustness of the  
Linear Mixed Effects Model to  
Non-normality of Longitudinal  
Data**

---

*Author:*

Eze Sunday

*Supervisor:*

Assoc. prof. B. Auestad

*A thesis research work submitted for completion  
of Master of Science*

*in Mathematics and Physics*

June 2016

# Declaration of Authorship

I, Eze Sunday, declare that this thesis titled, 'Computational Statistics and Data Analysis: Robustness of the Linear Mixed Effects Model to Non-normality of Longitudinal Data' and the work presented in it are mine. I attest that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Stavanger.
- I clearly indicated where I have consulted the published work of other authors.
- Where I have quoted from others, the source is always provided. With the exception of such quotations if any, this thesis is entirely my own work.

Signed:

---

Date:

---

*”Those fields which most depend upon authoritative opinion for their data least contain known natural law.”*

L. Ron Hubbard

# *Abstract*

The Linear mixed effects model is based on one of the assumptions, which is that data are normally distributed as the errors and random effects in the model are assumed to have normal distributions; but in practice, longitudinal data are often either slightly or very skewed. Hence, they have non-normal distributions. This thesis is focused on studying the robustness of the linear mixed effects model to non-normality of error distribution.

The thesis is structured in this way: in the **chapter 1**, the literature review of the linear mixed effects model is presented. **Chapter 2** contains explanations of the statistical tools used in checking the goodness of fit for a fitted model. **Chapter 3** contains description of the research procedures employed in the implementations. **Chapter 4** contains the simulations and results. All in all, conclusions are presented in the **chapter 5**.

# *Acknowledgements*

I thank Assoc. prof. Bjorn Auestad for his guidance and support during this thesis research. He has been motivating and encouraging to ensure that its purpose is met.

I thank all the professors who had taught me throughout my duration of study in the Department of Mathematics and Natural Science, especially Prof. Jan Terje Kvaloy. Their dedications are the seeds of this fruition.

Also, my family members contributed immensely to my success. Many thanks go to my mother for her incomparable love and care.

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Longitudinal Data Analysis . . . . .	1
1.2 The Evolution of Longitudinal Data Analysis . . . . .	3
1.3 The Linear Mixed Effects Model . . . . .	6
1.4 Generality of the Linear Mixed Effects Model . . . . .	7
1.5 Covariance Structure . . . . .	11
1.6 The Distribution of Response Variable, $Y_i$ . . . . .	16
1.7 The Maximum Likelihood Estimation . . . . .	17
1.8 Restricted maximum likelihood estimation . . . . .	19
1.9 Robust Inference . . . . .	20
1.10 The Likelihood Ratio Test . . . . .	21
1.11 The $t$ -test . . . . .	22
1.11.1 The Confidence Interval . . . . .	22
1.12 The Information Criteria . . . . .	22
1.13 The assumptions of the LME Model . . . . .	23
<b>2 Checking an LME Fitted model</b>	<b>25</b>
2.1 The Goodness of Fit of a Model . . . . .	25
2.2 Cholesky Residuals . . . . .	25
2.3 $\chi^2$ Distribution for a Model Check . . . . .	27
2.4 Shapiro-Wilk Test . . . . .	28
<b>3 Research Procedure</b>	<b>29</b>

---

3.1	The Simulation Setup . . . . .	29
3.2	Sampling Process . . . . .	31
3.2.1	Balanced Data . . . . .	31
3.2.2	Unbalanced Data . . . . .	32
3.3	Models for Data Generation . . . . .	33
3.4	RIRS Models . . . . .	34
3.5	RI Models . . . . .	38
<b>4</b>	<b>Simulations and Results</b>	<b>39</b>
4.1	Case 1: Model Validation . . . . .	39
4.1.1	Simulation Settings for Case 1 . . . . .	40
4.2	Case 2: The Corrupted Distribution of Error . . . . .	44
4.2.1	Simulation Settings for Case 2 . . . . .	44
4.3	Case 3: Asymmetric Distribution of Error . . . . .	48
4.3.1	Simulation Settings for Case 3 . . . . .	48
4.4	Case 4: Gamma Distribution of Error . . . . .	52
4.4.1	Simulation Settings for Case 4 . . . . .	52
4.5	Case 5: Exponential Distribution of Error . . . . .	56
4.5.1	Simulation Settings for Case 5 . . . . .	56
<b>5</b>	<b>Conclusions</b>	<b>60</b>
<b>A</b>	<b>Appendix</b>	<b>63</b>
<b>B</b>	<b>Appendix</b>	<b>78</b>
<b>C</b>	<b>Appendix</b>	<b>80</b>
	<b>Bibliography</b>	<b>82</b>



# List of Figures

1.1	Example of the Graphical Representation of Logitudinal Data . . . .	3
2.1	A QQ-Plot for Checking Departure from a Normal Distribution . . .	26
2.2	A QQ-Plot for Checking Departure from a Normal Distribution . . .	27
3.1	The Symmetric Corrupted Normal Distribution of Error . . . . .	35
3.2	The Asymmetric Corrupted Normal Distribution of Error . . . . .	36
3.3	The Skewed Distribution:Gamma . . . . .	37
3.4	The Very Skewed Distribution: Exponential . . . . .	38
4.1	Validation Model: qqnorm Plot for Residues . . . . .	40
4.2	Validation Model: qqnorm Plot for Random Effects . . . . .	41
4.3	Validation Model: Responses against Fitted Values . . . . .	41
4.4	Corrupted Distribution: qqnorm Plot for Residues . . . . .	45
4.5	Corrupted Distribution: qqnorm Plot for Random Effects . . . . .	45
4.6	Corrupted Distribution: Responses against Fitted Values . . . . .	46
4.7	Asymmetric Distribution: qqnorm Plot for Residues . . . . .	49
4.8	Asymmetric Distribution: qqnorm Plot for Random Effects . . . . .	49
4.9	Asymmetric Distribution: Responses against Fitted Values . . . . .	50
4.10	Gamma Distribution: qqnorm Plot for Residues . . . . .	53
4.11	Gamma Distribution: qqnorm Plot for Random Effects . . . . .	53
4.12	Gamma Distribution: Responses against Fitted Values . . . . .	54
4.13	Exponential Distribution: qqnorm Plot for Residues . . . . .	57
4.14	Exponential Distribution: qqnorm Plot for Random Effects . . . . .	57
4.15	Exponential Distribution: Responses against Fitted Values . . . . .	58

*First, to Jehovah, and to my father: Mr. Theophilus Eze.*

# Chapter 1

## Introduction

### 1.1 Longitudinal Data Analysis

The word *longitudinal* is often used to describe outcomes of repeated measures that are carried on objects or persons, who are under a treatment, over time to study their responses to the treatment. Hence, it is more applied to biotechnology and medicine than to engineering.

Longitudinal data analysis plays an important role in the medical field because it is mainly used to study a degree to which a response variable responds to its covariates. The analysis involves follow-up and measurements of the responses of the variable (a unit) to the treatment over time.

Ideally, the measurements of the responses are meant to be repetitively taken in such a way that each unit, under the same treatment, has the same number of measurements taken at the same time points over time; but in practice, it is not always feasible to realize such balanced measurements, so they (repeated measures of data) do not have to assume a balanced form, meaning it is allowed that some units do not have the same number of measurements as others in a sample.

**Definition:** *Longitudinal data* are the repeated measures of changes in responses to a treatment carried on  $N$  units over time, where  $N > 1$  is the total number of units. For example, a research to be conducted to find out the effects of a new treatment on a viral disease such as Zika virus can be carried out through administering a medicine on ten patients and taking the measures of counts of

viral loads over time on each patient. At each individual level, such repeated measures, taken on each individual, yield correlated data that account for each patient's unique responses to the medicine.

The measurements, taken on each patient over time, may be or are typically statistically correlated, but measurements obtained from different patients are independent. A variation of the measurements taken on each patient shows a within-individual variation and a variation of the measurements taken on different individuals (units) shows a between-individual variation [1]. The difference between the two variations is called a random effect, which represents the peculiarity of each individual in a population from which a sample is drawn.

$$\alpha_i = \beta_i - \beta \tag{1.1}$$

$E[\alpha_i] = E[\beta_i] - \beta = 0$  and  $\text{Var}[\alpha_i] = \text{Var}[\beta_i] = \sigma^2$ .  $\alpha_i \sim N(0, \sigma^2)$

For example, let  $i$  be the index for the units in a sample,  $\beta$  be the between-individual variation,  $\beta_i$  be the within-individual variation, and  $\alpha_i$  be the random effect for unit  $i$ , where  $i \geq 1$ . The random effect is defined in the equation 1.1.

It should be noted that  $\alpha_i \sim N(0, \sigma^2)$  is based on a linear-mixed-effects assumption.

In addition, the measurements taken in a closer interval of time are more correlated than the measurements taken in a wider interval of time since the correlation is reduced as the time interval of measurements is widened. This feature of longitudinal data is due to the fact that a characteristic of response, on which the measures are based, randomly changes in time. In other words, the responses of the units are random as the units are random variables.

As shown in the figure 1.1, the difference between each measured response value and its corresponding value on a linear model is the error of the measurement,  $e_{ij}$ ; where  $Y_{ij}$  is the response variable,  $\mu_{ij}$  is its corresponding mean value on the regression line,  $i$  is the index for the number of patients (units) in the study and  $j$  is the index for the number of the response data obtained from each patient over time, so  $\mu_{ij}$  is the expected value for unit  $i$  at time  $j$ .

$$e_{ij} = Y_{ij} - \mu_{ij} \tag{1.2}$$

$E[e_{ij}] = E[Y_{ij}] - \mu_{ij} = 0$  and  $\text{Var}[e_{ij}] = \sigma_e^2$ , so  $e_{ij} \sim N(0, \sigma_e^2)$  (assumption).

It should be noted that  $e_{ij} \sim N(0, \sigma_e^2)$  is based on the linear-mixed-effects, LME, assumption.

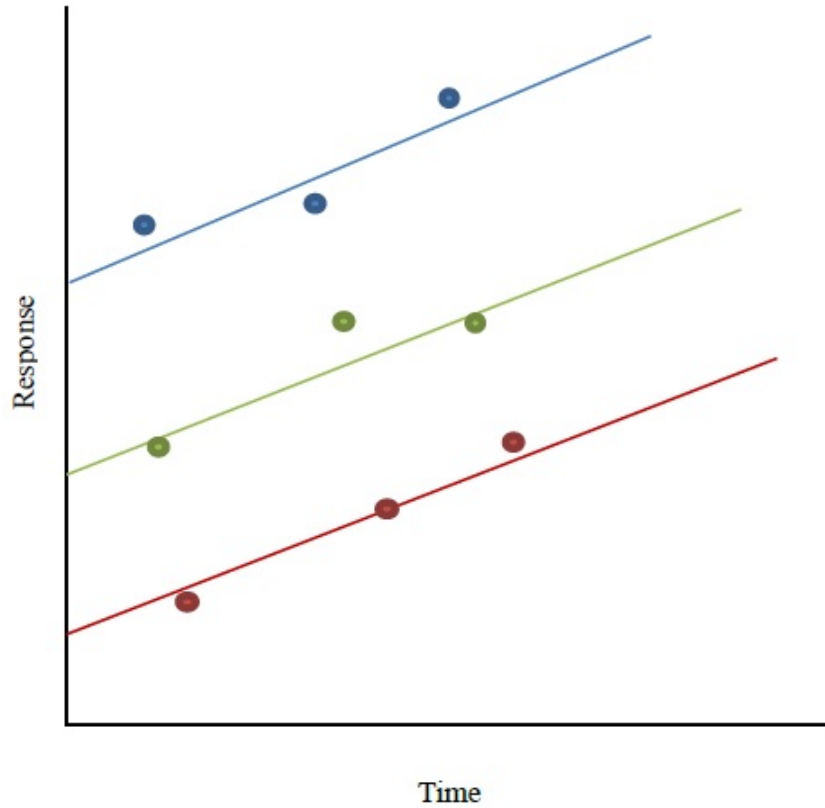


FIGURE 1.1: Example of the Graphical Representation of Logitudinal Data

## 1.2 The Evolution of Longitudinal Data Analysis

The concept of longitudinal data analysis is derived from the traditional regression analysis where the main goal is to model the relationship between a dependent variable and independent variables. The independent variables are often called the predictors because they determine a responsiveness of the dependent variable while the dependent variable itself is called the response variable.

$$Y_{ij} = \beta_0 + \beta_1 X_{1ij} + \beta_2 X_{2ij} + \beta_3 X_{3ij} + \dots + \beta_p X_{pij} + e_{ij} \quad (1.3)$$

The relationship between a dependent variable and independent variables is as shown in the equation 1.3. Where  $Y_{ij}$ —the response of unit  $i$  at  $j$  time;  $X_{1ij}, X_{2ij}, X_{3ij} \dots X_{pij}$  are the predictor variables,  $\beta_0, \beta_1, \beta_2, \beta_3, \dots, \beta_p$  are the fixed effects, where  $p$  is the total number of the fixed effects in a model; and  $e_{ij}$  are the errors. The essence of the fixed effects in the model is to quantify the impact of each predictor variable on a response variable.

As presented in the definition of longitudinal data analysis, let's suppose  $N$  individuals' counts of viral loads were measured at three different time points over time; hence,  $\mathbf{Y}_i = [Y_{i1}, Y_{i2}, Y_{i3}]^T$ , where  $i = 1 \dots N$  and  $j = 1 \dots n_i$ . The three repeated measures on each unit of  $N$  units over time is written as:

$$Y_{i1} = [1, X_{1i1}, X_{2i1}, X_{3i1}, \dots, X_{pi1}] \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} + e_{i1} \quad (1.4)$$

$$Y_{i2} = [1, X_{1i2}, X_{2i2}, X_{3i2}, \dots, X_{pi2}] \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} + e_{i2} \quad (1.5)$$

$$Y_{i3} = [1, X_{1i3}, X_{2i3}, X_{3i3}, \dots, X_{pi3}] \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} + e_{i3} \quad (1.6)$$

Where  $Y_{i1}$  is the response measurement of unit  $i$  at the time point,  $j = 1$ ;  $Y_{i2}$  is the response measurement on the same unit at a second successive time point,  $j = 2$ ;  $Y_{i3}$  is the response measurement on the unit at a third successive time

point,  $j = 3$ . In this case, the total measures for a unit  $i$ ,  $n_i$ , is three (3); that is  $n_i = 3$ .

It should be noted that  $n_i$ , the number of measurements taken on the unit  $i$ , is dependent on the unit  $i$ . That is  $n_i$  may be different for different units.

$$\mathbf{Y}_i = \begin{bmatrix} 1 & X_{1i1} & X_{2i1} & X_{3i1} & \dots & X_{pi1} \\ 1 & X_{1i2} & X_{2i2} & X_{3i2} & \dots & X_{pi2} \\ 1 & X_{1i3} & X_{2i3} & X_{3i3} & \dots & X_{pi3} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix} \quad (1.7)$$

Each individual responses to a treatment is written in the matrix form as shown in the equation 1.7.

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{e}_i \quad (1.8)$$

Where

$$\mathbf{Y}_i = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix}, \mathbf{X}_i = \begin{bmatrix} 1 & X_{1i1} & X_{2i1} & X_{3i1} & \dots & X_{pi1} \\ 1 & X_{1i2} & X_{2i2} & X_{3i2} & \dots & X_{pi2} \\ 1 & X_{1i3} & X_{2i3} & X_{3i3} & \dots & X_{pi3} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} \text{ and } \mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix}.$$

Then the equation 1.7 can be summarized as equation 1.8. That means that there may be different design matrix,  $\mathbf{X}_i$ , and vector of errors,  $\mathbf{e}_i$ , for different units,  $i$ , while the vector of fixed effects,  $\boldsymbol{\beta}$ , remains the same for each unit of a sample in a longitudinal data analysis.

Statistically, the measurements that are obtained from each unit are clustered since they are generally correlated. Hence, there are  $N$  clusters of data [2]. The covariance matrix for  $\mathbf{Y}_i$  is as shown in the equation 1.9.

$$\text{Var}(\mathbf{Y}_i) = \begin{bmatrix} \text{Var}(Y_{i1}) & \text{Cov}(Y_{i1}, Y_{i2}) & \text{Cov}(Y_{i1}, Y_{i3}) \\ \text{Cov}(Y_{i2}, Y_{i1}) & \text{Var}(Y_{i2}) & \text{Cov}(Y_{i2}, Y_{i3}) \\ \text{Cov}(Y_{i3}, Y_{i1}) & \text{Cov}(Y_{i3}, Y_{i2}) & \text{Var}(Y_{i3}) \end{bmatrix} \quad (1.9)$$

### 1.3 The Linear Mixed Effects Model

$$Y_{ij} = \beta_0 + \alpha_{0i} + \beta_1 t_{ij} + \alpha_{1i} t_{ij} + \beta_2 X_i + e_{ij} \quad (1.10)$$

$\alpha_{0i} \sim N(0, \sigma_0^2)$ ,  $\alpha_{1i} \sim N(0, \sigma_1^2)$ ,  $e_{ij} \sim N(0, \sigma_e^2)$ ;  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, n_i$ .

In the linear mixed effects model, the fixed effects and random effects are combined in a model to capture variabilities around the mean of responses of the units of a sample, drawn from a population. The regression model that is considered so far is a typical example of a fixed-effects model because the variability of each unit is not considered. A typical example of LME model is the simple growth curve model, which is shown in the equation 1.10.

The growth curve model is an example of a random intercept and random slope, RIRS, model because it comprises the two random effects: random intercept ( $\alpha_{0i}$ ) and random slope ( $\alpha_{1i}$ ). When random slope is omitted from the model, it is then reduced to a random intercept, RI, model.

The conceptual idea of the linear mixed effects model is to incorporate a variability of a units' responses to a treatment in the model. As given in the previous example, let's suppose that three measurements of count of viral load of Zika virus in a patient who is subjected to a follow-up are measured over time. The three measures, plotted against time, give the regression model for his responses to the treatment. The regression model obtained from the three measurements taken over time does not include the variability of unit's responses to the administered medicine.

The use of linear mixed effects model becomes crucial when there is a need to model the variabilities of units' responses to a treatment.

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \mathbf{b}_i + \mathbf{e}_i \quad (1.11)$$



The model shown in the equation 1.11 is a typical linear-mixed-effects model.  $n_i$  is the number of measurements taken on a unit  $i$ . Where  $\mathbf{Y}_i$  is  $n_i \times 1$  vector of responses,  $\mathbf{X}_i$  is  $n_i \times p$  matrix of covariates,  $\boldsymbol{\beta}$  is  $p \times 1$  vector of fixed effects,  $\mathbf{Z}_i$  is  $n_i \times q$  matrix of covariates associated with random effects,  $\mathbf{b}_i$  is  $q \times 1$  vector of random effects, where  $q$  is the number of covariates associated with the random effects;  $\mathbf{e}_i$  is  $n_i \times 1$  vector of errors of  $n_i$  measurements.

Using the equation 1.11 and three (3) measures over time per unit, the vectors assume these forms:

$$\mathbf{X}_i = \begin{bmatrix} 1 & t_{i1} & X_i \\ 1 & t_{i2} & X_i \\ 1 & t_{i3} & X_i \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}, \mathbf{Z}_i = \begin{bmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ 1 & t_{i3} \end{bmatrix}, \mathbf{b}_i = \begin{bmatrix} \alpha_{0i} \\ \alpha_{1i} \end{bmatrix}, \mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix}.$$

## 1.4 Generality of the Linear Mixed Effects Model

The linear mixed effects model, shown in the equation 1.11, is a model for a two-level repeated measures in which the fixed effects and random effects are crossed as a level of the factor is measured across multilevels of another factor. For instance, the responses that are measured across the units, as given in the previous example with Zika virus, is the two-level repeated measures with crossed fixed effects and random effects. For generality,  $Y_i$ , the responses for unit  $i$ , can be extended to

$$\mathbf{Y}_i = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \\ \vdots \\ Y_{in_i} \end{bmatrix} \sim MVN(\mathbf{X}_i\boldsymbol{\beta}, \mathbf{V}_i),$$

As previously stated, the number of measures in a unit,  $n_i$ , depends on unit  $i$ . Hence, the total number of observations,  $n = \sum_{i=1}^N n_i$ . Where  $N$  is the total number of units.

$$\mathbf{X}_i = \begin{bmatrix} 1 & X_{1i1} & X_{2i1} & X_{3i1} & \dots & X_{pi1} \\ 1 & X_{1i2} & X_{2i2} & X_{3i2} & \dots & X_{pi2} \\ 1 & X_{1i3} & X_{2i3} & X_{3i3} & \dots & X_{pi3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & X_{1in_i} & X_{2in_i} & X_{3in_i} & \dots & X_{pin_i} \end{bmatrix},$$

The  $n_i \times p$  matrix of covariates,  $\mathbf{X}_i$ , contains  $p$  number of covariates that are uniquely associated with the  $p$  number of fixed effects in an LME model. The first column of the matrix is usually assigned a vector of ones to recover the fixed effect,  $\beta_0$ , in the model.

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix},$$

The fixed-effects vector,  $\boldsymbol{\beta}$ , contains  $p$  number of fixed effect associated with the covariates arranged in the rows of the matrix of covariates,  $\mathbf{X}_i$ .

$$\mathbf{Z}_i = \begin{bmatrix} 1 & X_{1i1} & X_{2i1} & \dots & X_{qi1} \\ 1 & X_{1i2} & X_{2i2} & \dots & X_{qi2} \\ 1 & X_{1i3} & X_{2i3} & \dots & X_{qi3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1in_i} & X_{2in_i} & \dots & X_{qin_i} \end{bmatrix},$$

The  $n_i \times q$  matrix,  $\mathbf{Z}_i$ , contains covariates that are associated with random effects in the model. Hence  $q \leq p$  in the LME models.

The  $q \times 1$  column vector of random effects,  $\mathbf{b}_i$ , contains random effects for some covariates in the design matrix  $\mathbf{X}_i$ . The column vector of random effect is defined by

$$\mathbf{b}_i = \begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i3} \\ \vdots \\ \alpha_{iq} \end{bmatrix},$$

Under the LME assumptions that the random effects are independent of each other,  $\text{Var}(\mathbf{b}_i) = \boldsymbol{\Sigma}$ , reduces to a diagonal positive definite matrix and the  $\mathbf{b}_i$  follows a multivariate normal distribution with mean vector of zeros and the variance,  $\boldsymbol{\Sigma}$ .

$$\mathbf{b}_i \sim \text{MVN}(\mathbf{0}, \Sigma).$$

The covariance of  $\mathbf{b}_i$  is defined by

$$\Sigma = \text{Cov}(\mathbf{b}_i) = \begin{bmatrix} \text{Var}(\alpha_{i1}) & \text{Cov}(\alpha_{i1}, \alpha_{i2}) & \dots & \text{Cov}(\alpha_{i1}, \alpha_{iq}) \\ \text{Cov}(\alpha_{i2}, \alpha_{i1}) & \text{Var}(\alpha_{i2}) & \dots & \text{Cov}(\alpha_{i2}, \alpha_{iq}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(\alpha_{iq}, \alpha_{i1}) & \text{Cov}(\alpha_{iq}, \alpha_{i2}) & \dots & \text{Var}(\alpha_{iq}) \end{bmatrix} \quad (1.12)$$

For example, the matrix,  $\mathbf{Z}_i$ , reduces to a column vector of ones for the random intercept, RI, model. In the case of random intercept and random slope, RIRS, model with an intercept and a covariate respectively associated with a random effect for intercept and a random effect for the covariate, the  $\mathbf{Z}_i$  matrix reduces to:

$$\mathbf{Z}_i = \begin{bmatrix} 1 & X_{2i1} \\ 1 & X_{2i2} \\ 1 & X_{2i3} \\ \vdots & \vdots \\ 1 & X_{2in_i} \end{bmatrix},$$

The column matrix for the two random effects in the model reduces to:

$$\mathbf{b}_i = \begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \end{bmatrix},$$

The covariance matrix of random effects,  $\mathbf{b}_i$ , reduces to:

$$\Sigma = \text{Cov}(\mathbf{b}_i) = \begin{bmatrix} \text{Var}(\alpha_{i1}) & \text{Cov}(\alpha_{i1}, \alpha_{i2}) \\ \text{Cov}(\alpha_{i2}, \alpha_{i1}) & \text{Var}(\alpha_{i2}) \end{bmatrix}$$

In the case of the random effects being independent, the covariances become zeros and the matrix further reduces in number of parameter to:

$$\Sigma = \text{Cov}(\mathbf{b}_i) = \begin{bmatrix} \text{Var}(\alpha_{i1}) & 0 \\ 0 & \text{Var}(\alpha_{i2}) \end{bmatrix}$$

The errors,  $\mathbf{e}_i$ , which determine the growth curve path of a unit  $i$  around the mean,  $\boldsymbol{\mu}$ , is defined by

$$\mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \\ \vdots \\ e_{in_i} \end{bmatrix},$$

The errors,  $\mathbf{e}_i$ , are based on three LME assumptions: in one case, they are assumed to be uncorrelated; in the second case, they are assumed to be homoscedastic, meaning they have the same variance. In the third case, they are assumed to follow multivariate normal distribution with mean vector of zeros and the covariance matrix,  $\mathbf{H}_e$ .

$$\mathbf{e}_i \sim MVN(\mathbf{0}, \mathbf{H}_e),$$

Generally,

$$\mathbf{H}_e = \text{Var}(\mathbf{e}_i) = \begin{bmatrix} \text{Var}(e_{i1}) & \text{Cov}(e_{i1}, e_{i2}) & \dots & \text{Cov}(e_{i1}, e_{in_i}) \\ \text{Cov}(e_{i2}, e_{i1}) & \text{Var}(e_{i2}) & \dots & \text{Cov}(e_{i2}, e_{in_i}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(e_{in_i}, e_{i1}) & \text{Cov}(e_{in_i}, e_{i2}) & \dots & \text{Var}(e_{in_i}) \end{bmatrix} \quad (1.13)$$

For the assumption of uncorrelated errors, the covariance matrix of errors,  $H_e$ , takes the form

$$\mathbf{H}_e = \text{Var}(\mathbf{e}_i) = \begin{bmatrix} \text{Var}(e_{i1}) & 0 & \dots & 0 \\ 0 & \text{Var}(e_{i2}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \text{Var}(e_{in_i}) \end{bmatrix}$$

The vector of random effects,  $\mathbf{b}_i$ , and the vector of errors,  $\mathbf{e}_i$ , are assumed to be independent [3].

The vector of errors,  $\mathbf{e}_i$ , is dependent on  $i$ . This means that the number of errors in the vector corresponds with the number of measures carried on the unit. Hence, the vector of errors for the three measurements, as given in the example, reduces to:

$$\mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix}.$$

## 1.5 Covariance Structure

A covariance structure refers to the structure that determines the number of covariance parameters that are in a model. The covariance parameters are stored in a vector called vector of covariance parameters,  $\boldsymbol{\theta}$ .

The two most often used covariance structures for a positive definite symmetric matrix of a vector of random effects,  $\Sigma$  shown in equation 1.12, are the unstructured and structured covariance structure. Let's suppose that two random effects are used in a model, where the vector of random effects is defined by

$$\mathbf{b}_i = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

The unstructured covariance structure for the two-random-effects model is defined by

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} \quad (1.14)$$

The vector of covariance parameters,  $\boldsymbol{\theta}_\Sigma$ , for the covariance structure is defined by

$$\boldsymbol{\theta}_\Sigma = \begin{bmatrix} \sigma_1^2 \\ \sigma_{12} \\ \sigma_2^2 \end{bmatrix} \quad (1.15)$$

The vector of covariance parameters, shown in the equation 1.15, contains three elements because the covariances between the two random effects are the same, so the number of the elements contained in the vector is three.

A structured covariance structure of the two-random-effects model is defined by

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \quad (1.16)$$

The structure (structured covariance structure) is simpler because of independency between the two random effects. Such structure in which the covariance elements of a positive definite matrix are zeros except variance elements is called diagonal matrix. The variance elements of the matrix are not equal.

The corresponding vector of covariance parameters,  $\theta_\Sigma$ , is defined by

$$\theta_\Sigma = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \end{bmatrix} \quad (1.17)$$

The vector contains only two elements in this case.

If the variances are assumed equal, the diagonal matrix reduces to a multiple of an identity matrix, which is defined by

$$\Sigma = \sigma_1^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \sigma_2^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.18)$$

The corresponding vector of covariance parameters,  $\theta_\Sigma$ , is defined by

$$\theta_\Sigma = \begin{bmatrix} \sigma_1^2 \end{bmatrix} \quad (1.19)$$

Moreover, for a model in which there are nested random effects in a random effect, a compound-symmetry or diagonal-block structure may represent a true covariance structure. Let's suppose that there are  $i$  units in which  $j$  subcomponents are nested, where  $j = 1, 2, 3$ ; each unit has a random effects,  $\alpha_i$ , and each subcomponent nested within the unit has the random effect,  $\alpha_{i,j}$ , where  $\alpha_i \sim N(0, \sigma_1)$  and  $\alpha_{i,j} \sim N(0, \sigma_2)$ . The vector of the random effects for unit  $i$  could be defined by

$$\mathbf{b}_i = \begin{bmatrix} \alpha_i + \alpha_{i,1} \\ \alpha_i + \alpha_{i,2} \\ \alpha_i + \alpha_{i,2} \end{bmatrix} \quad (1.20)$$

Under the assumption of independency among the random effects and the same variance,  $\sigma_1^2$ , for the random effects of unit and the same variance,  $\sigma_2^2$ , for the random effects of subcomponents, the covariance structure is defined by

$$\Sigma = \begin{bmatrix} \sigma_1^2 + \sigma_2^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_2^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma_2^2 \end{bmatrix} \quad (1.21)$$

The covariance structure (1.21) is called compound-symmetry [4]. Its corresponding vector of covariance parameters,  $\theta_\Sigma$ , is defined by

$$\theta_\Sigma = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \end{bmatrix} \quad (1.22)$$

The vector of the random effects,  $\mathbf{b}_i$ , could also be defined by

$$\mathbf{b}_i = \begin{bmatrix} \alpha_i \\ \alpha_{i,1} \\ \alpha_{i,2} \\ \alpha_{i,2} \end{bmatrix} \quad (1.23)$$

The block-diagonal structure for the vector (1.23) is defined by

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 & \sigma_2^2 \end{bmatrix} \quad (1.24)$$

The covariance structure (1.24) is derived from these assumptions: the random effects are independent; the variances for the units are same,  $\sigma_1^2$ ; the variances for the subcomponents nested within the units are also the same,  $\sigma_2^2$ . The covariance structure is represented as block diagonals such as  $\sigma_1^2$  and  $\sigma_2^2\mathbb{I}$ . The identity matrix,  $\mathbb{I}$ , is a  $j \times j$  matrix, where  $j = 3$  in this case. Its vector of covariance parameters is the same with that of the compound-symmetry structure (1.22).

The block-diagonal covariance matrix may take other forms depending on a vector of random effects,  $\mathbf{b}_i$ . If a vector of random effects of a model is defined by

$$\mathbf{b}_i = \begin{bmatrix} \alpha_i \\ \alpha_{i,j} \\ \alpha_{i,k} \\ \alpha_{i,l} \end{bmatrix} \quad (1.25)$$

where  $j = 1, 2, 3, 4$ ,  $k = 1, 2, 3$ , and  $l = 1, 2, 3$ ;  $\alpha_i \sim N(0, \sigma_1^2)$ ,  $\alpha_{ij} \sim N(0, \sigma_2^2)$ ,  $\alpha_{ik} \sim N(0, \sigma_3^2)$ , and  $\alpha_{il} \sim N(0, \sigma_4^2)$ . The block-diagonal covariance structure takes a form that is different from the first block-diagonal structure (1.24). The covariance structure for the vector of random effects, (1.25), becomes

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_3^2 \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_4^2 \mathbb{I} \end{bmatrix} \quad (1.26)$$

There are different orders of the identity matrixes in the covariance structure (1.26):  $\sigma_2^2 \mathbb{I}$  has a  $j \times j$  order of identity matrix,  $\sigma_3^2 \mathbb{I}$  has a  $k \times k$  order of identity matrix, and  $\sigma_4^2 \mathbb{I}$  has an  $l \times l$  order of identity matrix. Its vector of covariance parameter has four parameters as defined by

$$\boldsymbol{\theta}_\Sigma = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \sigma_3^2 \\ \sigma_4^2 \end{bmatrix} \quad (1.27)$$

Furthermore, the two most used covariance structures for the positive definite symmetric matrix,  $\mathbf{e}_i$  shown in the equation 1.13, are diagonal symmetric structure and compound symmetric structure.

A diagonal structure contains the same variance along the diagonal of the covariance matrix and zeros in other parts of the matrix. The reason for the diagonality is due to the assumption that the errors are not correlated and they have the same variance. Hence, the diagonal structure assumes this form for a single unit  $i$ :



$$\mathbf{H}_e = \text{Var}(\mathbf{e}_i) = \begin{bmatrix} \sigma_e^2 & 0 & \dots & 0 \\ 0 & \sigma_e^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_e^2 \end{bmatrix} \quad (1.28)$$

The vector of the covariance parameters,  $\boldsymbol{\theta}_{\mathbf{H}_e}$ , contains only one parameter,

$$\boldsymbol{\theta}_{\mathbf{H}_e} = \sigma_e^2. \quad (1.29)$$

A compound symmetry structure is derived from the combination of the diagonal structure (symmetric covariance structure), which is based on the assumptions that errors are uncorrelated and homoscedastic and the unsymmetric covariance structure, which is based on the assumptions that the errors are correlated. Hence, the compound symmetric structure is obtained from the combination of both structures.

$$\begin{bmatrix} \sigma_e^2 & 0 & \dots & 0 \\ 0 & \sigma_e^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_e^2 \end{bmatrix} + \begin{bmatrix} \sigma_1 & \sigma_1 & \dots & \sigma_1 \\ \sigma_1 & \sigma_1 & \dots & \sigma_1 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1 & \sigma_1 & \dots & \sigma_1 \end{bmatrix} = \begin{bmatrix} \sigma_e^2 + \sigma_1 & \sigma_1 & \dots & \sigma_1 \\ \sigma_1 & \sigma_e^2 + \sigma_1 & \dots & \sigma_1 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1 & \sigma_1 & \dots & \sigma_e^2 + \sigma_1 \end{bmatrix}$$

Thus, the compound symmetric structure for the covariance matrix of a vector of errors is

$$\mathbf{H}_e = \text{Var}(\mathbf{e}_i) = \begin{bmatrix} \sigma_e^2 + \sigma_1 & \sigma_1 & \dots & \sigma_1 \\ \sigma_1 & \sigma_e^2 + \sigma_1 & \dots & \sigma_1 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1 & \sigma_1 & \dots & \sigma_e^2 + \sigma_1 \end{bmatrix} \quad (1.30)$$

The vector of covariance parameters,  $\boldsymbol{\theta}_{\mathbf{H}_e}$ , for the compound symmetric structure is

$$\boldsymbol{\theta}_{\mathbf{H}_e} = \begin{bmatrix} \sigma_e^2 \\ \sigma_1 \end{bmatrix} \quad (1.31)$$

The diagonal structure and structured covariance structure are based on homoscedastic assumption. In the case of the heteroscedastic assumption, the variances and covariances may differ, which impose structures that are different from the ones presented here.

The vectors of covariance parameters,  $\boldsymbol{\theta}_\Sigma$  and  $\boldsymbol{\theta}_{H_e}$ , are contained in a vector,  $\boldsymbol{\theta}$ ; that is,

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_\Sigma \\ \boldsymbol{\theta}_{H_e} \end{bmatrix} \quad (1.32)$$

## 1.6 The Distribution of Response Variable, $Y_i$

The marginal distribution of the vector variable of responses,  $\mathbf{Y}_i$ , can be derived from the equation 1.11. The equation can be presented as,

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{R}_i \quad (1.33)$$

where

$$\mathbf{R}_i = \mathbf{Z}_i\mathbf{b}_i + \mathbf{e}_i$$

As both  $\mathbf{b}_i$  and  $\mathbf{e}_i$  are assumed to follow multivariate normal distributions with zero mean vector,  $\mathbf{b}_i \sim \text{MVN}(\mathbf{0}, \Sigma)$  and  $\mathbf{e}_i \sim \text{MVN}(\mathbf{0}, \mathbf{H}_e)$ . Where  $\mathbf{b}_i$  and  $\mathbf{e}_i$  are assumed independent. The expectation of  $\mathbf{R}_i$  is

$$\text{E}[\mathbf{R}_i] = \mathbf{Z}_i \mathbf{0} + \mathbf{0} = \mathbf{0}$$

The variance of  $\mathbf{R}_i$ ,  $\mathbf{V}_i$ , is

$$\text{Var}[\mathbf{R}_i] = \mathbf{Z}_i \text{Cov}(\mathbf{b}_i) \mathbf{Z}_i^T + \text{Cov}(\mathbf{e}_i)$$

Then

$$\mathbf{V}_i = \text{Var}[\mathbf{R}_i] = \mathbf{Z}_i \sum \mathbf{Z}_i^T + \mathbf{H}_e \quad (1.34)$$

Hence,  $\mathbf{R}_i \sim \text{MVN}(\mathbf{0}, \mathbf{V}_i)$  and  $\mathbf{Y}_i \sim \text{MVN}(\mathbf{X}_i\boldsymbol{\beta}, \mathbf{V}_i)$

## 1.7 The Maximum Likelihood Estimation

The response vector variable for unit  $i$ ,  $\mathbf{Y}_i$ , which is given in the equation 1.11 has a multivariate normal probability distribution function.

$$f(\mathbf{y}_i; \boldsymbol{\beta}) = \left(\frac{1}{2\pi}\right)^{-\frac{n_i}{2}} |\mathbf{V}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}((\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}))} \quad (1.35)$$

Where the  $\mathbf{V}_i$  is given in the equation 1.34.

The maximum likelihood function,  $L_i(\boldsymbol{\beta}; \boldsymbol{\theta})$ , for observation  $\mathbf{y}_i$  of the response vector variable,  $\mathbf{Y}_i$ , for unit  $i$  becomes,

$$L_i(\boldsymbol{\beta}; \boldsymbol{\theta}) = \left(\frac{1}{2\pi}\right)^{-\frac{n_i}{2}} |\mathbf{V}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}((\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}))}$$

As the units are independent, the joint likelihood function for all the units becomes

$$\begin{aligned} L(\boldsymbol{\beta}; \boldsymbol{\theta}) &= \prod_{i=1}^N L_i(\boldsymbol{\beta}; \boldsymbol{\theta}) \\ &= \prod_{i=1}^N \left(\frac{1}{2\pi}\right)^{-\frac{n_i}{2}} |\mathbf{V}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}((\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}))} \end{aligned}$$

The log likelihood function for the joint likelihood function becomes

$$l(\boldsymbol{\beta}; \boldsymbol{\theta}) = -\frac{1}{2}n \ln(2\pi) - \frac{1}{2} \sum_{i=1}^N \ln(|\mathbf{V}_i|) - \frac{1}{2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) \quad (1.36)$$

Where the total number of data,  $n = \sum_{i=1}^N n_i$ .

The score of the log-likelihood function,  $U(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$  when  $\boldsymbol{\theta}$  is constant, is

$$U(\boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) + \frac{1}{2} \sum_{i=1}^N \mathbf{X}_i \mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T.$$

The score function,  $U(\boldsymbol{\beta})$ , is zero when the fixed-effects vector,  $\boldsymbol{\beta}$ , is very close to its maximum value or at the maximum value. Hence, estimator for the fixed-effects vector,  $\tilde{\boldsymbol{\beta}}$ , is obtained by setting  $U(\boldsymbol{\beta}) = 0$ .

$$\tilde{\boldsymbol{\beta}} = \left( \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1} \mathbf{X}_i \right)^{-1} \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1} \mathbf{y}_i \quad (1.37)$$

The estimator of the fixed effects,  $\tilde{\boldsymbol{\beta}}$ , is unbiased as its expectation is the same with the estimate for the fixed effects,  $E[\tilde{\boldsymbol{\beta}}] = \boldsymbol{\beta}$ . This can be shown by replacing the expectation of the response variable,  $E[\mathbf{y}_i]$ , with  $\mathbf{X}_i\boldsymbol{\beta}$  in the model, shown in the equation 1.37.

The next is estimation of  $\boldsymbol{\theta}$ . The parameters in the vector are estimated using numerical approach as it ( $\boldsymbol{\theta}$ ) cannot be expressed in a closed form; because of the reason, a profiled log likelihood is constructed by replacing  $\boldsymbol{\beta}$  with its estimate, as shown in the model of equation 1.37, in the log likelihood function. Then the profiled log likelihood function becomes the function of a vector of covariance parameters.

$$l(\boldsymbol{\theta}) = -\frac{1}{2} \ln(2\pi) \sum_{i=1}^N n_i - \frac{1}{2} \sum_{i=1}^N \ln(|\mathbf{V}_i|) - \frac{1}{2} \sum_{i=1}^N \mathbf{e}_i^T \mathbf{V}_i^{-1} \mathbf{e}_i \quad (1.38)$$

where

$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{X}_i \left( \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1} \mathbf{X}_i \right)^{-1} \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1} \mathbf{y}_i \quad (1.39)$$

The estimation of a vector of covariance parameters,  $\boldsymbol{\theta}$ , is done by applying numerical optimization: the optimization concept implemented in R by Pinheiro (2010) used a combination of the expectation-maximization method and the Newton-Raphson method to find estimate of  $\boldsymbol{\theta}$ ,  $\tilde{\boldsymbol{\theta}}$ , using the profiled log likelihood function.

The expectation-maximization method is based on an iterative method: the vector of covariances,  $\boldsymbol{\theta}$ , is firstly calculated. Then, the calculated vector is inserted in the log-likelihood function to find the expectation of the (log likelihood) function.

The expectation of the log likelihood function is derived and the derived function is maximized to obtain  $\boldsymbol{\theta}^k$ . The steps are repeated iteratively until the 25 iterations are elapsed. Where  $1 \leq k \leq 25$ .

The next step of the method is to use the vector,  $\boldsymbol{\theta}^k$ , obtained after the twenty-five iterations to optimize the estimates of the parameters contained in the vector by using the Newton-Raphson method [5].

$$\boldsymbol{\theta}^m = \boldsymbol{\theta}_k^{m-1} + \frac{U(\boldsymbol{\theta}_k^{m-1})}{U'(\boldsymbol{\theta}_k^{m-1})} \quad (1.40)$$

Where  $U(\boldsymbol{\theta}_k)$  is the score vector and  $U'(\boldsymbol{\theta}_k)$  is a partial derivative of the score vector, which is the same with second derivative of the log likelihood function. The iteration is repeated until convergence in the vector of covariance parameters occurs; that is  $\boldsymbol{\theta}^m$  is approximately the same with  $\tilde{\boldsymbol{\theta}}$  at the iteration at which the convergence occurs.

Moreover, the vector of covariance parameters is unpacked and its elements are correspondingly arrayed in  $\mathbf{H}_e$  and  $\sum$  to calculate the  $\mathbf{V}_i$ .

$$\tilde{\mathbf{V}}_i = \mathbf{Z}_i \sum \mathbf{Z}_i^T + \tilde{\mathbf{H}}_e \quad (1.41)$$

Then the fixed effects can be estimated:

$$\tilde{\boldsymbol{\beta}} = \left( \sum_{i=1}^N \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \mathbf{X}_i \right)^{-1} \sum_{i=1}^N \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \mathbf{y}_i \quad (1.42)$$

The variance of the fixed-effects vector,  $\text{Var}(\tilde{\boldsymbol{\beta}})$ , can be estimated using the equation 1.42. It is defined by

$$\text{Var}(\tilde{\boldsymbol{\beta}}) = \mathbf{Q}_i \left( \sum_{i=1}^N \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \text{Var}(\mathbf{y}_i) \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \right) \mathbf{Q}_i^T \quad (1.43)$$

Where the symmetric matrix  $\mathbf{Q}_i = \left( \sum_{i=1}^N \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \mathbf{X}_i \right)^{-1}$ .

The equation 1.43 can be reduced to:

$$\text{Var}(\tilde{\boldsymbol{\beta}}) = \mathbf{Q}_i \mathbf{Q}_i^{-1} \text{Var}(\mathbf{y}_i) \tilde{\mathbf{V}}_i^{-1} \mathbf{Q}_i^T = \mathbf{Q}_i^T$$

Thus, the variance of the estimator (1.42) is defined by

$$\text{Var}(\tilde{\boldsymbol{\beta}}) = \left( \sum_{i=1}^N \mathbf{X}_i^T \tilde{\mathbf{V}}_i^{-1} \mathbf{X}_i \right)^{-1} \quad (1.44)$$

## 1.8 Restricted maximum likelihood estimation

The restricted maximum likelihood estimation method is used to estimate unbiased covariance parameters contained in the vector,  $\boldsymbol{\theta}$ . This method achieved

unbiasedness of the covariance parameters through compensating the lost degrees of freedom that occurred during estimation of the fixed effects ( $\boldsymbol{\beta}$ ) and adding an additional term to the log likelihood function. That is

$$l(\boldsymbol{\theta}) = -\frac{1}{2}(n-p) \ln(2\pi) - \frac{1}{2} \sum_{i=1}^N \ln(|\mathbf{V}_i|) - \frac{1}{2} \sum_{i=1}^N \mathbf{e}_i^T \mathbf{V}_i^{-1} \mathbf{e}_i - \frac{1}{2} |\mathbf{X}_i^T \mathbf{V}_i^{-1} \mathbf{X}_i| \quad (1.45)$$

Where the column vector of errors,  $\mathbf{e}_i$ , is shown in the equation 1.39. The expectation-maximization method and the Newton-Raphson method are applied to the restricted log likelihood to estimate the covariance parameters of the vector,  $\boldsymbol{\theta}$ . Then, the vector is unpacked and its parameters are correspondingly arrayed in  $\mathbf{H}_e$  and  $\Sigma$  to calculate  $\mathbf{V}_i$  (equation 1.41). Then,  $\mathbf{V}_i$  is used to estimate  $\boldsymbol{\beta}$  as shown in the maximum likelihood method. The estimate of  $\boldsymbol{\beta}$  from this method is called restricted maximum likelihood estimate, REML, of fixed effects.

## 1.9 Robust Inference

The estimation of the variance of fixed-effects vector,  $\boldsymbol{\beta}$ , shown in the equation 1.44 is dependent on the covariance matrix,  $\mathbf{V}_i$ . Hence, the correct specification of covariance structure of the matrix,  $\mathbf{V}_i$ , contributes to an unbiasedness of the estimator,  $\text{Var}(\tilde{\boldsymbol{\beta}})$ .

For robust inference, Liang and Zeger (1986) proposed sandwich estimator. The estimator is obtained by replacing  $\text{Var}(\tilde{\mathbf{y}}_i)$  with  $\mathbf{e}_i \mathbf{e}_i^T$  in the equation 1.43, where  $\mathbf{e}_i = \mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta}$  [6].

$$\text{Var}(\tilde{\boldsymbol{\beta}}(\tilde{\boldsymbol{\theta}})) = \mathbf{Q}_i \left( \sum_{i=1}^N \mathbf{X}_i^T \mathbf{V}_i^{-1}(\tilde{\boldsymbol{\theta}}) \mathbf{e}_i \mathbf{e}_i^T \mathbf{X}_i^T \mathbf{V}_i^{-1}(\tilde{\boldsymbol{\theta}}) \right) \mathbf{Q}_i^T \quad (1.46)$$

Where  $\tilde{\boldsymbol{\theta}}$  is the REML estimates for the parameters in the vector,  $\boldsymbol{\theta}$ .

## 1.10 The Likelihood Ratio Test

The likelihood ratio test is mainly used to reduce a number of parameters of a model. To reduce parameters of a model, a vector of parameters that defines the minimum number of parameters that can be in a model,  $\Omega_{min}$ , and another vector that defines the maximum number of parameters in the model are evaluated,  $\Omega_{max}$ .

The two vectors of parameters are used to parametrize two log likelihood functions. Then the log likelihood ratio test is used to choose between the two vectors of parameters considering the difference between the values of the two parameterized log likelihood functions. The difference between the values of the two functions is called deviance,  $\Lambda$ .

$$\Lambda = 2(l(\Omega_{max}) - l(\Omega_{min})) \quad (1.47)$$

where  $\Omega = \beta_{min}, \beta_{max}$  or  $\Omega = \theta_{min}, \theta_{max}$ . The deviance,  $\Lambda$ , is  $\chi_{df}^2$  distributed with the degree of freedom,  $df$ , being equivalent to the difference between the numbers of parameters in the two vectors. The log likelihood function,  $l$ , is the same as shown in equation 1.36. Then two hypotheses are set up to assess the validity of the two models.

$$H_0: \Omega = \Omega_{min} \text{ versus } H_1: \Omega \neq \Omega_{min}$$

If the deviance,  $\Lambda$ , is sufficiently small, there is evidence for the null hypothesis, which means that a model with a vector (of parameters) that has the minimum number of parameters,  $\Omega_{min}$ , is the best fit for the data; but when the deviance is sufficiently large, there is evidence against the null hypothesis, which indicates that the model with the vector that has the maximum number of parameters,  $\Omega_{max}$ , is the best fit for the data.

$$p = \frac{1}{2}(1 - \chi_{\Lambda,1}^2) - \frac{1}{2}(1 - \chi_{\Lambda,2}^2) \quad (1.48)$$

The p-value,  $p$ , is the probability with which to assess whether the null hypothesis should be accepted or rejected: if the deviance, which is the test statistic, is significant on  $\alpha$ , the null hypothesis is rejected if  $p < \alpha$ ; but it is accepted if  $p > \alpha$  [5].

## 1.11 The $t$ -test

The  $t$ -test is used to test fixed effects in the linear mixed effects model. It tests the two hypotheses

$$H_0: \beta = 0 \text{ versus } H_1: \beta \neq 0$$

The  $t$ -test statistic is defined by

$$t = \frac{\tilde{\beta}}{se(\tilde{\beta})} \quad (1.49)$$

The  $t$ -test follows a  $t$  distribution and the degree of freedom is evaluated by considering the grouping level at which a term, fixed effect, is estimated.

### 1.11.1 The Confidence Interval

In the linear mixed effects model implemented by J.C Pinheiro, the confidence intervals,  $CI$ , are based on an approximated  $t$  distribution rather than  $Z$  distribution as in the Wald confidence interval. The interval is created for each fixed effect in the model. Let's suppose that a vector of fixed effects,  $\beta$ , comprises  $\beta_j$  where  $j = 1, 2, 3 \dots, p$ . The interval for each fixed effect in the model is defined by

$$CI_j = \tilde{\beta}_j \pm t_{df, \frac{\alpha}{2}} \sqrt{Var(\tilde{\beta}_j)} \quad (1.50)$$

Where  $\tilde{\beta}_j$  is an estimate in the estimated vector of fixed-effects (1.42),  $\tilde{\beta}$ ,  $\alpha$  is the level which is usually assigned 0.05,  $Var(\tilde{\beta}_j)$  is a variance estimate for  $\tilde{\beta}_j$  in the matrix (1.44) or (1.46) and  $df$  is the degree of freedom.

## 1.12 The Information Criteria

The information criteria are used to choose the best fitted model for data. The two often used information criteria are Akaike Information Criterion and Bayesian Information Criterion.



The Akaike Information Criterion,  $AIC$ , is defined by

$$AIC = -2l(\tilde{\boldsymbol{\beta}}; \tilde{\boldsymbol{\theta}}) + 2p \quad (1.51)$$

where  $p$  is the number of parameters in a model.  $l(\tilde{\boldsymbol{\beta}}; \tilde{\boldsymbol{\theta}})$  is either the log likelihood function shown in the equation 1.38 or the restricted log likelihood function shown in the equation 1.45

The Bayesian Information Criterion is defined by

$$BIC = -2l(\tilde{\boldsymbol{\beta}}; \tilde{\boldsymbol{\theta}}) + 2p \ln(n) \quad (1.52)$$

where  $n = \sum_{i=0}^N n_i$  is the total number of data in a dataframe. In a case where AIC is used to choose a best fitted model to data, smaller AIC indicates a best model. In a model where BIC is used for the same purpose, smaller BIC indicates a best fitted model to the data. According to Pinheiro (2000), the  $BIC$  for restricted log likelihood can be calculated by using the equation 1.45 and substituting  $n$  with  $(n - p_\beta)$ . Where  $p_\beta$  is the number of fixed effects parameters [5] .

### 1.13 The assumptions of the LME Model

The linear mixed effects models, LME, are built on four keys assumptions that are structured on the normal distribution and covariance structure. The key assumptions are as listed in the following items:

- That the random effects have normal distributions;
- That the errors are normally distributed;
- That the errors are independent as responses are independent given random effects though not in general. The errors are sometimes correlated;
- That the errors are independent of the random effects, and
- That the errors are homoscedastic though not also in general. They are sometimes heteroscedastic.

The assumptions had led to many researches: it had been demonstrated that the maximum likelihood inference is robust to non-normal distributions of random effects (Butler and Louis, 1992; Verbeke and Lasaffre, 1997; Zhang and Davidian, 2001)[7].

As shown above, the maximum likelihood estimation (MLE) of a model parameters is usually obtained with the equation of the log likelihood function, shown below, under the Guassian assumption. In the case where optimization is required, the Newton-Raphson Method is often applied.

$$l(\tilde{\beta}) = -\frac{1}{2} \sum_{i=1}^N \{n_i \ln(2\pi) + \ln |\tilde{\mathbf{V}}_i| + (\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\beta})^T \tilde{\mathbf{V}}_i^{-1} (\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\beta})\}$$

$$\tilde{\beta}(\tilde{\theta}) = (\mathbf{X}^T \mathbf{V}(\tilde{\theta})^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V}(\tilde{\theta})^{-1} \tilde{\mathbf{Y}}$$

$$\tilde{\beta} = \tilde{\beta}^{i-1} + \frac{U(\tilde{\beta}^{i-1})}{U'(\tilde{\beta}^{i-1})}$$

Furthermore, Lange and Laird (1989) have showed that the variance of MLE of fixed effects is strongly dependent on a covariance structure, but the variance estimates of fixed effects in a model with random intercept and random slope are unbiased even when the true covariance structure implies more random effects than specified.

So far, the robustness of the fixed effects of the LME models to non-normal distribution of random effects and misspecified covariance structure had been well researched on, but robustness of the fixed effects, from the MLE inference, to non-normal distribution of errors does not seem to have gotten enough attention of researchers. Then these questions come up: *Is maximum likelihood inference from the linear mixed effects model robust to non-normal distribution of errors? If it is, how robust is it?* The answers to these questions are the motivations of this thesis [7].

# Chapter 2

## Checking an LME Fitted model

### 2.1 The Goodness of Fit of a Model

One of the traditional practices of fitting with the linear mixed effects model is checking goodness of fit of an LME fitted model. The checking of the fitted model consists of verification of whether the assumptions of the normal distributions of the random effects and the errors and homoscedasticity (constant variance of errors) are achieved in the model.

There is a number of test, which is often applied to check a model's goodness of fit such as the QQ plot of Cholesky residuals, the Shapiro-Wilk test, the Royston's  $V'$  Statistic, and  $\chi^2$  distribution. The tests are mainly used to check a deviation from the LME assumption of normal distribution.

### 2.2 Cholesky Residuals

In the linear mixed effects model, the Cholesky residuals are crucial to check a departure from the normal distribution. Sometimes, residuals are correlated and cannot be used to perform the test for a deviation from the normal distribution. In the case of residual correlation, the Cholesky residuals are calculated which are used to perform the test.

$$\tilde{\mathbf{R}}_i = \tilde{\mathbf{L}}^T (\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}}) \tag{2.1}$$

The uncorrelated residuals are calculated by using the upper triangle of Cholesky decomposed covariance matrix ( $\mathbf{V}_i$ ),  $\mathbf{L}^T$ . The computation is done as shown in the equation 2.1. Then, the QQ-plot of the Cholesky residuals are used as the good basis to check departure from a normal distribution. The quantiles of standard normal are usually plotted against the standardized residuals to assess the assumption (of normal distribution).

The figure 2.1 shows that a fitted model is good as there is a linear relationship between the quantiles of standard normal and the standardized residuals. The linearity between the quantiles of standard normal and the standardized residuals means that the assumption (of the normal distribution) is satisfied.

However, the figure 2.2 shows a model that does not satisfy the assumption of the (normal) distribution since there is a non-linearity between the quantiles of standard normal and the standardized residuals [7].

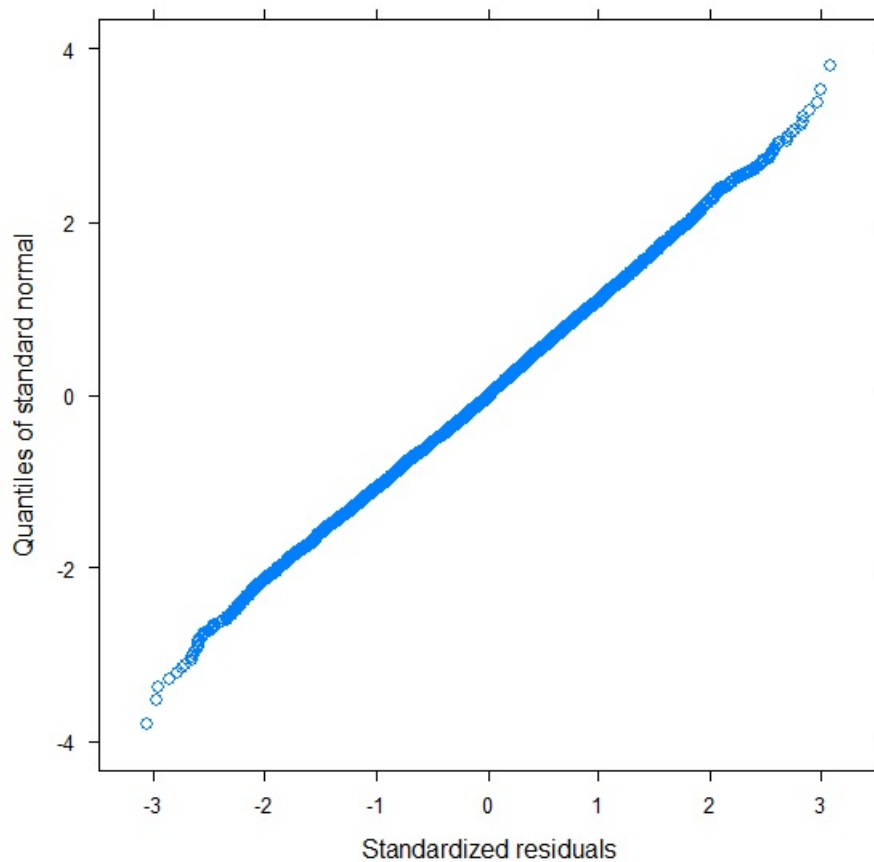


FIGURE 2.1: A QQ-Plot for Checking Departure from Normal a Distribution

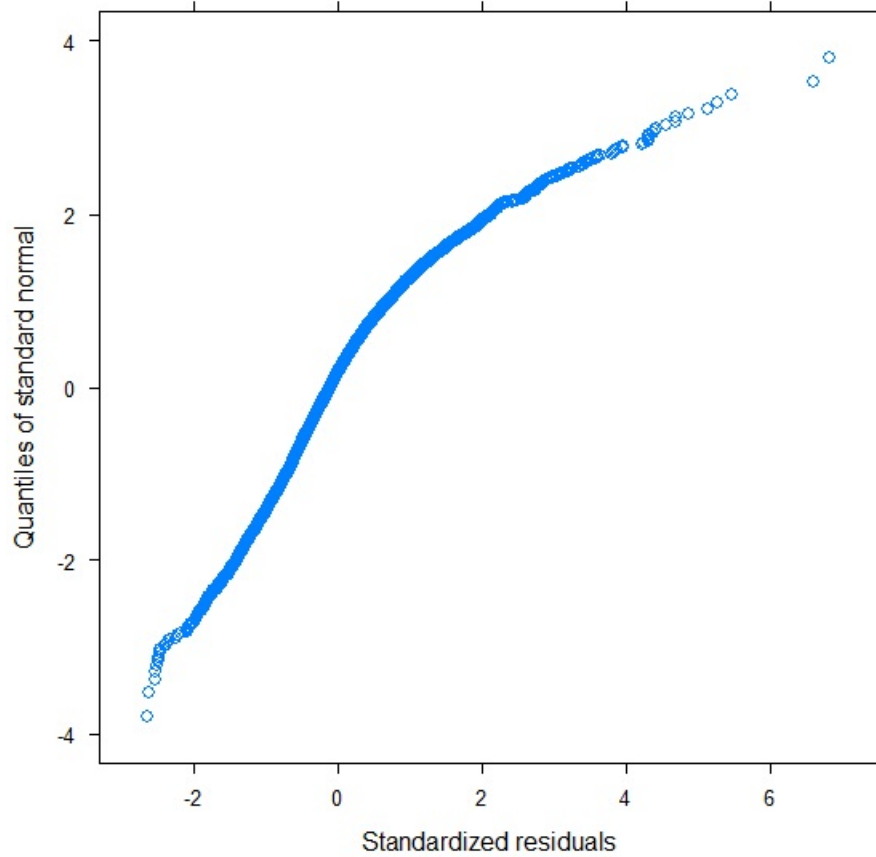


FIGURE 2.2: A QQ-Plot for Checking Departure from a Normal Distribution

### 2.3 $\chi^2$ Distribution for a Model Check

Sometimes, the need to check an influential subject or unit in a model may present itself. When such a need arises, it is accomplished using  $\chi^2$  distribution for the model check. In 2004, Park and Lee made such proposition to check departure from normal distribution using the equation 2.2. On the other hand, it is advisable to use the QQ-plot to check departure from a normal distribution because  $\chi^2$  is less sensitive to a deviation from normality assumption [7].

$$\tilde{\mathbf{R}}_i = (\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}})^T \tilde{\mathbf{V}}_i^{-1}(\tilde{\boldsymbol{\theta}})(\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}}) \quad (2.2)$$

$$\frac{\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}}}{\tilde{\mathbf{L}}} \sim N(0, 1) \text{ whereas } \left(\frac{\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}}}{\tilde{\mathbf{L}}}\right)^2 = (\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}})^T \tilde{\mathbf{V}}_i^{-1}(\tilde{\boldsymbol{\theta}})(\tilde{\mathbf{Y}}_i - \tilde{\mathbf{X}}_i \tilde{\boldsymbol{\beta}}) \sim \chi^2(n_i)$$

## 2.4 Shapiro-Wilk Test

The Shapiro-Wilk test is often used to test whether a sample is from a normal distribution. The test was published by Samuel Sanford Shapiro and Martin Wilk in 1965 . The test statistic is as shown in the equation 2.3.

$$p - value = \frac{(\sum_{i=1}^n c_i y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.3)$$

$y_i$  - the  $i$ th ordered statistic; that is the  $i$ th smallest number in a sample and  $\bar{y}$  - the average of the ordered statistic.

$$\tilde{\mathbf{C}} = \frac{\tilde{\mathbf{M}}^T \tilde{\mathbf{V}}^{-1}}{(\tilde{\mathbf{M}}^T \tilde{\mathbf{V}}^{-1} \tilde{\mathbf{V}}^{-1} \tilde{\mathbf{M}})^{\frac{1}{2}}} \quad (2.4)$$

The vector  $\tilde{\mathbf{C}} = [c_1, c_2, \dots, c_n]$  is calculated as shown in the equation 2.4, where  $\tilde{\mathbf{M}} = [m_1, m_2, \dots, m_n]$  - the expected values of the ordered statistic and  $\mathbf{V}^{-1}$  - the covariance matrix of the ordered statistic  $\tilde{\mathbf{Y}} = [y_1, y_2, \dots, y_n]$ .

The test is based on null hypothesis that a population is normally distributed: the null hypothesis is rejected if  $p - value$  is less than a chosen  $\alpha$  level, which means that the sample of the data is not from a normal distribution or not normally distributed. When the  $p - value$  is greater than the chosen  $\alpha$  level, it an evidence for the null hypothesis, which is that the sample is from a normal distribution or normally distributed [8].

For example, the often used  $\alpha$  level is 0.05. If  $p - value < 0.05$ , the sample of the data is not from a normal distribution. If  $p - value > 0.05$  , it an evidence that the sample is from a normal distribution.

# Chapter 3

## Research Procedure

### 3.1 The Simulation Setup

The simulation setup was planned in a way to study the robustness of the parameter estimates to the linear mixed effects model with independent, homoscedastic, but not normally distributed errors. The growth curve model shown in the equation 1.10 was used to generate datasets with conformity to the LME assumptions such as the normal distributions of random effects and errors, homoscedasticity of error variances, and independence of errors.

The datasets were generated from a random intercept and random slope, RIRS, model and random intercept, RI, model using the growth curve model (equation 1.10). The reason for the generated datasets from the two models that conformed to the LME assumptions was to fit the generated data and study the estimation property of the linear mixed effects model to such dataset, which would pinpoint difference in the (estimation) properties when the errors are not from a normal distribution.

Assuming correlated random effects, the values for the random intercept,  $\alpha_{i0}$ , and random slope,  $\alpha_{i1}$ , for the unit  $i$  in the sample were sampled from a multivariate normal distribution using Cholesky upper triangle matrix of the random-effects covariance matrix. The sampling model is shown in the equation 3.1.

$$\begin{bmatrix} \alpha_{i0} \\ \alpha_{i1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + chol\left( \begin{bmatrix} 3^2 & -0.225 \\ -0.225 & 0.1^2 \end{bmatrix} \right)^T \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \quad (3.1)$$

Where  $Z_1 \sim N(0, 1)$  and  $Z_2 \sim N(0, 1)$ . The normally distributed errors,  $e_{ij} \sim N(0, \sigma_e^2 = 25)$  and the sampled values for the random intercept and random slope for unit  $i$  were used in the growth curve model to generate datasets for the case where all the LME model assumptions were conformed to.

The true values that were assigned to the fixed effects of the growth curve model were obtained from the MADRS data. The Montgomery–Åsberg Depression Rating Scale, abbreviated as (MADRS), is a ten-item diagnostic questionnaire which psychiatrists use to measure the severity of depressive episodes in patients with mood disorders. It was designed in 1979 by British and Swedish researchers as an adjunct to the Hamilton Rating Scale for Depression (HAMD), which would be more sensitive to the changes brought on by antidepressants and other forms of treatment than the Hamilton Scale was [9]. Hence, the fixed effects were assigned as:  $\beta_0 = 27$ ,  $\beta_1 = -0.2$ ,  $\beta_2 = -2$ , and  $\beta_3 = -0.15$  using MADRS data.

$$Y_{ij} = \beta_0 + \alpha_{0i} + \beta_1 t_{ij} + \alpha_{i1} t_{ij} + \beta_2 X_i + \beta_3 X_i t_{ij} + e_{ij}. \quad (3.2)$$

The representations of the fixed effects can be inferred from the growth curve model shown in the equation 3.2. The  $\beta_3$  is the fixed effect for interaction between time and the discrete covariate,  $X_i$ , in the model, which can be estimated when data are fitted with interaction between the two covariates.

The discrete covariate,  $X_i$  that takes zero or one, is the covariate for a treatment received by a unit. let's suppose that the treatment is  $t$ . Then  $X_i$  is one for individuals (units) that received the  $t$  treatment, but it is zero for the individuals that did not receive the treatment.

$$X_i = \begin{cases} 1 & \forall t \\ 0 & \forall \textit{ otherwise} \end{cases}$$

The times of measurements,  $t_{ij}$ , were spaced at the interval of ten. That is the time vector,  $\mathbf{t}_i = [0, 10, 20, 30, 40]$  for sparsely balanced data; where  $j = 1, 2, 3, \dots, n_i = 5$ . For densely balanced data,  $\mathbf{t}_i = [0, 10, 20, 30, 40, 50, 60, 70]$  where  $j = 1, 2, 3, \dots, n_i = 8$ .

In the case of unbalanced data, the interval of measurement is the same as in the balanced measurement. The only difference is that the number of measurements,



$n_i$ , is dependent on a unit  $i$  where  $n_i = 3, 4, 5, 6$  or  $8$ . That is the length of the time vector,  $\mathbf{t}_i$ , can be three, four, five, six or eight in this case depending on a unit.

## 3.2 Sampling Process

The sample size,  $N$ , which is the number of units or individuals can be chosen from 6 to 1000. The reason for the design is to create many options for choosing sample size which is important to run a simulation on any setting and make an observation. Mainly, the sample sizes, 6, 10, 16, 20, 40, 60, 100 and 200, were used in the simulation. The intra-subject structure of generated datasets is made to assume balanced or unbalanced form.

### 3.2.1 Balanced Data

The balanced data are either densely balanced or sparsely balanced: the intra-subject structure of the sparsely balanced data has 5 calculated data at five different time points and the structure of the densely balanced data has 8 data calculated at 8 different time points for each individual in a sample to create a balanced longitudinal-data form; that is the same with saying that  $n_1 = n_2, \dots, = n_N$ , where  $i = 1, 2, \dots, N$ . The examples of the generated sparsely balanced data and densely balanced data are shown in the table 3.1 and the table 3.2 respectively.

TABLE 3.1: Sparsely balanced data,  $n_i = 5$

	Unit	$X_i$	$t_{ij}$	$Y_{ij}$
1	1	0	0	29.490115
2	1	0	10	25.001415
3	1	0	20	19.365212
4	1	0	30	14.967274
5	1	0	40	11.056047
6	2	1	0	25.617040
7	2	1	10	27.536993
8	2	1	20	29.151730
9	2	1	30	31.393381
10	2	1	40	33.669943

TABLE 3.2: Densely balanced data,  $n_i = 8$ 

	Unit	$X_i$	$t_{ij}$	$Y_{ij}$
1	1	0	0	29.795796
2	1	0	10	26.463944
3	1	0	20	25.716627
4	1	0	30	21.327139
5	1	0	40	18.638561
6	1	0	50	15.919713
7	1	0	60	13.419375
8	1	0	70	11.824865
9	2	1	0	29.060260
10	2	1	10	26.887011
11	2	1	20	24.660482
12	2	1	30	22.797123
13	2	1	40	20.557486
14	2	1	50	18.147173
15	2	1	60	16.132496
16	2	1	70	13.997739

### 3.2.2 Unbalanced Data

Unlike the balanced data, the unbalanced data have intra-subject structure in which individuals can have different number of data calculated at different number of time points.

In the case of unbalanced data, there is a random intra-subject structure. The random intra-subject structure is modeled with the equation 3.3.

$$n_r = s_{scale}(n_i - 1) \quad (3.3)$$

where  $n_i$  was randomly sampled from this set of number:  $\{3,4,5,6,8\}$ ,  $s_{scale}$  - the longitudinal interval between two data,  $n_r$  - the randomly sampled number of data calculated for an individual in a sample. An example of generated unbalanced data is shown in the table 3.3.

TABLE 3.3: Unbalanced data

	Unit	$X_{ij}$	$t_{ij}$	$Y_{ij}$
1	1	0	0	37.537016
2	1	0	10	18.138316
3	1	0	20	27.103178
4	1	0	30	23.413606
5	1	0	40	33.874424
6	1	0	50	31.232964
10	2	0	0	35.082150
11	2	0	10	21.260379
12	2	0	20	9.909056
13	2	0	30	6.443633
19	3	1	0	31.987610
20	3	1	10	14.960095
21	3	1	20	17.879594
22	3	1	30	10.838117
28	4	1	0	20.765595
29	4	1	10	18.750420
30	4	1	20	9.934463
31	4	1	30	4.668222
32	4	1	40	8.720113

### 3.3 Models for Data Generation

The data used in running the simulation were generated using the model shown in equation 1.10. The datasets for the random intercept and random slope model, abbreviated as RIRS, were generated by sampling from two different normal distributions for the random effects in the model -  $\alpha_{i0} \sim N(0, \sigma_0^2)$  and  $\alpha_{i1} \sim N(0, \sigma_1^2)$ .

The errors were either sampled from a normal distribution or skewed distribution depending on a simulation setting. The fixed effects used in the model were the ones specified on the MADRS data description. Also, the  $R$  facilities for  $U[0, 1]$ ,  $N(\mu, \sigma^2)$ , and other distributions such as exponential distribution and gamma distribution were used for the data generation.

On the other hand, the data for the random intercept, RI, model were generated by omitting the random slope and sampling from normal distribution for the random intercept in the model and hence, the effect of the random slope was not considered in the RI model. Apart from the omission of the random slope, the fixed effects and sampling processes in the RI model were implemented exactly as they were

done in the RIRS model. The data generated are in long format as shown in the table [3.1 3.2, 3.3].

### 3.4 RIRS Models

#### Step One

In the first step, the two functions called *BalancedData* and *UnbalancedData* were implemented to generate data from RIRS model where all the LME assumptions were exactly implemented as stated in the assumptions; they are:  $\alpha_{i0} \sim N(0, \sigma_0^2)$ ,  $\alpha_{i1} \sim N(0, \sigma_1^2)$ ,  $e_{ij} \sim N(0, \sigma_e^2)$  and  $\Sigma = \sigma_e^2 I$  which implies independency and homoscedasticity of errors. The data generated in the step one were used as validation bases.

The only difference between the two functions is that the *BalancedData* is used to generate balanced longitudinal data, but the function, *UnbalancedData*, is used to generate unbalanced data. Data generated from the two functions are important to study and verify whether there is any impact on the LME model due to the balancedness or unbalancedness of data.

#### Step Two

In the step two, data were generated using the same model (1.10). However, a corrupted normal distribution was used as the error distribution in this case. While generating each datum, the error used in the calculation was sampled from the corrupted normal distribution which is a combination of two normal distributions.

The functions that were implemented and used to generate the data in this step were called *SymBalancedData* and *SymUnbalancedData*. The prefix of the function, *Sym*, was derived from the fact that the error distribution is symmetric. The distribution is as shown in the following.

$$f(e_{ij}) = 0.5 \frac{1}{0.6\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(e_{ij} - 1.9)^2}{0.6^2}\right) + 0.5 \frac{1}{0.6\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(e_{ij} + 1.9)^2}{0.6^2}\right) \quad (3.4)$$

Let's suppose that the first normal distribution is  $N_1$  and the second normal distribution is  $N_2$  in the equation 3.4. Hence the equation is reduced to:

$$f(e_{ij}) = 0.5N_1(-1.9, \sigma^2 = 0.6^2) + 0.5N_1(1.9, \sigma^2 = 0.6^2),$$

$$E(e_{ij}) = 0.5E[N_1] + 0.5E[N_2] = 0.5 \times -1.9 + 0.5 \times 1.9 = 0,$$

$$\text{Var}(e_{ij}) = 0.5^2\text{Var}[N_1] + 0.5^2\text{Var}[N_2] = 0.5^2 \times 0.6^2 + 0.5^2 \times 0.6^2 = 0.18.$$

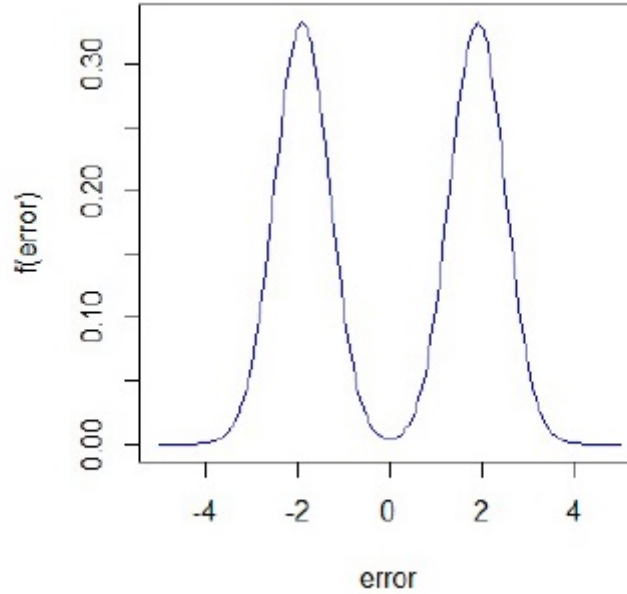


FIGURE 3.1: The Symmetric Corrupted Normal Distribution of Error

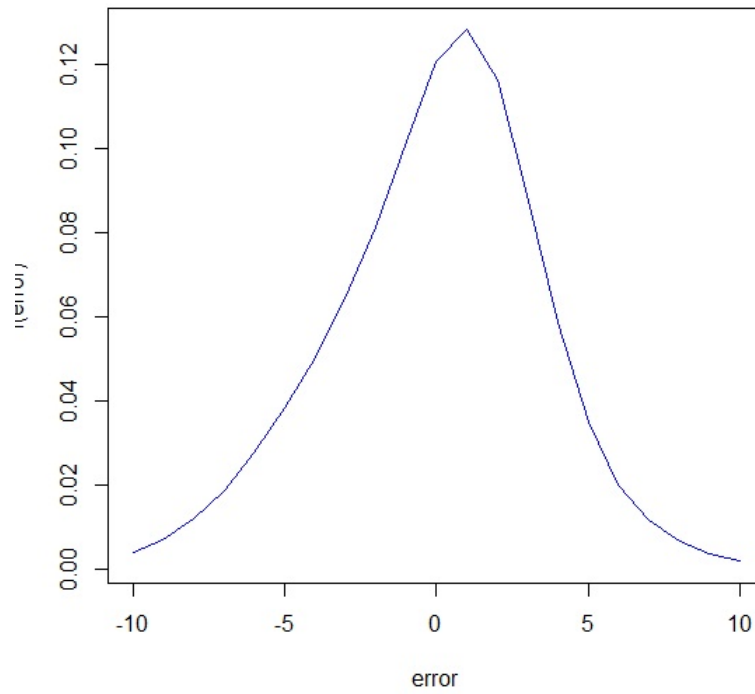
### Step Three

$$f(e_{ij}) = 0.3 \frac{1}{1.9\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(e_{ij} - 1.3)^2}{1.9^2}\right) + 0.7 \frac{1}{3.1\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(e_{ij} + 0.6)^2}{3.1^2}\right) \quad (3.5)$$

In the step three, the errors were sampled from a slightly asymmetric distribution shown in the equation 3.5. The sampled errors were also used to produce the data which their errors are from the (asymmetric) distribution. The distribution has expectation that is approximately zero.

$$f(e_{ij}) = 0.3N_1(1.3, \sigma^2 = 1.9^2) + 0.7N_1(-0.6, \sigma^2 = 3.1^2),$$

$$E(e_{ij}) = 0.3E[N_1] + 0.7E[N_2] = 0.3 \times 1.3 + 0.7 \times -0.6 \approx 0,$$




---

FIGURE 3.2: The Asymmetric Corrupted Normal Distribution of Error

$$\text{Var}(e_{ij}) = 0.3^2 \text{Var}[N_1] + 0.7^2 \text{Var}[N_2] = 0.3^2 \times 1.9^2 + 0.7^2 \times 3.1^2 = 5.$$

The functions that were implemented and used to generate data in the step three were called *AsymBalancedData* and *AsymUnbalancedData*.

#### Step Four

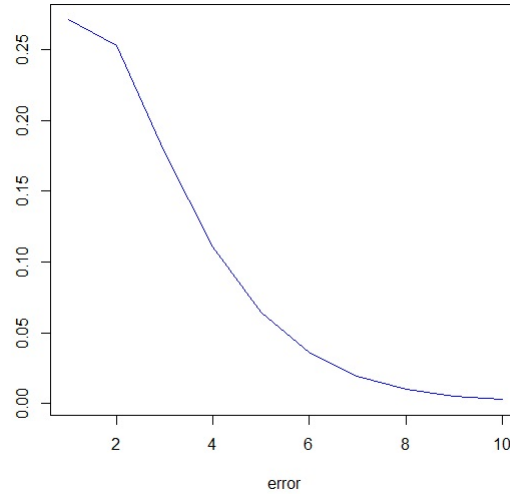
The gamma probability distribution function is

$$f(x) = \frac{b^a x^{a-1} \exp(-bx)}{\Gamma(a)} \quad (3.6)$$

The expectation of the probability distribution function (3.6),  $E[X] = \frac{a}{b}$  and the variance,  $\text{Var}(X) = \frac{a}{b^2}$ .

$$e_{ij} = X - \frac{a}{b} \quad (3.7)$$

Where  $X$  was sampled from the gamma distribution with the shape parameter,  $a = 2$ , and rate parameter,  $b = \sqrt{\frac{1.9}{2}}$ . Then the  $E[e_{ij}] = 0$  and  $\text{Var}[e_{ij}] = 2\frac{2}{19}$ .




---

FIGURE 3.3: The Skewed Distribution: Gamma

The functions that were implemented and used to generate the skewed data with the error samples from the distribution shown in the equation 3.7 were: *GamBalancedData* and *GamUnbalancedData*.

### Step Five

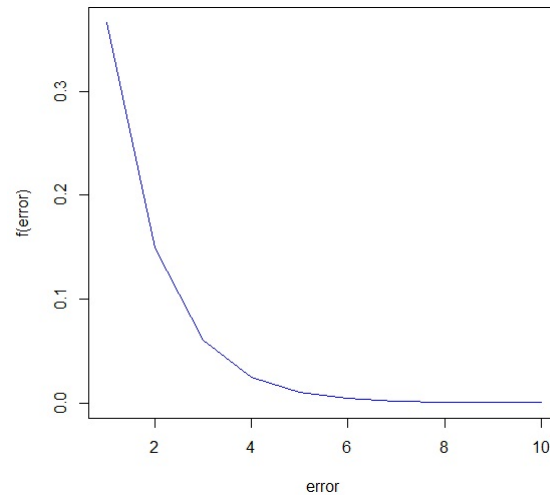
Highly skewed data were synthesized by using the errors sampled from an exponential distribution with mean that is equal to zero. The rate parameter,  $\lambda = 0.9$ .

$$f(x) = \lambda \exp(-\lambda x) H(x) \quad (3.8)$$

$$e_{ij} = X - \frac{1}{\lambda} \quad (3.9)$$

Where  $X$  was sampled from the exponential distribution (3.8). Hence  $E[e_{ij}] = 0$ . and  $\text{Var}[e_{ij}] = \frac{1}{\lambda^2}$ .

The functions that were implemented and used to produce the highly skewed data were: *ExpoBalancedData* and *ExpoUnbalancedData*.



---

FIGURE 3.4: The Very Skewed Distribution: Exponential

### 3.5 RI Models

The data of the random intercept, RI, model were generated in the same way other datasets in the random intercept and random slope, RIRS, model were generated. The only difference between the data of the two models is that the impact of random slope is omitted in the RI model, but random slope is present in the RIRS model and influenced its data generation.

The functions that were used in the RIRS models were used in the RI models with random slope deleted from the model of the equation [1.10](#).



# Chapter 4

## Simulations and Results

In this chapter, simulations to study the robustness of the LME model to non-normal distributions of errors were carried out and the results presented. There are five cases for which the simulations were performed. In the **case 1**, the errors and random effects are normally distributed. However, only the probability distribution of error is deformed in contrast to the LME assumptions in the **case 2** through to the case **case 5**.

The simulations were carried out methodically; that is that the 'goodness-of-fit' analyses were firstly checked to assess the distributions of the errors and random effects. The observed responses were plotted against the fitted values of the units to check for a mismatch or an outlying observation before assessing the robustness of the LME model to a deformed error distribution.

### 4.1 Case 1: Model Validation

The case 1 was used as the validation case (because all the assumptions for the random effects and errors in the typical LME models were strictly considered in this case as it would help to pinpoint a change in the parameter estimates due to a deformation of error distribution). The error distribution is normal with mean,  $\mu = 0$  and the variance,  $\sigma_e^2 = 25$ . The random effects are correlated and they are generated from the multivariate normal distribution as shown in the equation 3.1.

### 4.1.1 Simulation Settings for Case 1

In reference to the equation 1.10,  $Y_{ij} = \beta_0 + \alpha_{0i} + \beta_1 t_{ij} + \alpha_{i1} t_{ij} + \beta_2 X_i + \beta_3 X_i t_{ij} + e_{ij}$ , the errors,  $e_{ij} \sim N(0, \sigma_e^2 = 25)$  and the random-effects vector,  $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{0i} \\ \alpha_{1i} \end{bmatrix} \sim MVN(\mathbf{0}, \boldsymbol{\Sigma})$ . Where

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3^2 & -0.225 \\ -0.225 & 0.1^2 \end{bmatrix}$$

In this case 1, the data generated from the model (1.10), in which all the LME assumptions were considered, were fitted using the LME method implemented by Jose C. Pinheiro. The 'goodness-of-fit' analysis and the validation of the LME estimation property to the error distribution were assessed as shown in the following.

#### Checking the Goodness of Fit for Case 1

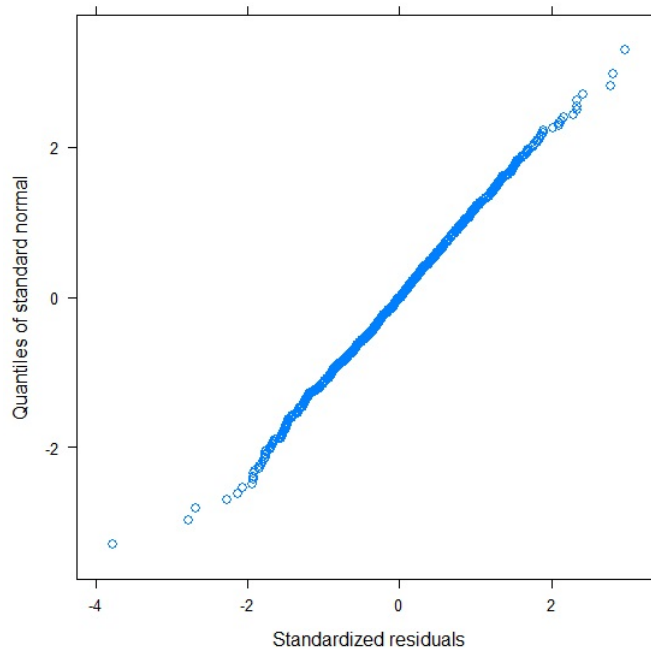


FIGURE 4.1: Validation Model: qqnorm Plot for Residuals

The figure 4.1 shows that residuals were samples drawn from a normal distribution, so errors,  $e_{ij}$ , are normally distributed. The samples of the residuals were also tested with the Shapiro-Wilk test and the test statistic shows that the samples

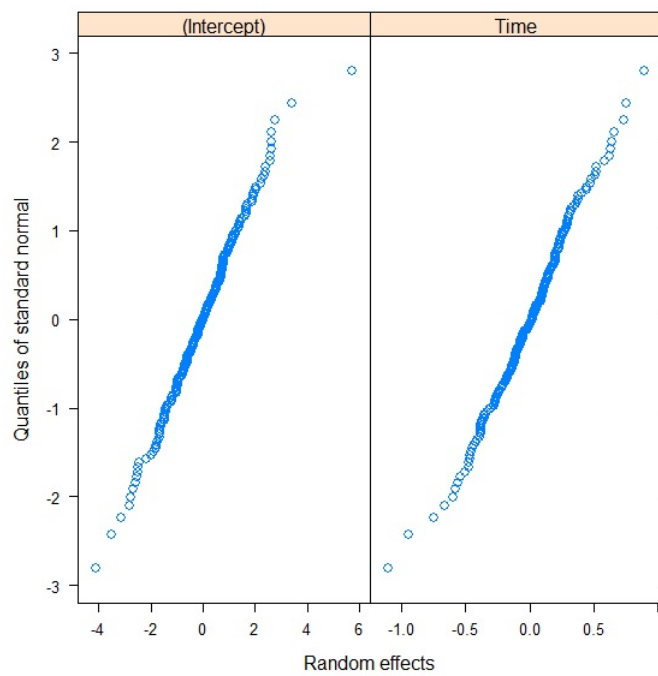


FIGURE 4.2: Validation Model: qqnorm Plot for Random Effects

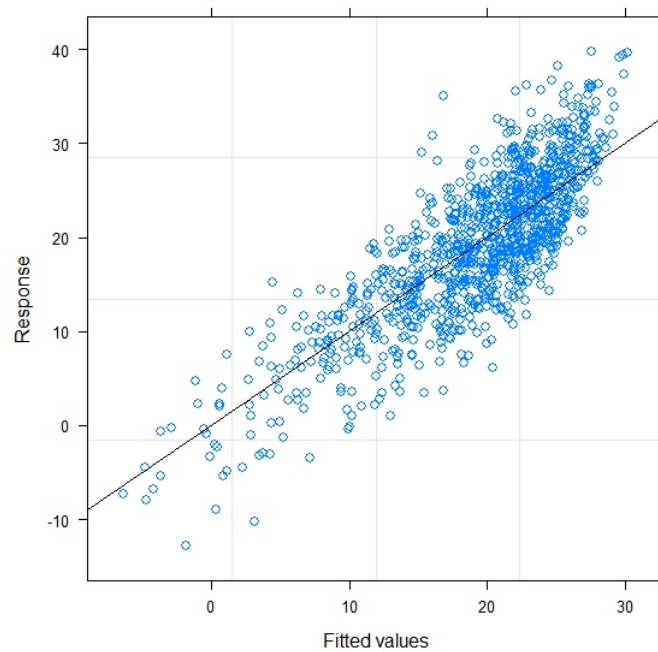


FIGURE 4.3: Validation Model: Responses against Fitted Values

were also from a normal distribution as the value of the statistic is,  $W = 0.9988$  with the  $p$ -value = 0.7419. The  $p$ -value implies that null hypothesis that the

samples were drawn from a normal distribution is not rejected.

The figure 4.2 shows that the samples of the random effects for intercept and slope were also drawn from normal distributions. The figure 4.3, the plot of the responses against fitted values, shows the responses match with the fitted values.

### Case 1: the LME Validation

The validation of the LME model was simulated and the obtained results are presented in the following tables. The LME validation as well as the robustness was measured with coverage probability, the proportion of times a confidence interval contains the parameter values. Such interval is mostly based on 95% (0.95) confidence interval.

The simulation iteration was set to 1000. The number of units,  $N$ , used in the simulation were 6,10, 16, 20, 40, 60, 100, and 200. The number of data points in each unit,  $n_i$ , used in the simulation were 5 and 8. In the case of unbalanced data,  $n_i = 3, 4, \dots$ , or 8.

The simulations were run for balanced random intercept and random slope model, abbreviated as  $RIRS_b$ , unbalanced random intercept and random slope model,  $RIRS_u$ , balanced random intercept model,  $RI_b$ , and unbalanced random intercept model,  $RI_u$ .

The tables are arranged in two columns: the first column contains tables for simulated models where  $n_i = 5$  for sparsely balanced data . The second column contains tables for simulated models where  $n_i = 8$  for densely balanced data. However, the unbalanced data where the number of measurements,  $n_i = 3, 4, 5, \dots$  or 8, is random and dependent on a unit  $i$  is included both in the first column and the second column.

	$N = 6$				$N = 6$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.972	0.952	0.950	0.940	0.965	0.951	0.937	0.946
$\beta_1$	0.964	0.974	0.954	0.940	0.954	0.951	0.957	0.956
$\beta_2$	0.986	1.000	0.990	0.985	0.984	0.981	0.983	0.984
$\beta_3$	0.954	0.960	0.947	0.950	0.933	0.961	0.947	0.948

$N = 10$					$N = 10$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.965	0.963	0.944	0.939	0.948	0.948	0.930	0.942
$\beta_1$	0.967	0.971	0.947	0.950	0.947	0.956	0.942	0.949
$\beta_2$	0.975	0.976	0.971	0.974	0.975	0.981	0.960	0.967
$\beta_3$	0.957	0.957	0.951	0.948	0.954	0.954	0.958	0.952
$N = 16$					$N = 16$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.948	0.943	0.942	0.943	0.944	0.954	0.947	0.949
$\beta_1$	0.963	0.945	0.965	0.950	0.939	0.951	0.955	0.934
$\beta_2$	0.969	0.970	0.951	0.955	0.961	0.960	0.967	0.962
$\beta_3$	0.957	0.946	0.943	0.962	0.946	0.949	0.945	0.943
$N = 20$					$N = 20$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.956	0.958	0.960	0.934	0.945	0.949	0.935	0.939
$\beta_1$	0.944	0.944	0.953	0.950	0.944	0.934	0.951	0.950
$\beta_2$	0.966	0.960	0.975	0.951	0.962	0.975	0.955	0.947
$\beta_3$	0.956	<b>0.926</b>	0.952	0.948	0.931	0.944	0.943	0.940
$N = 40$					$N = 40$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.944	0.931	0.942	0.952	0.937	0.953	0.943	0.947
$\beta_1$	0.960	0.949	0.940	0.950	0.947	0.940	0.952	0.953
$\beta_2$	0.963	0.953	0.959	0.952	0.937	0.940	0.963	0.950
$\beta_3$	0.960	<b>0.925</b>	0.947	0.943	0.949	0.936	0.957	0.944
$N = 60$					$N = 60$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.933	0.946	0.948	0.955	0.935	0.959	0.960	0.941
$\beta_1$	0.948	<b>0.931</b>	0.958	0.944	0.942	0.942	0.963	0.948
$\beta_2$	0.965	0.949	0.964	0.954	0.945	0.955	0.952	0.962
$\beta_3$	0.936	<b>0.926</b>	0.954	0.958	0.944	0.947	0.950	0.946

$N = 100$					$N = 100$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.950	0.939	0.940	0.944	0.933	0.947	0.949	0.956
$\beta_1$	0.949	0.944	0.934	0.950	0.937	0.942	0.946	0.944
$\beta_2$	0.954	0.954	0.947	0.955	0.946	<b>0.935</b>	0.964	0.948
$\beta_3$	0.953	0.952	0.942	0.957	0.947	0.945	0.951	0.950

$N = 200$					$N = 200$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.955	0.942	0.946	0.947	0.942	0.946	0.946	0.957
$\beta_1$	0.959	0.950	0.958	0.951	0.943	0.942	0.943	0.948
$\beta_2$	0.941	0.946	0.957	0.948	0.943	0.949	0.941	0.959
$\beta_3$	0.951	0.956	0.957	0.944	0.942	0.940	0.949	0.952

## 4.2 Case 2: The Corrupted Distribution of Error

### 4.2.1 Simulation Settings for Case 2

As the simulation settings of case 1, the same settings were used in the case 2 except that the corrupted normal distribution, a combination of two normal distributions, was used in this case as the distribution of error. The expectation of the corrupted distribution is zero. The distribution is shown in the equation 3.4; that is:

$$f(e_{ij}) = 0.5N(1.9, 0.6) + 0.5N(-1.9, 0.6). \text{ Where } E[e_{ij}] = 0 \text{ and } \text{Var}[e_{ij}] = 0.18.$$

#### Checking the Goodness of Fit for Case 2

The figure 4.4 shows that residuals were drawn from a normal distribution as the standardized residuals linearly match with the quantiles of standard normal. The Shapiro-Wilk test was tested on the residuals and the test statistic also showed that the residuals were from the normal distribution as the value is  $W = 0.9979$  with  $p\text{-value} = 0.2224$  for acceptance against 0.05 for rejection of the hypothesis.

Also, the figure 4.5 shows that the random effects for intercept and slope of the model were also drawn from normal distributions. Hence, they are normally distributed.

The figure 4.6, the plot of responses against fitted values, depicts that the responses are matches of the fitted values as outliers are not present in the plot.

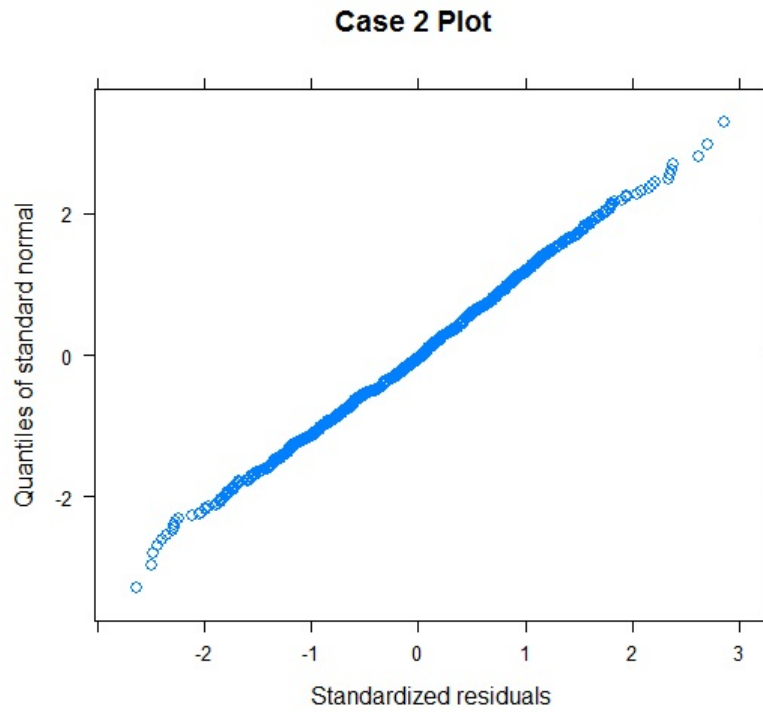


FIGURE 4.4: Corrupted Distribution: qqnorm Plot for Residuals

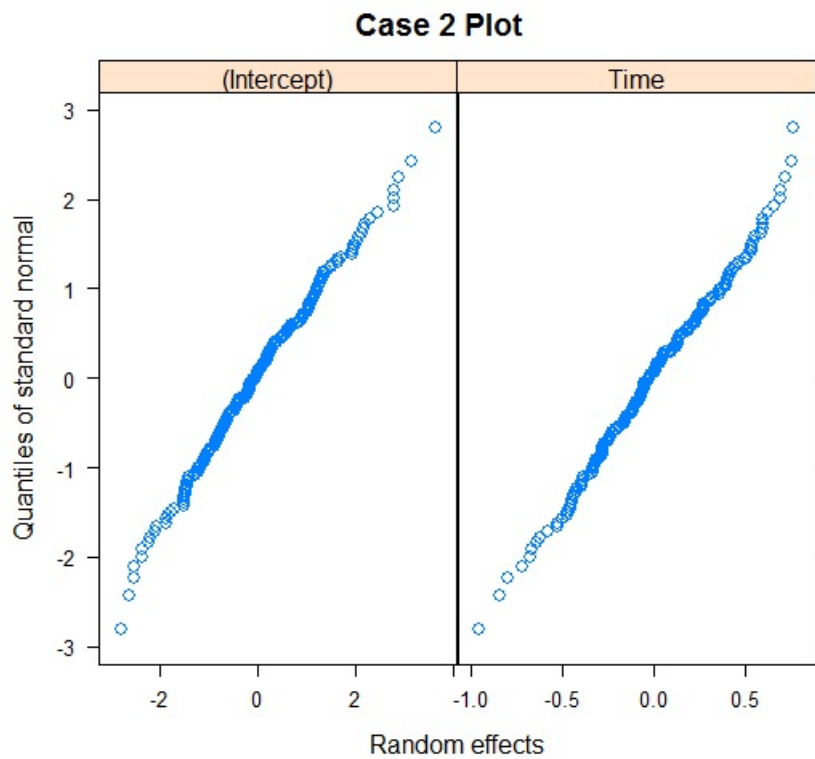


FIGURE 4.5: Corrupted Distribution: qqnorm Plot for Random Effects

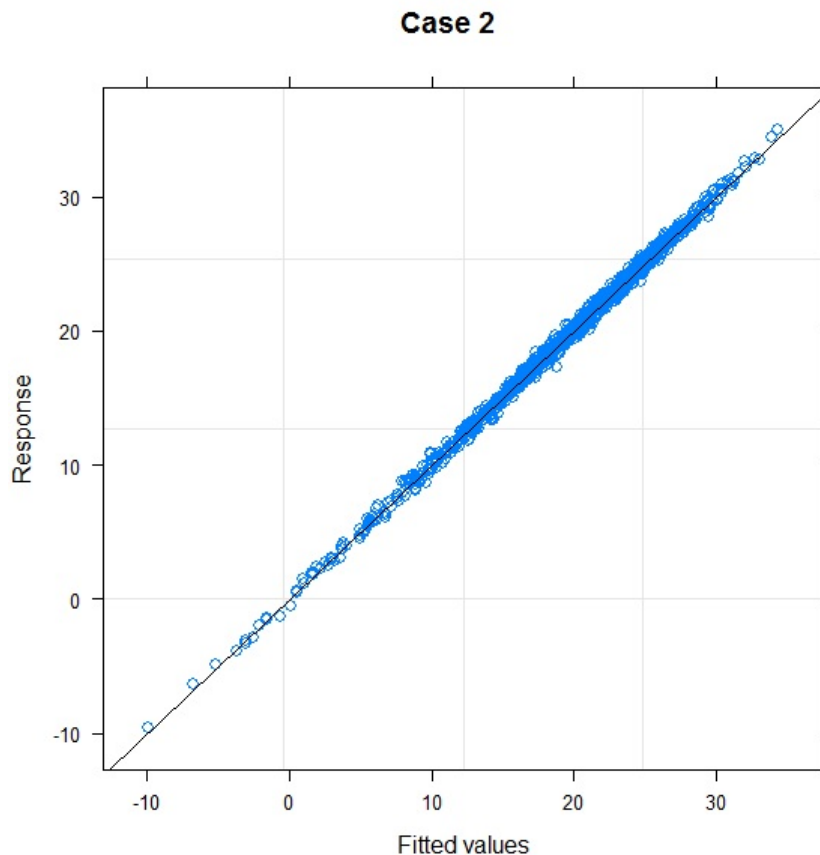


FIGURE 4.6: Corrupted Distribution: Responses against Fitted Values

**The LME Robustness for Case 2**

$N = 6$					$N = 6$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.900</b>	<b>0.886</b>	<b>0.885</b>	<b>0.901</b>	<b>0.892</b>	<b>0.894</b>	<b>0.893</b>	<b>0.902</b>
$\beta_1$	<b>0.900</b>	<b>0.898</b>	0.950	0.958	<b>0.895</b>	<b>0.904</b>	0.956	0.943
$\beta_2$	0.944	0.946	0.955	0.955	0.952	0.954	0.955	0.943
$\beta_3$	<b>0.895</b>	<b>0.887</b>	0.945	0.948	<b>0.884</b>	<b>0.910</b>	0.948	0.949

$N = 10$					$N = 10$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.905</b>	<b>0.923</b>	<b>0.920</b>	0.934	<b>0.919</b>	<b>0.917</b>	<b>0.909</b>	<b>0.920</b>
$\beta_1$	<b>0.911</b>	<b>0.922</b>	0.948	0.946	<b>0.917</b>	<b>0.931</b>	0.956	0.950
$\beta_2$	0.939	0.950	0.952	0.952	0.953	0.945	0.937	0.945
$\beta_3$	<b>0.920</b>	<b>0.920</b>	0.951	0.950	<b>0.920</b>	<b>0.916</b>	0.953	0.953



$N = 16$					$N = 16$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.926</b>	0.930	0.939	0.949	0.942	0.937	0.938	0.942
$\beta_1$	<b>0.930</b>	<b>0.921</b>	0.950	0.949	0.940	<b>0.931</b>	0.963	0.956
$\beta_2$	<b>0.933</b>	0.951	0.968	0.950	0.947	0.943	0.946	0.943
$\beta_3$	0.932	<b>0.924</b>	0.951	0.959	0.934	0.931	0.939	0.951

$N = 20$					$N = 20$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.938	0.938	<b>0.926</b>	0.945	<b>0.928</b>	0.946	0.932	0.943
$\beta_1$	0.942	0.940	0.945	0.958	0.941	0.940	0.939	0.949
$\beta_2$	0.948	0.954	0.953	0.955	0.946	0.956	0.949	0.952
$\beta_3$	0.937	0.942	0.946	0.949	<b>0.926</b>	0.932	0.946	0.955

$N = 40$					$N = 40$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.946	0.940	0.949	0.948	0.935	0.953	0.954	0.943
$\beta_1$	0.938	0.944	0.954	0.952	0.938	0.948	0.953	0.951
$\beta_2$	0.945	0.954	0.948	0.950	0.949	0.946	0.958	0.959
$\beta_3$	0.949	0.952	0.941	0.944	<b>0.927</b>	0.947	0.954	0.946

$N = 60$					$N = 60$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.953	0.949	0.936	0.945	0.944	0.951	0.956	0.940
$\beta_1$	0.945	0.942	0.971	0.943	0.939	0.953	0.953	0.941
$\beta_2$	0.945	0.959	0.936	0.945	0.959	0.955	0.947	<b>0.935</b>
$\beta_3$	0.937	0.954	0.955	0.935	0.956	0.950	0.953	0.950

$N = 100$					$N = 100$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.942	0.949	0.940	0.963	0.941	0.951	0.946	0.945
$\beta_1$	0.938	0.936	0.948	0.949	0.945	0.940	0.947	0.949
$\beta_2$	0.945	0.949	0.940	0.960	0.956	0.937	0.949	0.950
$\beta_3$	0.937	0.947	0.942	0.955	0.946	0.934	0.962	0.960

$n_i$	$N = 200$				$N = 200$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.945	0.939	0.941	0.956	0.950	0.941	0.941	0.954
$\beta_1$	0.948	0.961	0.946	0.944	0.934	0.955	0.947	0.961
$\beta_2$	0.942	0.938	0.942	0.955	0.945	0.948	0.954	0.960
$\beta_3$	0.944	0.951	0.950	0.952	0.939	0.944	0.953	0.959

### 4.3 Case 3: Asymmetric Distribution of Error

#### 4.3.1 Simulation Settings for Case 3

In the case 3, asymmetric distribution was used as the probability density function of the error. The distribution is the combination of two normal probability density functions. The difference between the corrupted distribution and the asymmetric distribution is that it is more skewed than the corrupted distribution. The distribution is detailed in the equation 3.5. That is  $f(e_{ij}) = 0.3N(1.3, 1.9) + 0.7N(-0.6, 3.1)$ . The  $E[e_{ij}] \approx 0$  and  $\text{Var}[e_{ij}] = 5$ .

The simulation settings were the same with case 1 except that the probability distribution of error used in this case was asymmetric.

#### Checking the Goodness of Fit for Case 3

The figure 4.7 depicts that the samples for the errors were sampled from a normal distribution as in the previous cases. The Shapiro-Wilk test was conducted with the samples and it indicates that the errors were from the distribution as the values of the test statistic was  $W = 0.9991$  with  $p\text{-value} = 0.8917$  for acceptance against 0.05 for rejection of the null hypothesis. The figure 4.8 shows also that the random effects were drawn from a normal distribution. The figure 4.9 shows that the responses match with the fitted values.

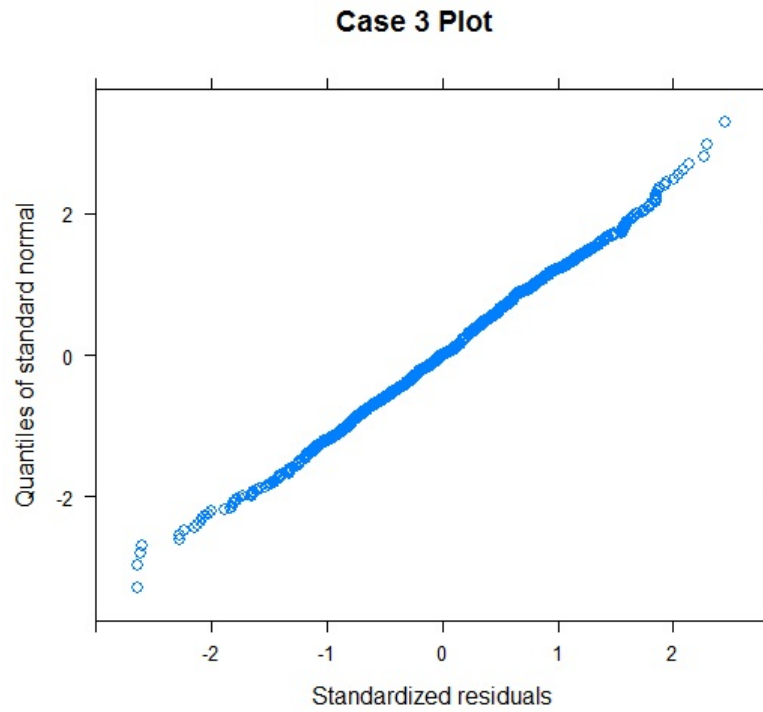


FIGURE 4.7: Asymmetric Distribution: qqnorm Plot for Residuals

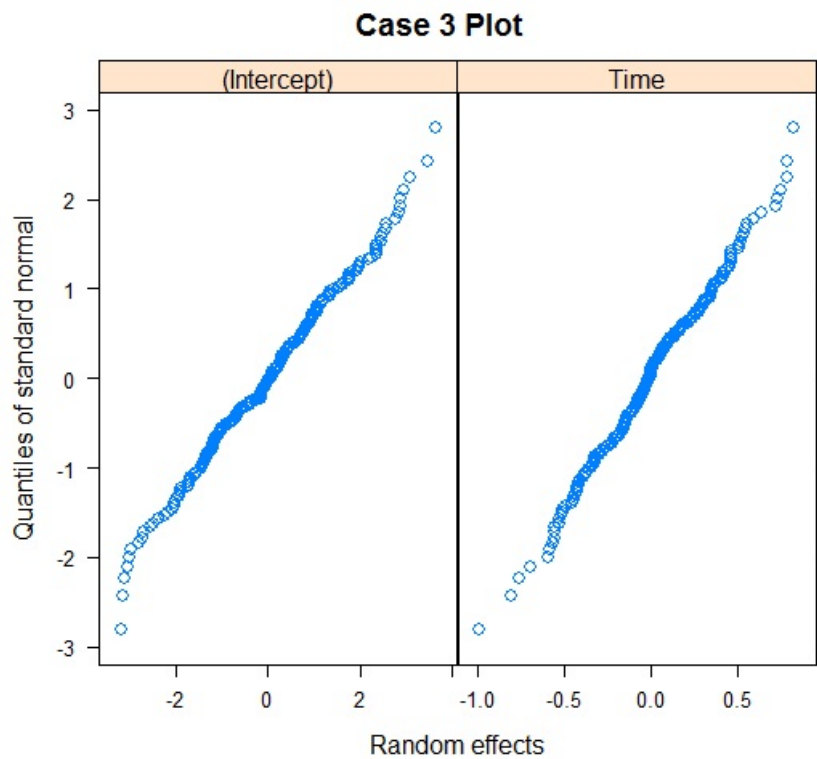


FIGURE 4.8: Asymmetric Distribution: qqnorm Plot for Random Effects

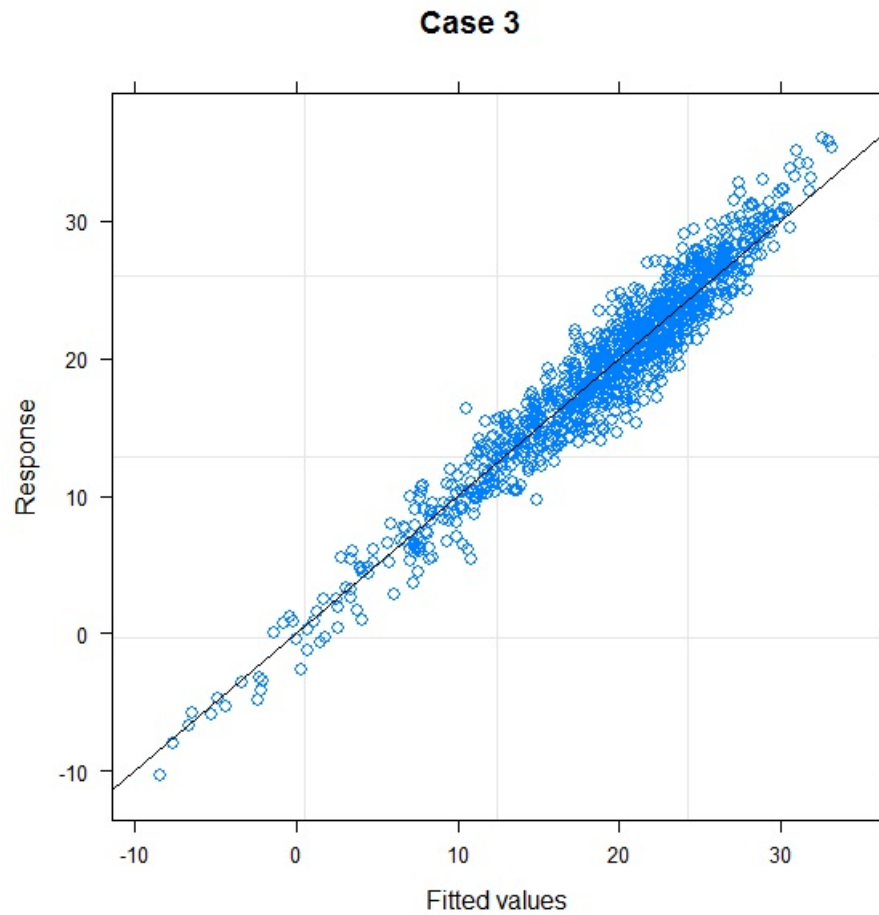


FIGURE 4.9: Asymmetric Distribution: Responses against Fitted Values

**The LME Robustness for Case 3**

$n_i$	$N = 6$				$N = 6$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.944	<b>0.923</b>	<b>0.925</b>	<b>0.920</b>	<b>0.924</b>	0.938	<b>0.889</b>	<b>0.919</b>
$\beta_1$	0.960	<b>0.923</b>	0.953	0.946	<b>0.896</b>	0.942	0.946	0.945
$\beta_2$	0.981	0.972	0.977	0.967	0.974	0.978	0.980	0.976
$\beta_3$	0.958	<b>0.921</b>	0.946	0.943	<b>0.905</b>	0.954	0.937	0.955

$N = 10$					$N = 10$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	<b>0.927</b>	0.938	0.939	0.930	<b>0.928</b>	<b>0.915</b>	0.933	<b>0.925</b>	
$\beta_1$	0.956	<b>0.926</b>	0.951	0.934	<b>0.926</b>	<b>0.922</b>	0.949	0.947	
$\beta_2$	0.957	0.958	0.950	0.964	0.959	0.957	0.954	0.965	
$\beta_3$	0.963	0.936	0.949	0.939	<b>0.924</b>	<b>0.920</b>	0.953	0.945	

$N = 16$					$N = 16$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.944	0.934	0.941	0.935	0.931	0.935	0.938	0.943	
$\beta_1$	0.951	<b>0.917</b>	0.946	0.950	<b>0.932</b>	0.936	0.957	0.950	
$\beta_2$	0.961	0.940	0.961	0.961	<b>0.935</b>	0.958	0.950	0.963	
$\beta_3$	0.942	<b>0.923</b>	0.948	0.956	0.931	<b>0.915</b>	0.950	0.961	

$N = 20$					$N = 20$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.952	0.934	0.950	0.947	0.933	0.942	0.931	0.948	
$\beta_1$	0.943	0.937	<b>0.932</b>	0.950	<b>0.928</b>	0.934	0.951	0.957	
$\beta_2$	0.959	0.952	0.960	0.957	0.938	0.955	0.946	0.953	
$\beta_3$	0.946	0.931	0.948	0.944	<b>0.926</b>	0.931	0.955	0.958	

$N = 40$					$N = 40$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.946	0.941	0.951	0.950	0.949	0.961	0.945	0.949	
$\beta_1$	0.942	0.955	0.948	0.937	0.944	0.942	0.958	0.952	
$\beta_2$	0.952	0.957	0.951	0.954	<b>0.933</b>	0.961	0.936	0.943	
$\beta_3$	0.954	0.942	0.944	0.939	0.936	0.937	0.948	0.953	

$N = 60$					$N = 60$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.948	0.950	0.943	0.961	0.947	0.951	0.947	0.950	
$\beta_1$	0.947	0.940	0.933	0.948	0.948	0.942	0.962	0.959	
$\beta_2$	0.942	0.943	0.953	0.966	0.958	0.961	0.955	0.958	
$\beta_3$	0.942	0.944	0.947	0.951	0.955	0.945	0.947	0.952	

$n_i$	$N = 100$				$N = 100$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.942	0.951	0.950	0.953	0.949	0.949	0.949	0.946
$\beta_1$	0.942	0.948	0.943	0.940	0.954	0.946	0.960	0.948
$\beta_2$	0.943	0.949	0.951	0.964	0.942	0.949	0.955	0.950
$\beta_3$	0.933	0.948	0.948	0.946	0.950	0.945	0.951	0.953

$n_i$	$N = 200$				$N = 200$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.959	0.953	0.948	0.955	0.950	0.950	0.954	0.945
$\beta_1$	0.958	0.949	0.946	0.958	0.960	0.938	0.948	0.928
$\beta_2$	0.952	0.955	0.955	0.967	0.946	0.946	0.959	0.946
$\beta_3$	0.965	0.947	0.946	0.943	0.950	0.950	0.949	0.947

## 4.4 Case 4: Gamma Distribution of Error

Gamma distribution was used in the case 4 as the probability density function of the error. The distribution is significantly more skewed than the other distributions of error considered in the previous cases. For more details, reference can be made to the equation 3.6.

### 4.4.1 Simulation Settings for Case 4

The simulation settings in this case were the same with the settings in the case 1, the validation model. As usual, the only difference was that gamma distribution was used as the distribution of the error. That is  $f(x) = \text{Gamma}(b, a)$ . Where  $b = \sqrt{\frac{1.9}{2}}$  and  $a = 2$ ;  $e_{ij} = X - \frac{a}{b}$ . Then  $E[e_{ij}] = 0$  and  $\text{Var}[e_{ij}] = 2\frac{2}{19}$ .

### Checking the Goodness of Fit for Case 4

Figure 4.10 depicts that the errors were drawn from a non-normal distribution as the quantiles of standard normal are not linearly matched with the standardized residuals. Hence, the errors are not normally distributed. The Shapiro-Wilk test was performed with the samples and the test statistic was  $W = 0.9601$  with  $p - \text{value} < 2.2e^{-16}$ . The  $p - \text{value}$  shows that the null hypothesis that the error samples were drawn from a normal distribution was rejected.

Figure 4.11 depicts that the random effects for intercept and slope were drawn from normal distributions.

Figure 4.12 shows that the responses matched with the fitted values in this case.

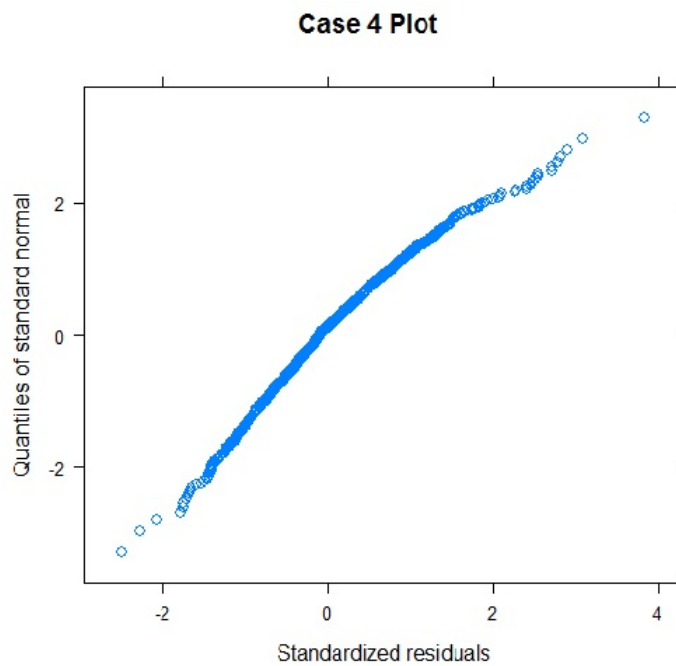


FIGURE 4.10: Gamma Distribution: qqnorm Plot for Residuals

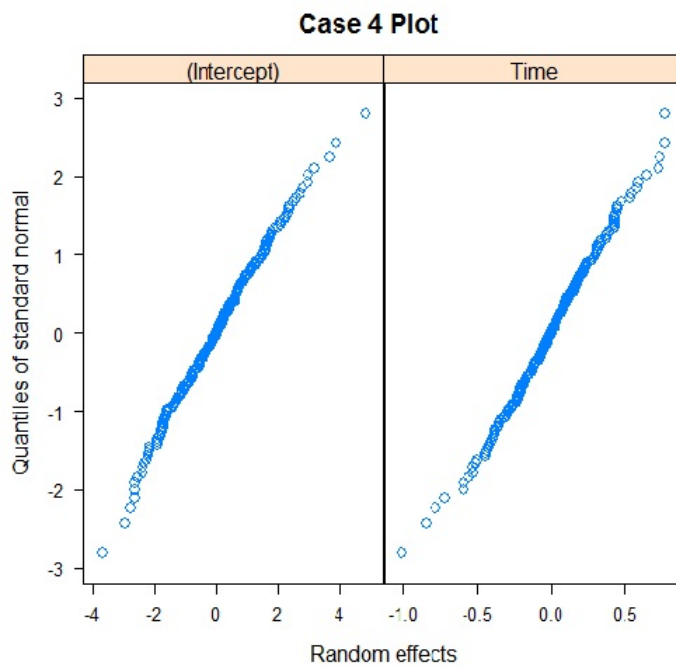


FIGURE 4.11: Gamma Distribution: qqnorm Plot for Random Effects

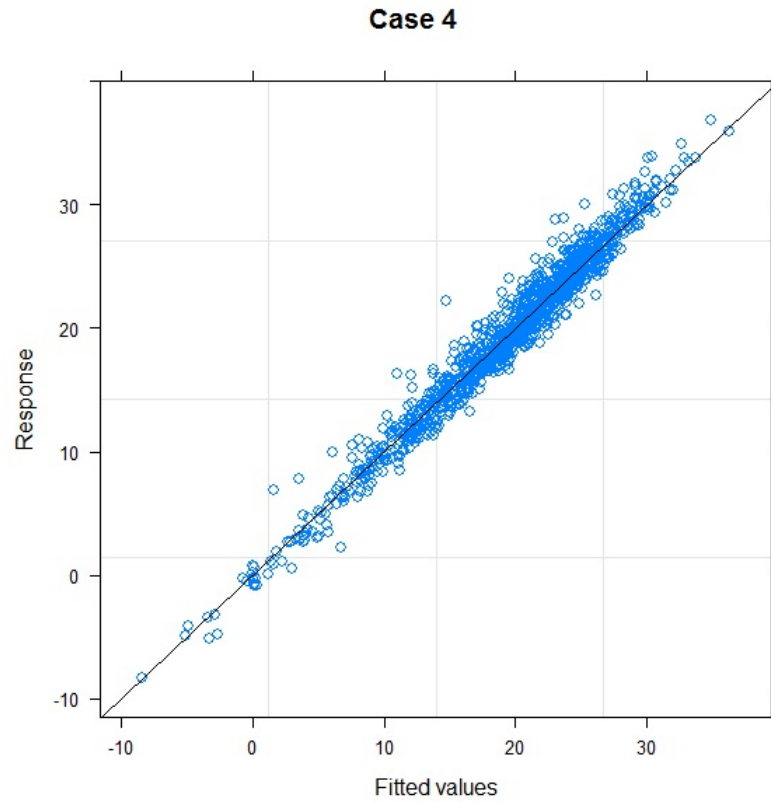


FIGURE 4.12: Gamma Distribution: Responses against Fitted Values

**The LME Robustness for Case 4**

$n_i$	$N = 6$				$N = 6$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.903</b>	<b>0.893</b>	<b>0.913</b>	<b>0.912</b>	<b>0.904</b>	<b>0.893</b>	<b>0.900</b>	<b>0.906</b>
$\beta_1$	<b>0.919</b>	<b>0.912</b>	0.952	<b>0.930</b>	<b>0.896</b>	<b>0.915</b>	0.936	0.957
$\beta_2$	0.961	0.957	0.963	0.962	0.975	0.963	0.956	0.969
$\beta_3$	<b>0.917</b>	<b>0.913</b>	0.956	0.952	<b>0.905</b>	<b>0.914</b>	0.960	0.960

$n_i$	$N = 10$				$N = 10$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.915</b>	0.931	<b>0.921</b>	<b>0.921</b>	0.934	0.943	<b>0.928</b>	<b>0.925</b>
$\beta_1$	<b>0.923</b>	<b>0.921</b>	0.942	0.950	0.933	<b>0.925</b>	0.951	0.947
$\beta_2$	0.963	0.954	0.950	0.951	0.956	0.970	0.953	0.955
$\beta_3$	0.937	0.934	0.937	0.942	<b>0.928</b>	0.943	0.957	0.946



$N = 16$					$N = 16$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.933	0.935	0.945	<b>0.929</b>	<b>0.929</b>	0.946	0.937	0.939
$\beta_1$	0.933	<b>0.929</b>	0.958	0.940	0.938	<b>0.932</b>	0.961	0.954
$\beta_2$	0.942	0.941	0.953	0.944	0.955	0.966	<b>0.932</b>	0.954
$\beta_3$	0.946	<b>0.928</b>	0.948	0.949	<b>0.929</b>	<b>0.915</b>	0.946	0.957

$N = 20$					$N = 20$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.930	0.931	0.943	0.952	0.951	<b>0.929</b>	<b>0.922</b>	0.943
$\beta_1$	0.934	<b>0.921</b>	0.940	0.949	0.940	0.937	0.961	0.950
$\beta_2$	0.939	0.940	0.950	0.953	0.946	0.944	0.956	0.953
$\beta_3$	<b>0.926</b>	<b>0.927</b>	0.942	0.958	0.944	0.936	0.944	0.957

$N = 40$					$N = 40$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.950	0.941	0.945	0.945	0.948	0.956	0.938	0.950
$\beta_1$	0.941	0.943	0.957	0.939	0.968	0.952	0.953	0.957
$\beta_2$	0.956	0.938	0.958	0.938	0.959	0.952	0.946	0.957
$\beta_3$	0.949	0.951	0.957	0.948	0.964	0.952	0.949	0.951

$N = 60$					$N = 60$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.947	0.935	0.952	0.948	0.954	0.945	0.946	0.950
$\beta_1$	0.936	0.940	0.948	0.960	0.952	0.939	0.944	0.957
$\beta_2$	0.944	0.947	0.950	0.951	0.951	0.936	0.939	0.946
$\beta_3$	0.947	0.934	0.950	0.950	0.940	0.945	0.950	0.957

$N = 100$					$N = 100$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.946	0.953	0.950	0.941	0.954	0.937	0.951	0.958
$\beta_1$	0.958	0.947	0.946	0.956	0.949	0.937	0.952	0.955
$\beta_2$	0.954	0.954	0.956	0.956	0.949	0.943	0.944	0.955
$\beta_3$	0.951	0.953	0.963	0.951	0.945	0.943	0.954	0.965

$n_i$	$N = 200$				$N = 200$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	0.941	0.952	0.949	0.946	0.946	0.954	0.952	0.945
$\beta_1$	0.949	0.957	0.945	0.960	0.939	0.954	0.956	0.947
$\beta_2$	0.940	0.950	0.952	0.943	0.959	0.952	0.944	0.961
$\beta_3$	0.960	0.951	0.944	0.937	0.947	0.948	0.946	0.951

## 4.5 Case 5: Exponential Distribution of Error

In this case, the most skewed data were generated. The exponential distribution was used. It was more skewed than the gamma distribution used in the case 4. For more details about the distribution, reference can be made to the equation 3.8.

### 4.5.1 Simulation Settings for Case 5

Also, the simulation settings used in the case 5 were the same with the settings used in the case 1, the validation model, except that the exponential probability density function was used as the distribution of the error. That is  $f(x) = \exp(-\lambda)$ ; where  $\lambda = 0.9$ ;  $e_{ij} = X - \frac{1}{\lambda}$ . Then  $E[e_{ij}] = 0$  and  $\text{Var}[e_{ij}] = \frac{1}{\lambda^2}$ .

#### Checking the Goodness of Fit for Case 5

The figure 4.13 depicts that the error samples were drawn from a non-normal distribution because the quantiles of standard normal were not linearly matched with standardized residuals. The Shapiro-Wilk test conducted on the samples confirmed that the errors were not from a normal probability distribution as the value of the statistic was  $W = 0.9601$  with  $p - \text{value} < 2.2e^{-16}$ . The value shows that the hypothesis that the samples were from a normal probability density function was rejected as the value is smaller than 0.05.

The figure 4.15 illustrates that responses correspond with the fitted values. The outliers are not present in the figure.

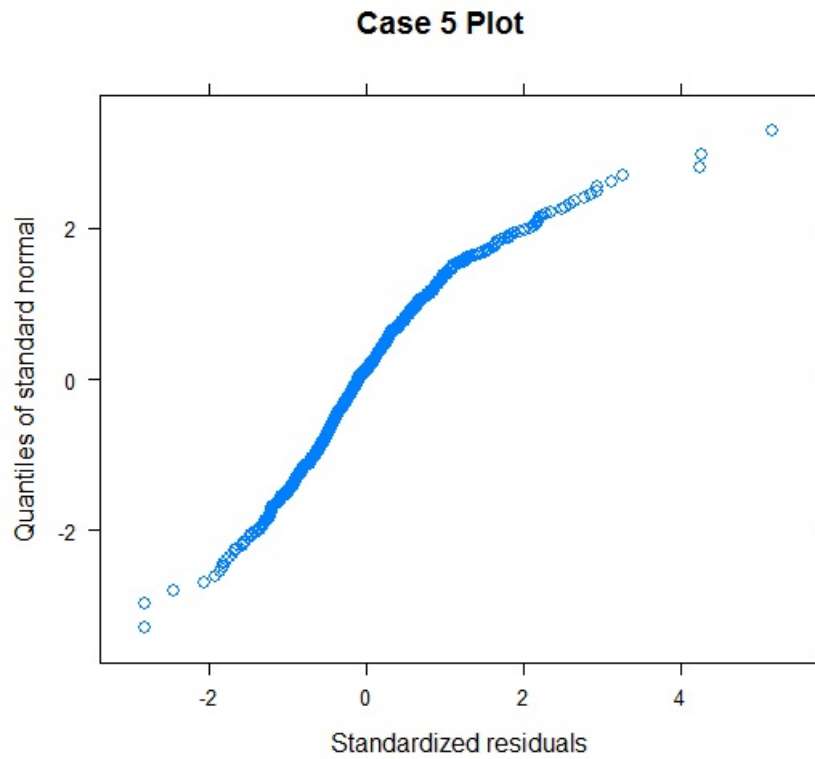


FIGURE 4.13: Exponential Distribution: qqnorm Plot for Residuals

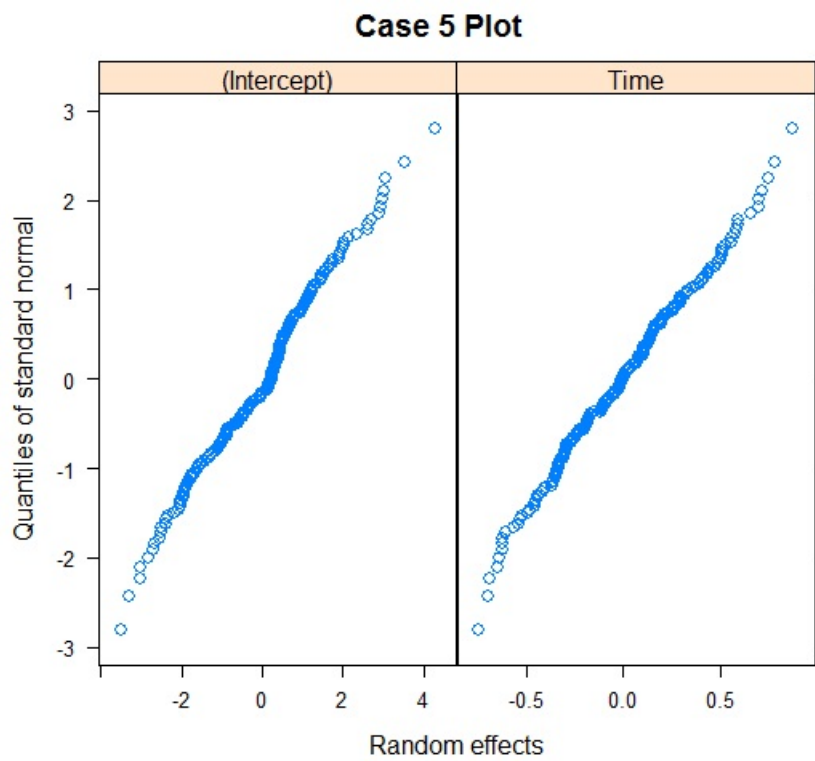


FIGURE 4.14: Exponential Distribution: qqnorm Plot for Random Effects

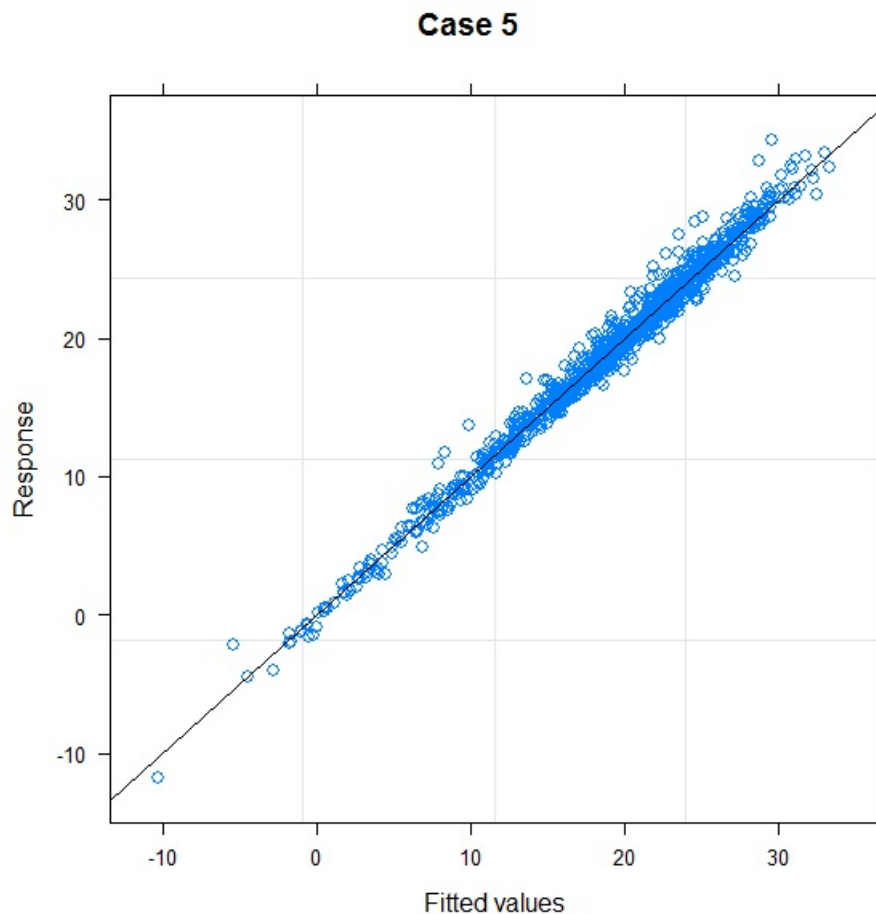


FIGURE 4.15: Exponential Distribution: Responses against Fitted Values

**The LME Robustness for the Case 5**

$N = 6$					$N = 6$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.892</b>	<b>0.902</b>	<b>0.887</b>	<b>0.909</b>	<b>0.898</b>	<b>0.895</b>	<b>0.907</b>	<b>0.886</b>
$\beta_1$	<b>0.896</b>	<b>0.923</b>	0.953	0.946	<b>0.897</b>	<b>0.908</b>	0.949	0.960
$\beta_2$	0.956	0.973	0.955	0.965	0.960	0.957	0.961	0.959
$\beta_3$	<b>0.901</b>	<b>0.927</b>	0.968	0.952	<b>0.900</b>	<b>0.905</b>	0.958	0.939

$N = 10$					$N = 10$			
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	8	8	3-8	3-8
$\beta_0$	<b>0.922</b>	<b>0.916</b>	<b>0.922</b>	<b>0.924</b>	<b>0.918</b>	<b>0.913</b>	<b>0.927</b>	<b>0.929</b>
$\beta_1$	<b>0.919</b>	0.934	0.942	0.942	<b>0.923</b>	<b>0.924</b>	0.944	0.946
$\beta_2$	0.952	0.953	0.956	0.946	0.955	0.951	0.963	0.954
$\beta_3$	<b>0.926</b>	<b>0.920</b>	0.963	0.942	<b>0.916</b>	0.932	0.952	0.950

$N = 16$					$N = 16$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.942	0.940	0.947	0.938	$\beta_0$	<b>0.922</b>	0.947	0.938	0.937
$\beta_1$	<b>0.915</b>	0.935	0.950	0.942	$\beta_1$	<b>0.918</b>	<b>0.923</b>	0.956	0.944
$\beta_2$	0.953	0.956	0.963	0.954	$\beta_2$	0.941	0.951	0.951	0.952
$\beta_3$	<b>0.926</b>	0.934	0.954	0.957	$\beta_3$	0.944	<b>0.923</b>	0.949	0.949
$N = 20$					$N = 20$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.945	0.942	0.940	0.941	$\beta_0$	0.930	0.934	0.942	0.936
$\beta_1$	0.936	0.942	0.958	0.952	$\beta_1$	<b>0.929</b>	<b>0.932</b>	0.943	0.955
$\beta_2$	0.953	0.951	0.962	0.954	$\beta_2$	0.953	0.947	0.950	0.952
$\beta_3$	0.936	0.936	0.963	0.942	$\beta_3$	0.938	0.935	0.941	0.955
$N = 40$					$N = 40$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.944	0.956	0.941	0.937	$\beta_0$	0.940	0.943	0.943	0.953
$\beta_1$	0.952	0.964	0.953	0.948	$\beta_1$	0.940	0.948	0.955	0.945
$\beta_2$	0.942	0.960	0.945	0.947	$\beta_2$	0.945	0.937	0.951	0.955
$\beta_3$	0.945	0.956	0.946	0.960	$\beta_3$	0.933	0.948	0.952	0.949
$N = 60$					$N = 60$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.950	0.950	0.944	0.952	$\beta_0$	0.957	0.947	0.950	0.941
$\beta_1$	0.941	0.959	0.951	0.953	$\beta_1$	0.945	0.949	0.952	0.942
$\beta_2$	0.948	0.953	0.950	0.958	$\beta_2$	0.956	0.953	0.958	0.965
$\beta_3$	0.949	0.960	0.956	0.963	$\beta_3$	0.956	0.948	0.954	0.942
$N = 100$					$N = 100$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.951	0.951	0.944	0.953	$\beta_0$	0.955	0.948	0.946	0.958
$\beta_1$	0.950	0.952	0.949	0.955	$\beta_1$	0.959	0.949	0.942	0.955
$\beta_2$	0.944	0.961	0.951	0.949	$\beta_2$	0.965	0.938	0.938	0.954
$\beta_3$	0.960	0.945	0.946	0.941	$\beta_3$	0.957	0.939	0.945	0.945
$N = 200$					$N = 200$				
	$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$		$RIRS_b$	$RI_b$	$RIRS_u$	$RI_u$
$n_i$	5	5	3-8	3-8	$n_i$	8	8	3-8	3-8
$\beta_0$	0.949	0.958	0.946	0.954	$\beta_0$	0.941	0.957	0.947	0.949
$\beta_1$	0.937	0.952	0.957	0.956	$\beta_1$	0.952	0.949	0.947	0.952
$\beta_2$	0.964	0.946	0.949	0.949	$\beta_2$	0.951	0.950	0.948	0.948
$\beta_3$	0.953	0.946	0.940	0.949	$\beta_3$	0.948	0.956	0.952	0.945

# Chapter 5

## Conclusions

The method and covariance structure used in the simulation were respectively restricted maximum likelihood (REML) (1.45) and general symmetric positive-definite matrix, (pdSym), which are defaults in the LME.

The rule of thumb,  $RT=[\tilde{p}-2\sigma, \tilde{p}+2\sigma]$ , is used to pinpoint an unusual estimate for the coverage probability,  $p$ ; where  $\tilde{p}$  is the average and  $\sigma$  is the standard deviation of the estimates for the probability. An interval of rule of thumb is created for each fixed effect in the model:  $\beta_0, \beta_1, \beta_2$ , and  $\beta_3$ . These are the averages of the estimates of coverage probability and rules of thumb for the fixed effects in the model;  $\beta_0 : \tilde{p}_0 = 0.949, RT_0 = [0.929, 0.966]$ ;  $\beta_1 : \tilde{p}_1 = 0.952, RT_1 = [0.932, 0.972]$ ;  $\beta_2 : \tilde{p}_2 = 0.963, RT_2 = [0.935, 0.991]$ ;  $\beta_3 : \tilde{p}_3 = 0.949, RT_3 = [0.930, 0.969]$ . Only the simulated results from the validation model were used with respect to the fixed effects to make the estimate for  $\tilde{p}$  and create the interval,  $RT$ , as they represent the true estimation properties of the linear mixed effects model, LME.

Using the intervals of rule of thumb for the fixed effects to assess the statistical uncertainty of the LME estimation property, the validation case shows a property by which a very few estimates for the coverage probability for the fixed effects fall outside the intervals of the rule of thumb. The falling of estimate outside the interval,  $RT$ , is observed among the estimates (of coverage probability) for the fixed effects in the random intercept, RI, model. In the tables, only the estimates which are below the lower boundary of the intervals (RTs) are indicated with bold face.

Generally, the simulation results (of the validation case) indicate that the validation case works as expected because the proportions of times the confidence intervals of the fixed-effects estimates contain the parameter values of the model converges to 0.95 (95 %), which is  $1 - \alpha$  confidence interval with the level,  $\alpha = 0.05$ . The details can be observed from the simulated results of the validation case.

In the case 2 where the symmetric corrupted normal distribution of error was used, the LME robustness is impaired when the number of units used in the simulation was set to  $N=6$  as it can be seen from the tables for case two (2) that many of the estimates fall outside their RT intervals though the fixed effect for discrete covariate ( $\beta_2$ ) remains highly robust as observed in the validation. The most affected are the fixed effect for the intercept ( $\beta_0$ ), time ( $\beta_1$ ), and interaction between discrete covariate and time ( $\beta_3$ ). However, the robustness was recovered as the number of units was increased from  $N=6$  to  $N=200$ . The details can be observed from the simulated results of the case two (2).

In the case 3 where the asymmetric corrupted normal distribution of error was used, the robustness, measured by coverage probability, of the fixed effect for the intercept ( $\beta_0$ ), slope ( $\beta_1$ ), interaction ( $\beta_3$ ) is impaired when  $N = 6$ . Meanwhile, the robustness of fixed effect for the discrete covariate,  $\beta_2$ , remains robust. While the number of units was increased from  $N=6$  to  $N=200$ , the robustness was recovered as in the case two. The details can be observed from the simulated results of the case three (3).

In the case 4 and case 5 where skewed responses and very skewed responses were respectively generated with gamma error distribution and exponential error distribution, the same patterns that were observed in the case 2 and case 3 were also observed in the case 4 and case 5.

When  $N = 6$ , another noticed property is that the robustness for the fixed effects for intercept and time were more impaired in the case 2 and case 5 where symmetric corrupted normal distribution and exponential distribution were used as the error distributions. This could be due to the fact that the symmetric corrupted normal distribution looks more corrupted than the asymmetric corrupted normal distribution and exponential distribution is more skewed than the gamma distribution.

The observations that are extracted from the simulation results coincide with the previous observations from the research conducted earlier by Jacqmin-Gadda H. et

al: that the fixed-effects parameters from the linear mixed effects model are robust irrespective of whether the error distribution is normal distribution or non-normal distribution [7].

Additionally, the robustness of the fixed-effects parameters from the LME model are generally more stable when the number of units is greater than or equal to sixty,  $N \geq 60$ . However, the robustness of the fixed effects for intercept ( $\beta_0$ ), time ( $\beta_1$ ), and interaction between time and discrete covariate ( $\beta_3$ ) are impaired when  $N < 60$  in the cases where the errors have non-normal or corrupted normal distribution. This could only be attributed to the growth curve model studied in this thesis research.

The reason why the LME robustness is recovered when the number of units,  $N$ , is increased, but impaired when the number of the units is decreased may be because of the central limit theorem. The central limit theorem seems manifesting itself in the equation 1.42. Hence, the estimate for the fixed-effects vector,  $\boldsymbol{\beta}$ , is unbiased and standard deviation of the estimate,  $SD(\boldsymbol{\beta})$ , is fairly realistic which lead to making a realistically better confidence interval when there is a sufficient number of units irrespective of whether the errors have normal or non-normal distributions. However, the confidence interval may shift to the right because of larger standard deviation (larger variance) and the estimates for the fixed effects which are close to the lower boundary of (confidence) interval may fall out of the interval, which reduces the measure of robustness when the number of units is not sufficient.



# Appendix A

## Appendix

The function, CoverageProbability, is used to estimate the coverage probability.

---

```
*****
How to run the function:
Choose the option:
  1 - Validation model
  2 - Corrupted distribution: symmetric distribution
  3 - Asymmetric distribution
  4 - Skewed distribution: Gamma
  5 - Very Skewed distribution: Exponential
Example, option <- 1

iteration <- 1000

N_Group <- 10,20...or 200 % Number of Units. Note: Any even number.

TimeVector <- c(0,10,20,30,40) or c(0,10,20,30,40,50,60,70)
MAX <- 9 % The maximum column number of a matrix
Step_Size <- 10
Start <- 0
*****

CoverageProbability <- function(option,iteration, N_Groups, TimeVector, MAX, Step
  _Size,Start){

#Creation of Vector Variables
Bool.Vector1 <- vector(mode = "integer", length = iteration)
Bool.Vector2 <- vector(mode = "integer", length = iteration)
Bool.Vector3 <- vector(mode = "integer", length = iteration)
Bool.Vector4 <- vector(mode = "integer", length = iteration)
Bool.Vector5 <- vector(mode = "integer", length = iteration)
Bool.Vector6 <- vector(mode = "integer", length = iteration)
Bool.Vector7 <- vector(mode = "integer", length = iteration)
Bool.Vector8 <- vector(mode = "integer", length = iteration)
Bool.Vector9 <- vector(mode = "integer", length = iteration)
```

```

Bool.Vector10 <-vector(mode = "integer", length = iteration)
Bool.Vector11 <-vector(mode = "integer", length = iteration)
Bool.Vector12 <-vector(mode = "integer", length = iteration)
Bool.Vector13 <-vector(mode = "integer", length = iteration)
Bool.Vector14<-vector(mode = "integer", length = iteration)
Bool.Vector15<-vector(mode = "integer", length = iteration)
Bool.Vector16<- vector(mode = "integer", length = iteration)

i <- 1
B0 <- 27
B1 <- -0.2
B2 <- -2
B3 <- - 0.15

if(option==1){

  #For Balanced with Random Intercept and Random Slope

  for(i in 1:iteration){
    data <- BalancedData(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector1[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector2[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector3[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector4[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( data.lme)!= 'try-error' & class( interval.BalancedData)!= '
try-error'){

      Bool.Vector1[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector2[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector3[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector4[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

  #For Balanced Without Random Slope

  for(i in 1:iteration){
    data <- BalancedData_RI(N_Groups,TimeVector)

```

```

data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE)
)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector5[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector6[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector7[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector8[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector5[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector6[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector7[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector8[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}

}

#For Unbalanced with Random Intercept and Random Slope

for(i in 1:iteration){
  data <- UnbalancedData(N_Groups,MAX,Step_Size,Start)
  data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
  interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

  if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
    Bool.Vector9[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector10[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector11[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector12[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  }

  else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
    Bool.Vector9[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)

```

```

    Bool.Vector10[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
    Bool.Vector11[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
    Bool.Vector12[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
  }
}

#For Unbalanced without Random Slope

for(i in 1:iteration){
data <- UnbalancedData_RI(N_Groups,MAX,Step_Size,Start)
data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE)
)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector13[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector14[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector15[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector16[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector13[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector14[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector15[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector16[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}
}

}

if(option==2){

#For Balanced With Random Intercept and Random Slope

for(i in 1:iteration){
data <- SymBalancedData(N_Groups,TimeVector)
data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

```

```

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector1[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector2[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector3[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector4[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector1[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector2[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector3[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector4[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

  #For Balanced Without Random Slope

  for(i in 1:iteration){
    data <- SymBalancedData_RI(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector5[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector6[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector7[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector8[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector5[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector6[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector7[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector8[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)

```

```

}

}

#For Unbalanced with Random Intercept and Random Slope

for(i in 1:iteration){
data <- SymUnbalancedData(N_Groups,MAX,Step_Size,Start)
data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector9[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector10[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector11[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector12[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector9[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector10[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector11[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector12[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}
}

#For Unbalanced without Random Slope

for(i in 1:iteration){
data <- SymUnbalancedData_RI(N_Groups,MAX,Step_Size,Start)
data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector13[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector14[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector15[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}
}

```

```

    Bool.Vector16[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  }

  else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
    Bool.Vector13[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
    Bool.Vector14[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
    Bool.Vector15[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
    Bool.Vector16[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
  }
}

}

if(option==3){
#For Balanced With Random Intercept and Random Slope

  for(i in 1:iteration){
    data <- AsymBalancedData(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector1[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector2[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector3[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector4[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector1[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector2[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector3[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector4[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

#For Balanced Without Random Slope

```

```

for(i in 1:iteration){
  data <- AsymBalancedData_RI(N_Groups, TimeVector)
  data.lme <- try(lme(Response~Time*Type, data=data, random=~1|Id), silent=TRUE)
)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector5[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector6[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector7[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector8[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector5[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector6[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector7[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector8[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}

}

#For Unbalanced with Random Intercept and Random Slope

for(i in 1:iteration){
  data <- AsymUnbalancedData(N_Groups, MAX, Step_Size, Start)
  data.lme <- try(lme(Response~Time*Type, data=data, random=~Time|Id), silent=
TRUE)
  interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

  if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
    Bool.Vector9[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector10[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector11[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector12[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  }
}

```



```

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector9[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector10[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector11[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector12[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

  #For Unbalanced without Random Slope

  for(i in 1:iteration){
    data <- AsymUnbalancedData_RI(N_Groups,MAX,Step_Size,Start)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector13[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector14[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector15[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector16[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector13[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector14[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector15[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector16[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }
}

if(option==4){
  #For Balanced With Random Intercept and Random Slope

  for(i in 1:iteration){
    data <- GamBalancedData(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)
  }
}

```

```

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector1[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector2[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector3[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector4[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector1[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector2[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector3[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector4[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

  #For Balanced Without Random Slope

  for(i in 1:iteration){
    data <- GamBalancedData_RI(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector5[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector6[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector7[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector8[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector5[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector6[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector7[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)

```

```

    Bool.Vector8[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
  }

}

#For Unbalanced with Random Intercept and Random Slope

for(i in 1:iteration){
data <- GamUnbalancedData(N_Groups,MAX,Step_Size,Start)
data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector9[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector10[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector11[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector12[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector9[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector10[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector11[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector12[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}
}

#For Unbalanced without Random Slope

for(i in 1:iteration){
data <- GamUnbalancedData_RI(N_Groups,MAX,Step_Size,Start)
data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector13[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector14[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}
}

```

```

    Bool.Vector15[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector16[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  }

  else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
    Bool.Vector13[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
    Bool.Vector14[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
    Bool.Vector15[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
    Bool.Vector16[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
  }
}

}

if(option==5){
#For Balanced With Random Intercept and Random Slope

  for(i in 1:iteration){
    data <- ExpoBalancedData(N_Groups,TimeVector)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~Time|Id), silent=
TRUE)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector1[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector2[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector3[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector4[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector1[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector2[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector3[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector4[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }
}

#For Balanced Without Random Slope

```

```

for(i in 1:iteration){
  data <- ExpoBalancedData_RI(N_Groups, TimeVector)
  data.lme <- try(lme(Response~Time*Type, data=data, random=~1|Id), silent=TRUE)
}
interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
  Bool.Vector5[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector6[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector7[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  Bool.Vector8[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
}

else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
  Bool.Vector5[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
  Bool.Vector6[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
  Bool.Vector7[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
  Bool.Vector8[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
}

}

#For Unbalanced with Random Intercept and Random Slope

for(i in 1:iteration){
  data <- ExpoUnbalancedData(N_Groups, MAX, Step_Size, Start)
  data.lme <- try(lme(Response~Time*Type, data=data, random=~Time|Id), silent=
TRUE)
  interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

  if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
    Bool.Vector9[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector10[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector11[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    Bool.Vector12[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
  }
}

```

```

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector9[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector10[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector11[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector12[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }

  #For Unbalanced without Random Slope

  for(i in 1:iteration){
    data <- ExpoUnbalancedData_RI(N_Groups,MAX,Step_Size,Start)
    data.lme <- try(lme(Response~Time*Type, data=data,random=~1|Id), silent=TRUE
)
    interval.BalancedData <-try(intervals(data.lme), silent=TRUE)

    if (class(data.lme)== 'try-error' | class( interval.BalancedData)== 'try-
error'){
      Bool.Vector13[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector14[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector15[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
      Bool.Vector16[i] <- (3*(class(data.lme)== 'try-error' | class( interval.
BalancedData)== 'try-error'))
    }

    else if(class( interval.BalancedData)!= 'try-error' & class( interval.
BalancedData)!= 'try-error'){
      Bool.Vector13[i] <- 1*(interval.BalancedData$fix[1,1] < B0 & interval.
BalancedData$fix[1,3] > B0)
      Bool.Vector14[i] <- 1*(interval.BalancedData$fix[2,1] < B1 & interval.
BalancedData$fix[2,3] > B1)
      Bool.Vector15[i] <- 1*(interval.BalancedData$fix[3,1] < B2 & interval.
BalancedData$fix[3,3] > B2)
      Bool.Vector16[i] <- 1*(interval.BalancedData$fix[4,1] < B3 & interval.
BalancedData$fix[4,3] > B3)
    }
  }
}

B_IRS<-c(mean(Bool.Vector1[Bool.Vector1!=3]),mean(Bool.Vector2[Bool.Vector2!=3])
,mean(Bool.Vector3[Bool.Vector3!=3]),mean(Bool.Vector4[Bool.Vector4!=3]))
U_IRS <-c(mean(Bool.Vector5[Bool.Vector5!=3]),mean(Bool.Vector6[Bool.Vector6!=
3]),mean(Bool.Vector7[Bool.Vector7!=3]),mean(Bool.Vector8[Bool.Vector8!=3]))
B_RI<-c(mean(Bool.Vector9[Bool.Vector9!=3]),mean(Bool.Vector10[Bool.Vector10!=3])
,mean(Bool.Vector11[Bool.Vector11!=3]),mean(Bool.Vector12[Bool.Vector12!=3]))

```

---

```
U_RI<-c(mean(Bool.Vector13[Bool.Vector13!=3]),mean(Bool.Vector14[Bool.Vector14!=
  3]),mean(Bool.Vector15[Bool.Vector15!=3]),mean(Bool.Vector16[Bool.Vector16!=
  3]))
Coverage.Prob.Matrix <- cbind(B_RIRS,B_RI,U_RIRS,U_RI)
return(signif(Coverage.Prob.Matrix,digits=3))
}
```

---

LISTING A.1: Code for Simulating the LME Robustness

# Appendix B

## Appendix

The function, `BalancedData`, generates balanced data in a dataframe. There are eight other functions that are derived from this function with respect to different distributions of errors and the random slope, but they are not included here. They are saved to the CD.

---

```
*****
How to run the function

N_Data_Groups <- 10,20,or 200 % Number of units. Note: any even number
FollowUp_Times <- c(0,10,20,30,40) or c(0,10,20,30,40,50,60,70)
*****

BalancedData<-function(N_Data_groups,FollowUp_Times){
  ##Header Files
  library(nlme)
  N_Measures <-length(FollowUp_Times)
  N_Dataset <-N_Data_groups*N_Measures
  LikeForLike.Response <-matrix(nrow=N_Data_groups,ncol=N_Measures)
  Week <- matrix(nrow=N_Data_groups,ncol=N_Measures)
  discrete.covariate <-matrix(nrow=N_Data_groups,ncol=N_Measures)
  PatId <- matrix(nrow=N_Data_groups,ncol=N_Measures)
  Samples <- matrix(nrow=N_Data_groups,ncol=N_Measures)
  ##Generating samples for discrete covariate: categorical variable X
  Discrete.sample <- sample(rep(0:1,each=(N_Dataset/2)))
  discrete.covariate <- matrix(sort(Discrete.sample), nrow=N_Data_groups, ncol=N_
    Measures, byrow=TRUE)
  B.Intercept <- 27
  B.Time <- -0.2
  B.Xi <- -2
  B.X.time <- -0.15
  ##Assuming Correlated Random Effects and sampling from multivariate normal
  distribution
  S.Intercept <- 3*3
}
```



---

```

S.Slope<- 0.1*0.1
Corr.01 <- -0.75
Cov.01 <- Corr.01*3*0.1
Sigma <- matrix(c(S.Intercept,Cov.01,Cov.01,S.Slope),ncol=2,nrow=2)
U<-t(chol(Sigma))
i<-1
j <-1
for(i in 1:N_Data_groups){
  FollowUp_Times
  b.rand <- U%*%c(rnorm(2,0,1))
  for(j in 1:N_Measures){
    Errors <- rnorm(1, mean=0, sd=5)
    Week[i,j] <- FollowUp_Times[j]
    PatId[i,j] <- i
    LikeForLike.Response[i,j] <- ((B.Intercept + b.rand[1]) + ((B.Time + b.rand
[2])*FollowUp_Times[j])+ (B.Xi*discrete.covariate[i,j])+ (B.X.time*FollowUp_
Times[j]*discrete.covariate[i,j])+Errors)# equation 1
  }
}
discrete.covariate <- as.vector(t(discrete.covariate))
## Arranging the different Data in a Data frame
Response <- as.vector(t(LikeForLike.Response))
Type <- sort(as.factor(discrete.covariate ))
Time<- as.vector(t(Week))
Id <- as.vector(t(PatId))
Dataset <- data.frame(Id,Type,Time,Response )
Dataset[, 'Type'] <- as.factor(Dataset[, 'Type'])
return(Dataset)
}

```

---

LISTING B.1: Code for Generating Balanced Data

# Appendix C

## Appendix

The function, `UnbalancedData`, generates unbalanced data in a dataframe. There are eight other functions that are derived from this function with respect to different distributions of errors, but they are not included here. They are saved to the CD.

---

```
*****
How to run the function

DATA_GROUP <- 10,20,or 200 % Number of units. Note: any even number
PRE_SET_MAX <- 9 % Note always choose 9
STEP_SIZE <- 10
FROM <- 0
*****

UnbalancedData<- function(DATA_GROUP,PRE_SET_MAX,STEP_SIZE,FROM )
{##Header File
library(nlme)
UNBALANCED_CD4<- matrix(nrow=DATA_GROUP,ncol=PRE_SET_MAX, byrow=TRUE)
WEEK <- matrix(nrow=DATA_GROUP,ncol=PRE_SET_MAX, byrow=TRUE)
GROUPINGS <- matrix(nrow=DATA_GROUP,ncol=PRE_SET_MAX, byrow=TRUE)
TRTART <- matrix(nrow=DATA_GROUP,ncol=PRE_SET_MAX, byrow=TRUE)
j <-1
i<-1
TimePoints <- 0
B.Intercept <- 27
B.Time <- -0.2
B.Xi <- -2
B.X.time <- -0.15
##Assuming Correlated Random Effects and sampling from multivariate normal
distribution
S.Intercept <- 3*3
S.Slope <- 0.1*0.1
Corr.01 <- -0.75
Cov.01 <- Corr.01*3*0.1
```

```

Sigma <- matrix(c(S.Intercept,Cov.01,Cov.01,S.Slope),ncol=2,nrow=2)
U<-t(chol(Sigma))
## Random Generation of Samples for Discrete Variable, which is later converted
  to factors.
Trt<-matrix(sort(sample(rep(0:1,each=(PRE_SET_MAX*DATA_GROUP/2))))),nrow=DATA_
  GROUP, ncol=PRE_SET_MAX,byrow=TRUE) #matrix(runif(PRE_SET_MAX*DATA_GROUP,min
  =0, max=1),nrow=DATA_GROUP, ncol=PRE_SET_MAX,byrow=TRUE)
for(i in 1:DATA_GROUP){
n <- runif(1, min=0, max=1)
n_i1 <- 3*(0 <= n & n <0.2)
n_i2 <- 4*(0.2 <= n & n < 0.4)
n_i3 <- 5*(0.4 <= n & n < 0.6)
n_i4 <- 6*(0.6 <= n & n < 0.8)
n_i5 <- 7*(0.8 <= n & n < 1)
n_i <- (n_i1 + n_i2 + n_i3 + n_i4 +n_i5) # the number of measurement
TO <- STEP_SIZE*(n_i-1)
TimePoints <- seq(from=FROM,to=TO, by=STEP_SIZE)
b.rand <- U%*%c(rnorm(2,0,1)) # Multivariate Sampling - Just one sample for each
  random effect
for(j in 1:PRE_SET_MAX) {
  if (j <= n_i)
    {
      Errors <-rnorm(1,mean=0,sd=5)
      UNBALANCED_CD4[i,j]<- ((B.Intercept + b.rand[1] ) + ((B.Time + b.rand
[2])*TimePoints[j]) + (B.Xi*Trt[i,j]) + (B.X.time*Trt[i,j]*TimePoints[j])+
Errors)
      TRTART[i,j] <-Trt[i,j]
      WEEK[i,j] <- TimePoints[j]
      GROUPINGS[i,j] <- i
    }
  }
}
## Building the Data Frame to contain all datasets in a one frame
Id <- as.vector(t(GROUPINGS)) # converting each matrix to vector, so that they
  can fit into dataframe structure.
Type <- as.factor(sort(Trt))
Time <- as.vector(t(WEEK))
Response <- as.vector(t(UNBALANCED_CD4))
Dataframe <- data.frame(Id,Type,Time,Response)
NA.in.Dataframe.Removed <-na.omit(Dataframe)
NA.in.Dataframe.Removed[, 'Type'] <- as.factor(NA.in.Dataframe.Removed[, 'Type'])
return(NA.in.Dataframe.Removed)
}

```

LISTING C.1: Code for Generating Unbalanced Data

# Bibliography

- [1] Veierød M.B et al. *Medical statistics in clinical and epidemiological research*. Gyldendal akademisk, 2012.
- [2] Fitzmaurice M.G et al. *Applied longitudinal data analysis*, volume 998. John Wiley & Sons, 2012.
- [3] Johnson R.A. et al. *Applied multivariate statistical analysis*, volume 4. Prentice hall Englewood Cliffs, NJ, 1992.
- [4] Pinheiro J. et al. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2006.
- [5] Østgård E.T. Statistical modeling and analysis of repeated measures, using the linear mixed effects model.
- [6] Verbeke G. Linear mixed models for longitudinal data. In *Linear Mixed Models in Practice*, pages 63–153. Springer, 1997.
- [7] Jacqmin-Gadda H. et al. Robustness of the linear mixed model to misspecified error distribution. *Computational Statistics & Data Analysis*, 51(10):5142–5154, 2007.
- [8] Wikipedia. Shapiro–wilk test, 2016. URL [https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk\\_test](https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test). [Online; accessed 16th June, 2016].
- [9] Wikipedia. Montgomery–Åsberg depression rating scale, 2016. URL [https://en.wikipedia.org/wiki/Montgomery%E2%80%93Åsberg\\_Depression\\_Rating\\_Scale](https://en.wikipedia.org/wiki/Montgomery%E2%80%93Åsberg_Depression_Rating_Scale). [Online; accessed 16th June, 2016].