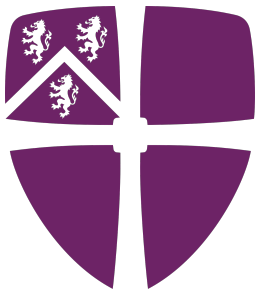


Durham University

Using Mixed Effect Models for the Analysis of Longitudinal Data

Project IV - MATH4072

Jack Lloyd

Department of Mathematical Sciences

2023-2024

Supervised by Dr. Reza Drikvandi

Acknowledgements

I would like to thank Dr. Reza Drikvandi for his guidance and supervision throughout the entire process of creating this report. I would also like to thank my colleague, Benjamin Drabble, for his cooperation in the development of the `R` functions discussed in Section 6.3.1. Thank you both for your support.

Plagiarism Declaration

This piece of work is a result of my own work and I have complied with the Department's guidance on multiple submission and on the use of AI tools. Material from the work of others not involved in the project has been acknowledged, quotations and paraphrases suitably indicated, and all uses of AI tools have been declared.

Contents

List of Figures

List of Tables

1 Introduction

Longitudinal data studies have played a crucial role in society for over a century and have been used for research in key areas such as politics, psychology, sociology and medicine. Longitudinal data is data that contains repeated observations of the same subjects over time. Analysing how individuals change over time allows researchers to draw key conclusions about how different factors are correlated. There are countless examples where the analysis of longitudinal data has been critical in the publication of groundbreaking research. The Framingham Heart Study [1] is a longitudinal study run by the National Heart, Lung and Blood institute that began in 1948 with 5,209 adult subjects in Framingham, Massachusetts. Back in 1948, little was known about cardiovascular disease. It was (correctly) assumed that heart health could be influenced by lifestyle and environmental factors, however there was little understanding of disease prevention. In the first twenty years of the study, analysis of the data allowed researchers to draw several key conclusions which are still at the centre of cardiovascular disease prevention. Researchers at Framingham were some of the first to propose that cigarette smoking, high blood pressure and cholesterol levels, obesity and a lack of physical activity all increased the risk of heart disease. These conclusions could only be reached by having observed thousands of individuals at multiple time points over ten years and modelling the response based on many factors.

This poses a natural question: how does one model longitudinal data? Simple modelling techniques, such as linear regression, do not suffice. The main reason behind this is because in such models, no consideration is given to the correlation structure between individuals in a study and their measurements. Laird and Ware (1982) [2] argue that "Models for the analysis of longitudinal data must recognise the relationship between serial observations on the same unit." It is clear that if we observe the same individual over time, their measurements are going to be correlated in some way. It is also naive to assume that different individuals in a study are not correlated. In the same paper, Laird and Ware (1982) introduced a model which accounts for these random sources of variation. This model is now widely known as a linear mixed effects model (LMEM) and has been a focus for examination by statisticians for the past fifty years. Mixed effects models have been favoured due to their flexibility and ability to deal with longitudinal datasets containing missing values, which are extremely common. However, more recently researchers have highlighted certain issues with mixed effects models and have produced work which provides solutions to these problems. In the first three months of 2024 alone, thousands of papers on the topic of mixed effects models have been published, which demonstrates their continued relevance in modern work.

This report contains five main chapters. We will begin by introducing the datasets we will use to demonstrate theoretical ideas related to mixed effects models; this chapter will also include plots and statistics from each dataset to allow the reader to become familiar with them. We will then present the initial theory behind linear mixed effects models via a two-stage method as done by Verbeke and Molenberghs (2000) [3] and fit an LMEM to a dataset. We will also explore different models and model selection criteria which indicate the performance of a model. The subsequent chapter will consider deeper theory focused on specific parameters of the linear mixed effects model and methods to avoid certain issues faced when fitting LMEMs. A further chapter will highlight basic theory on non-linear mixed effects models (NLMEMs) and look to fit an NLMEM to a dataset. In practice, measurements from individuals in a longitudinal dataset do not follow a linear relationship so it is crucial to consider how we extend the model to a non-linear case. Our final chapter will focus on missing data in longitudinal datasets, which is often a key issue in modelling longitudinal data.

The contributions made in this chapter will look to build on theory discussed in *Longitudinal Data Analysis* (Fitzmaurice et al. 2009) [4]. The topics outlined in this report aim to develop the reader's understanding of mixed effects models and the issues faced when modelling longitudinal data.

2 The Datasets

Before considering how we will model longitudinal data, and introducing the rich theory that comes with such a task, we will introduce the datasets that we will apply the ideas to. We will use a two datasets in order to allow the reader to feel comfortable with the data and so they can focus more on the theory. We have chosen these datasets as they allow us to apply different concepts, such as dealing with missing data and data that follows a non-linear relationship. We will also use simulated datasets in certain areas of the report.

2.1 The `ChickWeight` Dataset

The first dataset we will introduce is the `ChickWeight` data. This is longitudinal data of the body weights (in grams) of 50 chicks measured at birth and every two days afterwards until day 20. They were also measured on day 21. Assuming they survived the 21 days, they were measured on 12 different occasions. Out of the 50 chicks, five did not survive for at least 21 days. We will refer to these as 'dropout' cases. From birth, the chicks were put on one of four diets, each consisting of a different protein. We will refer to these diets as Diet A, Diet B, Diet C and Diet D.

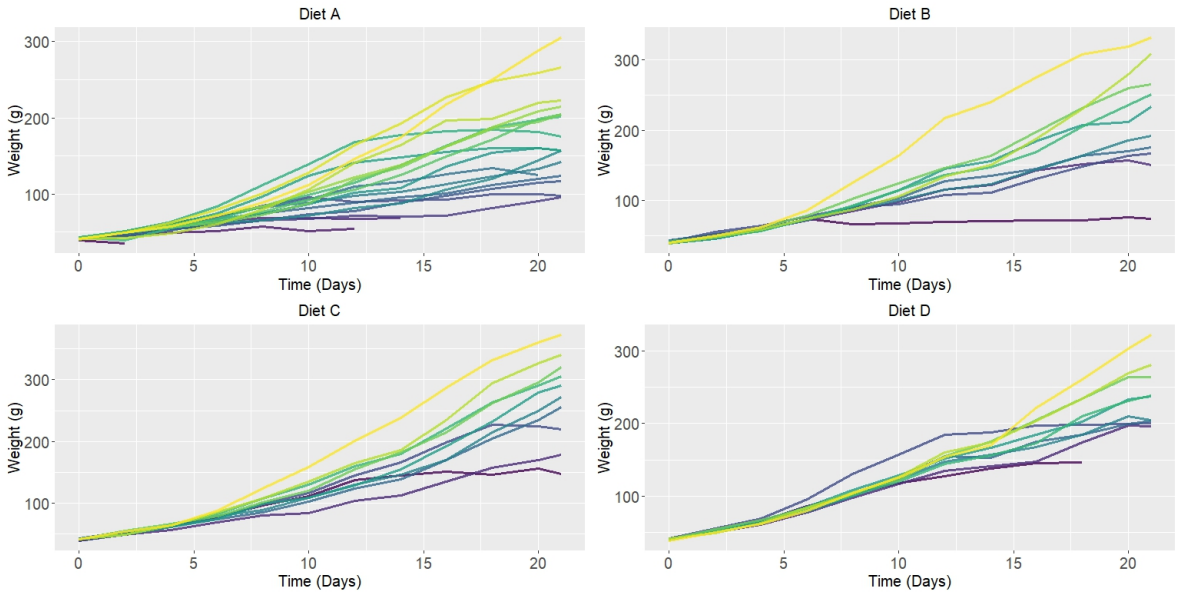


Figure 2.1: Chick Weights Plotted by Diet

Figure 2.1 illustrates key trends in our data. Note that there are not an equal number of chicks on each diet. We will sum up important statistics of the data in Table 2.1 below.

	Diet A	Diet B	Diet C	Diet D
Number of Subjects	20	10	10	10
Number of 'Dropouts'	4	0	0	1
Mean Final Weight$^{(*)}$ (g)	171.87	214.70	270.30	238.56
Maximum Final Weight$^{(*)}$ (g)	305	331	373	322
Mean Change in Weight Between Occasions$^{(**)}$ (g)	10.75	16.65	21.90	17.71

Table 2.1: Key Statistics from the `ChickWeight` Dataset

For statistics marked with $(*)$, we have omitted dropouts as this would greatly skew the mean due to having some chicks drop out at early points, therefore having an extremely low final weight. For our calculation marked $(**)$, all subjects have been taken into account. However, when considering the changes between occasions, one must note that, for subjects who did not drop out, the final difference between occasions was one day, namely from day 20 to day 21. For this reason, we have multiplied this difference by two as this accounts for the time period being half of the rest of the time periods, namely two days. We then calculated the average of all the differences.

For subjects that did drop out, the mean was calculated in the usual manner: the sum of the differences divided by the number of differences. Once we had calculated the mean for each chick, we then grouped the chicks by diet and calculated the mean for each diet group.

2.2 The `Theophylline` Dataset

The second dataset that we will discuss in this report is the `Theophylline` dataset. Theophylline is a drug that blocks the adenosine receptors and is used to treat chronic obstructive pulmonary disease and asthma. The `Theophylline` dataset contains data from twelve individuals who were given oral doses of 320mg of the theophylline drug and had their serum concentrations measured at ten different time points over the following 25 hours. The exact time points differ slightly between individuals. They also had their body weights measured. This dataset is complete so contains no missing data. We will use this dataset to apply multiple ideas from theory, in particular in our exploration of non-linear mixed effects models (Chapter 6). Table 2.2 contains key statistics from the dataset and Figure 2.2 visualises the data.

Number of Subjects	12
Highest Maximum Serum Concentration (mg/L)	11.4
Lowest Maximum Serum Concentration (mg/L)	6.44
Average Time Taken to Hit Maximum Serum Concentration (Hours)	1.788
Maximum Difference Between Initial and Final Serum Concentration (mg/L)	6.25
Minimum Difference Between Initial and Final Serum Concentration (mg/L)	0.08

Table 2.2: Key Statistics from the Theophylline Dataset

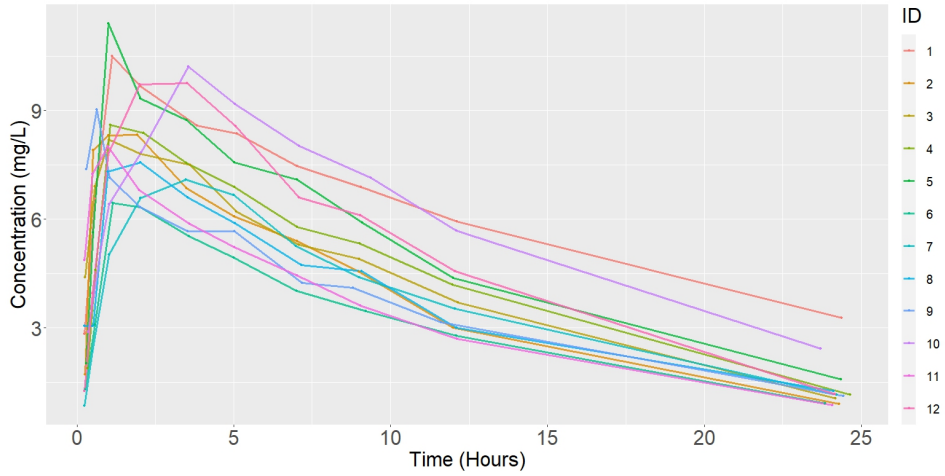


Figure 2.2: Serum Concentration Over Time for Each Individual

One can see that each individual exhibits a non-linear relationship between serum concentration and time. We can see that each individual's serum concentration initially increases at a similar rate, however the time taken to hit the peak concentration level differs substantially between subjects. For many of our subjects, by the 25^{th} hour their serum concentration has returned to its initial value. This is not the case for individuals 1 and 10, which suggests that they could be outliers in the trial. We will return to the `Theophylline` dataset in Chapter 5 where we look to fit a non-linear mixed effects model.

3 Linear Mixed Effect Models

We will first explore the need for linear mixed effects models (LMEMs) through the use of an example. Consider the `ChickWeight` dataset. If we consider an individual chick i, it should be clear that its measurements are going to be correlated. For example, its weight at time $t = 2$ is going to affect its weight at time $t = 4$. However, we must also acknowledge that there will be correlation between the chicks which may not be the same for each pair of chicks. It may not always be clear what causes these different correlations. It could be due to their diet or simply the environment they have been in during the trial. In longitudinal data studies, these sources of variation are often unknown. This poses the question: how can a model be created that accounts for the different unknown sources of variability?

Figure 3.1 highlights the futility of using a simple linear regression model to analyse the `ChickWeight` dataset. All we infer from this fitted model is that, on average, there is a positive linear relationship between the age and the weight of a chick. We do not learn anything about the individual chicks and the model does not account for any unknown variability between subjects. Hence, we require a more complex model to provide a solution to the problems that come with modelling longitudinal data. This section will discuss the basic theory of linear mixed effects models. A mixed effect model will allow an average intercept and slope to be fitted as fixed effects, but a random intercept and slope can also be included to account for the random unknown sources of variability.

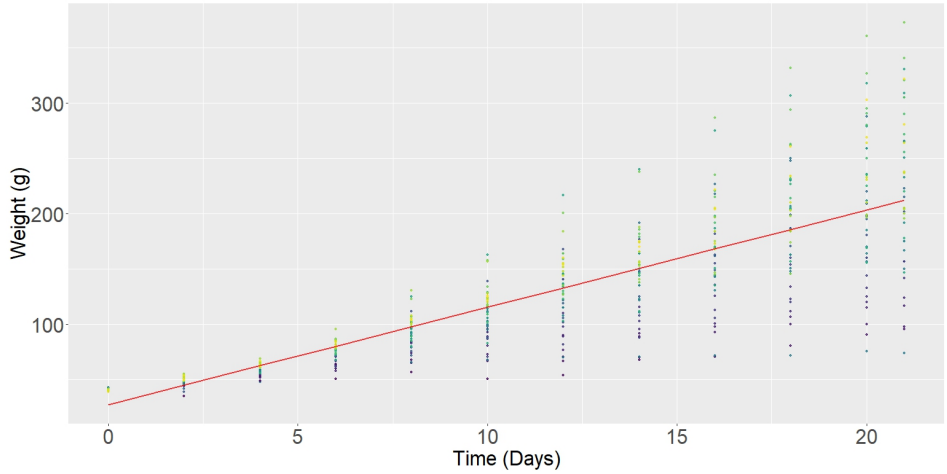


Figure 3.1: A Simple Linear Regression Model Fitted to `ChickWeight`

3.1 Building The Framework

We will build our framework in two stages before introducing the general model that is widely known as the linear mixed model. Verbeke and Molenberghs (2000)([3], p20-21) introduce the structure of linear mixed effects models in two stages. Note that this two-stage method does not fit two different models to the data, it is simply a helpful method to explain the theory.

3.1.1 First Theoretical Stage

Denote Y_{ij} as the response for individual i at time t_{ij}, $i = 1, ..., N$ (since we have N individuals in our data) and $j = 1, ..., n_i$ (n_i being the number of measurements for individual i). Note here that n_i can vary for each individual since we do not assume that we measure each individual an equal number of times and/or not at the same time points. The first stage assumes that $\boldsymbol{Y}_i$ follows a simple linear regression model:

$$\boldsymbol{Y}_i = \boldsymbol{Z}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i \ . \tag{3.1}$$

Here $\boldsymbol{Z}_i$ is the matrix that models how our data moves over time, it being a $n_i \times q$ matrix of covariates. $\boldsymbol{\beta}_i$ is a q-dimensional vector consisting of unknown regression coefficients which are specific to the certain individual; we call these 'subject specific'. $\boldsymbol{\epsilon}_i$ is a vector of residuals. In our setup, we assume that all of our residuals, $\boldsymbol{\epsilon}_i$, are normally distributed as such:

$$\boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I}_{n_i}) \ . \tag{3.2}$$

Here $\boldsymbol{I}_{n_i}$ is the n_i-dimensional identity matrix.

3.1.2 Second Theoretical Stage

In the second stage, we will highlight a model which will aim to explain the variability between subjects and include random effects which explain these variabilities with respect to their subject specific regression coefficients. We have the following model:

$$\boldsymbol{\beta}_i = \boldsymbol{M}_i\boldsymbol{\beta} + \boldsymbol{u}_i \ , \tag{3.3}$$

where $\boldsymbol{M}_i$ is $q \times p$ matrix of known covariates and $\boldsymbol{\beta}$ is a p-dimensional vector of unknown fixed regression parameters. Here $\boldsymbol{u}_i$ are our random effects and are assumed to be independent and follow a normal distribution as such:

$$\boldsymbol{u}_i \sim N(\boldsymbol{0}, \tilde{\boldsymbol{Q}}) \ . \tag{3.4}$$

Let us now put these two stages together, putting our second stage model into our first stage. This gives

$$\begin{aligned} \boldsymbol{Y}_i &= \boldsymbol{Z}_i(\boldsymbol{M}_i\boldsymbol{\beta}_i + \boldsymbol{u}_i) + \boldsymbol{\epsilon}_i \\ \boldsymbol{Y}_i &= \boldsymbol{Z}_i\boldsymbol{M}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{\epsilon}_i \ . \end{aligned}$$

$\boldsymbol{\beta}_i$ contains our fixed effects and $\boldsymbol{u}_i$ contains the random effects. For ease of notation, let us denote $\boldsymbol{X}_i = \boldsymbol{Z}_i\boldsymbol{M}_i$. In this case we have introduced the model in such a way that $\boldsymbol{X}_i$ is a linear combination of $\boldsymbol{Z}_i$. This does not always need to be true, however introducing the model in such a way helps explain the idea behind LMEMs. In the case where $\boldsymbol{X}_i$ is a linear combination of $\boldsymbol{Z}_i$, the model is known as a linear growth curve model. Dropping the subscripts, we denote the general framework for the model to be:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon} \ . \tag{3.5}$$

Here, $\boldsymbol{Y}$ is an N-dimensional vector of all repeated measurements for each of the individuals, having n_i repetitions for the i-th individual. $\boldsymbol{\epsilon}$ is a N-dimensional vector of residuals.

$$\boldsymbol{Y} = (y_{11}, ..., y_{1n_1}, y_{21}, ..., y_{2n_2}, ..., y_{n1}, ..., y_{nn_n})^T \in \mathbb{R}^N$$
$$\boldsymbol{\epsilon} = (\epsilon_{11}, ..., \epsilon_{1n_1}, \epsilon_{21}, ..., \epsilon_{2n_2}, ..., \epsilon_{n1}, ..., \epsilon_{nn_n})^T \in \mathbb{R}^N$$

Here, $\boldsymbol{X}$ is an $N \times p$ matrix of known covariates and $\boldsymbol{\beta}$ is a p-dimensional vector of unknown fixed effects. $\boldsymbol{Z}$ is an $N \times nq$ matrix of known covariates and $\boldsymbol{u}$ is an nq-dimensional vector of random effects with $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n)^T$, and n being the number of subjects. In a linear mixed effect model, we assume that

$$\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_N) \quad \text{and} \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{Q}) \ ,$$

where $\boldsymbol{\epsilon}$ and $\boldsymbol{u}$ are independent and

$$\boldsymbol{Q} = Var(\boldsymbol{u}) = \begin{pmatrix} Var(\boldsymbol{u}_1) & & \\ & \ddots & \\ & & Var(\boldsymbol{u}_n) \end{pmatrix} \ .$$

If we now take a marginal approach, we can calculate the expectation and variance of the model (3.5). The expectation is of the form

$$\mathbb{E}(\boldsymbol{Y}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\mathbb{E}(\boldsymbol{u}) + \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{X}\boldsymbol{\beta} \ , \tag{3.6}$$

since it is assumed that $\boldsymbol{\epsilon}$ and $\boldsymbol{u}$ are normally distributed with zero mean. The variance is calculated as

$$\text{Var}(\boldsymbol{Y}) = \boldsymbol{Z}\text{Var}(\boldsymbol{u})\boldsymbol{Z}^T + \sigma^2 \boldsymbol{I}_N = ZQZ^T + \sigma^2 I_N \ ,$$

since we have said that $\boldsymbol{X}\boldsymbol{\beta}$ contain the fixed effects so will not contribute to the overall variance. The other two components come from simple matrix algebra. This gives the "known" marginal variance

$$\boldsymbol{\Sigma} = \text{Var}(\boldsymbol{Y}) = \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2 \boldsymbol{I}_N \ . \tag{3.7}$$

3.2 Applying the Model to the `ChickWeight` Data

If we model the `ChickWeight` data using a linear mixed effects model, we would want the time and the diet to be part of the fixed effects and subject specific random effects which aim to capture variations in the intercepts. Our model would have the equation

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 B_i + \beta_3 C_i + \beta_4 D_i + u_i + \epsilon_{ij} \ , \tag{3.8}$$

where W_{ij} is the weight of chick i at time point j and t_{ij} is the time at time point j. The variables B_i, C_i and D_i are indicator variables defined to be one if the chick is on Diet B, C or D respectively and zero otherwise. Diet A is included as the reference level and absorbed into the intercept. If we

put this into the form of the linear mixed effects model equation (3.5), assuming chick i is on Diet B and is recorded at n_i different points, we get

$$X_i = \begin{pmatrix} 1 & t_{i1} & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_{in_i} & 1 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{n_i \times 5} \qquad \beta_i = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} \in \mathbb{R}^{5 \times 1}$$

In this case, since each chick has one random effect which accounts for the subject specific intercept, the matrix Z_i is just the 1×1 identity matrix and u_i is the individual subject specific intercept for chick i. The random intercept model is plotted in Figure 3.2. Using the `lmer` function, from the `lme4` package [5], in `R`, we get that the linear mixed effects model with subject specific intercepts to model the weight of chicks, using the REML method (see Section 4.1.3) to compute the regression coefficients.

$$W_{ij} = 11.244 + 8.717 t_{ij} + 16.210 B_i + 36.543 C_i + 30.013 D_i + u_i + \epsilon_{ij}$$

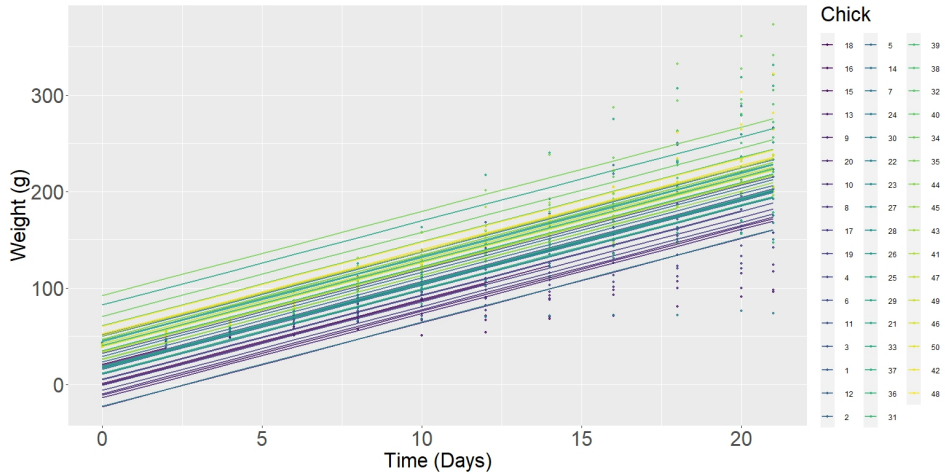


Figure 3.2: Random Intercept Model Applied to `ChickWeight`

What can be said about this? Although this is a good starting point, and much better than a single linear regression model, we can still see a lot of the data is not included in these linear models. We may prefer to also include a random slope to the model, to tackle the more 'extreme' data points in the later observations. If we include a random slope, the model will be of the form

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 B_i + \beta_3 C_i + \beta_4 D_i + u_{0i} + u_{1i} t_{ij} + \epsilon_i j \ , \tag{3.9}$$

where u_{0i} is the random intercept and u_{1i} is the random slope. The fitted model is shown in Figure 3.3 and the model is provided below:

$$W_{ij} = 26.356 + 8.444 t_{ij} + 2.839 B_i + 2.004 C_i + 9.255 D_i + u_{0i} + u_{1i} t_{ij} + \epsilon_{ij} \ .$$

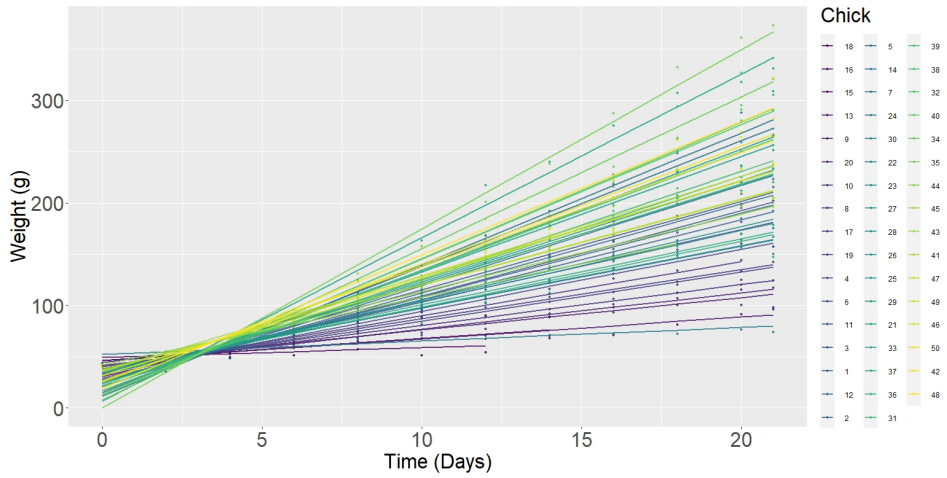


Figure 3.3: Random Slope and Intercept Model Applied to `ChickWeight`

One can argue that this model appears to fit better than the 'random intercept only' model as each linear model fitted seems to pick up more of the data. We can also observe the random effects of each model using the `sjPlot` package [6], which are calculated via the BLUP estimation method (see Section 4.2), in Figure 3.4.

Figure 3.4: Random Effects for the Random Slope Model (Left) and the Random Slope and Intercept Model (Right)

3.3 Model Selection Criteria

When it comes to higher dimensional data where there may be a large number of predictors, it may be 'best' to not include all of the predictors in our model. This may be due to the predictor having

an insignificant effect on our response variable or actually skewing our predictions. We will highlight criteria for which we can use to select our preferred model.

3.3.1 Akaike Information Criterion (AIC)

The AIC was first formally published in a 1974 paper by Hirotugu Akaike [7]. Simply stated, AIC is an estimator of prediction error and therefore provides the user with a quality measure for a statistical model for a specific dataset. AIC estimates the relative amount of information lost per model. The less information lost, the better the model. It deals with the trade off between the goodness-of-fit of the model and its simplicity. Akaike showed that based on his criterion, the model that should be chosen is the one that minimises

$$AIC = -2l_{\max} + 2k \ ,$$

where $l_{\max}$ is the maximum of the log-likelihood and k is the number of unknown parameters in the model. One can show that when all potential models use the same amount of observations, we can say

$$AIC = n\log(\hat{\sigma}^2) + 2k \ ,$$

where n is the number of individuals, $\hat{\sigma}^2 = n^{-1}||\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}_{LS}||^2$ is the variance and $\hat{\boldsymbol{\beta}}_{LS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ is the least squares estimate (Demidenko, 2013)([8] p.12). We prove this below.

One starts by using the normal distribution formula for $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. We have $\boldsymbol{Y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}_n)$.

$$f(\boldsymbol{Y}; \sigma^2, \boldsymbol{\beta}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}\exp\bigg\{ -\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\bigg\}$$

One then can calculate the log likelihood of $\boldsymbol{Y}$ as such:

$$l(\sigma^2, \boldsymbol{\beta}) = \log\bigg(\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}\exp\bigg\{ -\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\bigg\}\bigg) \ .$$

We can then simplify the expression into the form of

$$l(\sigma^2, \boldsymbol{\beta}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$$

At this stage we calculate the partial derivatives of our expression for the log likelihood with respect to σ^2 and $\boldsymbol{\beta}$ to obtain our $l_{\max}$.

$$\frac{\partial l}{\partial d\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$$

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = -\frac{1}{\sigma^2}\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}))$$

Setting both expressions equal to zero and solving for $\boldsymbol{\beta}$ and σ^2 gives the maximum likelihood estimates for $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$. We obtain

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \frac{1}{n}||\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||^2 \ .$$

Note that $\hat{\boldsymbol{\beta}}$ is the least squares estimate, $\hat{\boldsymbol{\beta}}_{\text{LS}}$. Plugging this back into the log-likelihood, we obtain our maximum log-likelihood as

$$\begin{aligned} l_{\max} &= -\frac{n}{2}\log(2\pi\hat{\sigma}^2) - \frac{n}{2} \\ &= -\frac{n}{2}\log(\hat{\sigma}^2) - \frac{n}{2}\log(2\pi) - \frac{n}{2} \ . \end{aligned}$$

When performing model selection, note that the 2^{nd} and 3^{rd} term in our expression do not matter, as we are comparing the AIC value, so these terms will lie in every AIC we compare. Hence, they can just be considered a shifting constant, which does not change the order of the AICs. Hence, we can finally obtain

$$\begin{aligned} AIC &= -2l_{\max} + 2k \\ &= n\log(\hat{\sigma}^2) + 2k \ , \end{aligned}$$

as required.

Although the AIC is one of the most used criteria in statistics, Demidenko (2013)([8] p.12) highlights a key weakness: the AIC underperforms in the case of multicollinearity. Multicollinearity occurs when one or more predictor variables are highly correlated with one another, therefore this can make regression coefficients unstable. For example, consider the situation of finding a model using a large set of covariates N. Assume that we are happy with $t < N$ covariates $\{x_1, ..., x_t\}$ being included in our model and we want to add a new variable q or p to our model. $\{x_1, ..., x_t, q\}$ and $\{x_1, ..., x_t, p\}$ yield very similar residual sum of squares (RSS) values, and consequently, similar $\hat{\sigma}^2$ values. Hence the difference in AIC is extremely small and not large enough to allow us to express a preference between models. However, if q is highly correlated with a covariate in the set $\{x_1, ..., x_t\}$ we will have multicollinearity and the model will have large standard errors and low t-statistics for the estimates. The model involving predictor p will not have this issue, however the AIC criterion will not pick up on this.

One can navigate around the drawback of AIC. Demidenko (2013)([8] p.12-13) uses the penalised log-likelihood to develop an alternate model selection criterion. We say without proof that the marginal penalised log-likelihood can be approximated as

$$l(w) \approx -k\log w + l(\mathbf{b}; \mathbf{y}) + g(w^{-1}\mathbf{b}) \ ,$$

where w is a positive scaling parameter, $\mathbf{b}$ is a k-dimensional random parameter vector and $g()$ is a penalty function. Assuming the prior distribution of the parameters is normal, one obtains

$$l \approx -\frac{k}{2}\log(w^2) + l(\mathbf{b}; \mathbf{y}) - \frac{1}{2w^2}||\mathbf{b}||^2 \ .$$

We achieve the maximum of l at $w^2 = \frac{||\mathbf{b}||^2}{k}$, so Demidenko (2013) ([8] p.12-13) defines the Healthy AIC as

$$HAIC = H - 2l_{\max} + 2k = H + AIC \ , \tag{3.10}$$

where $H = k(\log(\frac{||\hat{\mathbf{b}}_{ML}||^2}{k}) - 1)$.

3.3.2 Bayesian Information Criterion (BIC)

The Bayesian information criteria (BIC) arises in the Bayesian approach for model selection. We first define the BIC to be

$$BIC = -2l_{\max} + k\log(n) \ , \tag{3.11}$$

where n is the number of observations and k is the number of parameters chosen to be included in the model. We can see that generally the BIC has a heavier penalty term than the AIC since $\log(N) > 2$ for $N > 7$. Therefore, when using the BIC to select a model, it would favour a model with fewer variables than the AIC would. However, a key issue is faced when using the BIC to analyse linear mixed effects models. Shen and González (2021) [9] highlights that the calculation of the BIC is made assuming that the observations are independent and identically distributed (IID), however the theory behind LMEMs assumes that the measurements are not independent. Shen and González (2021) [9] argues for replacing the sample size n in (3.11) with the effective sample size, n_e, to give an improved BIC. The effective sample size of an estimator of a population mean is defined as

$$n_e = \frac{\sigma^2}{\text{Var}(\hat{\mu})} = \text{Population variance} \times \text{Precision of the Estimator} \ .$$

Hence, the 'improved BIC', introduced by Shen and González (2021) [9], is of the form

$$\text{BIC}_{n_e} = -2l_{\max} + k\log(n_e) \ . \tag{3.12}$$

We will include a final idea for an alternate BIC, also introduced by Shen and González (2021) ([9], p12). This is altering the original BIC defined at the start of the section by using the hybrid sample size for the candidate model. The hybrid approach BIC in [9] is defined as

$$\text{BIC}_h = -2l_{\max} + |\theta_R|\log N + |\theta_F|\log n \ , \tag{3.13}$$

where θ_R is the random components of the parameter vector θ in the model and $|\theta_R|$ is the dimension of θ_R. $|\theta_F|$ is the dimension of θ_F, our fixed components of the parameter vector. In this definition N is the number of subjects in the linear mixed effects model and n is the total number of observations across all subjects. This makes intuitive sense, instead of multiplying the entire number of parameters by the total number of observations, the hybrid approach considers the fixed and random parts of the model separately.

We will now find the BIC, improved BIC (3.12) and hybrid BIC (3.13) for four models which we have fitted to the `ChickWeight` dataset. The four models we fitted are detailed below.

Model 1: Weight Modelled with Fixed Time Effects and a Random Intercept

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + u_{i0} + \epsilon_{ij}$$

Model 2: Weight Modelled with Fixed Time and Diet Effects and a Random Intercept

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 B_i + \beta_3 C_i + \beta_4 D_i + u_{i0} + \epsilon_{ij}$$

Model 3: Weight Modelled with Fixed Time Effects and a Random Slope and Intercept

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + u_{i0} + u_{i1} t_{ij} + \epsilon_{ij}$$

Model 4: Weight Modelled with Fixed Time and Diet Effects and a Random Slope and Intercept

$$W_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 B_i + \beta_3 C_i + \beta_4 D_i + u_{i0} + u_{i1} t_{ij} + \epsilon_{ij}$$

We created functions in `R` to calculate the improved BIC (3.12) and hybrid BIC (3.13) and calculated the statistics for each model. The statistics are displayed in Table 3.1.

	BIC	Improved BIC	Hybrid BIC
Model 1	5646.301	5644.822	5625.605
Model 2	5628.46	5607.036	5629.986
Model 3	4865.636	4846.700	4862.226
Model 4	4860.912	4841.654	4857.559

Table 3.1: BIC Criteria for Models 1, 2, 3 and 4

We can see that all criteria select the fourth model, since the different BIC values are smallest, which is what we would expect. This model is the most robust since we are considering the different diets and including a random slope as well as intercept which acknowledges the different growth rates of individuals.

4 Analysis of the Fixed Effects and Random Effects

4.1 Inference for the Fixed Effects

Recall the framework set up in Section 3.1:

$$\boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{Q})$$
$$\boldsymbol{Y}|\boldsymbol{u} \sim N(\boldsymbol{X\beta} + \boldsymbol{Zu}, \sigma^2 \boldsymbol{I}_N)$$
$$\boldsymbol{Y} \sim N(\boldsymbol{X\beta}, \boldsymbol{ZQZ}^T + \sigma^2 \boldsymbol{I}_N)$$

We want to be able to estimate our fixed effect parameters vector $\boldsymbol{\beta}$ so we can learn more and make inferences about the model. Unfortunately, we cannot always use the same method to estimate $\boldsymbol{\beta}$. Our choice is entirely down to how much we know about the variance components, $\boldsymbol{Q}$ and σ^2. We will call the set of these variance parameters $\boldsymbol{\gamma} = \{\boldsymbol{Q}, \sigma^2\}$.

4.1.1 Variance Components Are Fully Known

Let us consider the first case, where $\boldsymbol{\gamma}$ is fully known, hence $\boldsymbol{\Sigma}$ is fully known.

Theorem 4.1 - Generalised Least Squares (GLS) Estimate for β
Einbeck (2023) [10] states that the estimate for the vector of fixed effects $\boldsymbol{\beta}$ when $\boldsymbol{\gamma}$ is fully known is

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} \ . \tag{4.1}$$

Proof. When estimating our parameter, we can use the maximum likelihood estimator (MLE) to provide us with an answer. To do this we first must calculate the likelihood of $\boldsymbol{Y}$. Recall that

$$\boldsymbol{Y} \sim N(\boldsymbol{X\beta}, \boldsymbol{ZQZ}^T + \sigma^2 \boldsymbol{I}_N) \ .$$

We will use the multivariate normal distribution formula to aid our proof. We can say that

$$f(\boldsymbol{Y}) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\boldsymbol{\Sigma}^{\frac{1}{2}}} \exp\bigg(-\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X\beta})\bigg) \ . \tag{4.2}$$

We will now determine the log-likelihood up to a constant. We choose the log-likelihood here since it deals nicely with our rather tricky exponential expression. We get

$$l(\boldsymbol{Y}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log(\boldsymbol{\Sigma}) - \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X\beta}) \ . \tag{4.3}$$

We now must differentiate $l(\boldsymbol{Y})$ with respect to $\boldsymbol{\beta}$ and set the expression to 0. It is clear to see that our first two components will disappear, so before we differentiate, we will expand the matrix algebra.

$$\begin{aligned}(\boldsymbol{Y} - \boldsymbol{X\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X\beta}) &= (\boldsymbol{Y}^T - \boldsymbol{\beta}^T\boldsymbol{X}^T)\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X\beta}) \\ &= (\boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})(\boldsymbol{Y} - \boldsymbol{X\beta}) \\ &= \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X\beta} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X\beta}\end{aligned}$$

Now we have expanded our expression, we can differentiate with respect to $\boldsymbol{\beta}$:

$$\frac{\partial l(\boldsymbol{Y})}{\partial \boldsymbol{\beta}} = -\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X} + 2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X} \ .$$

We now argue that $\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} = \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}$. We have this result due to symmetry of the problem. We now have

$$\frac{\partial l(\boldsymbol{Y})}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\boldsymbol{\beta} \ ,$$

which we can set equal to 0 and rearrange to obtain $\hat{\boldsymbol{\beta}}$:

$$\begin{aligned} -2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} &= 0 \ , \\ \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} &= \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} \ , \\ \hat{\boldsymbol{\beta}}_{\text{GLS}} &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} \ . \end{aligned}$$

□

We can also show that this estimate is unbiased since, when calculating the expectation, we only rely on the fact that $\mathbb{E}(\boldsymbol{Y}) = \boldsymbol{X}\boldsymbol{\beta}$, and then see that the majority of the expression cancels out.

$$\begin{aligned} \mathbb{E}(\hat{\boldsymbol{\beta}}_{\text{GLS}}) &= \mathbb{E}((\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}) \\ &= \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\mathbb{E}(\boldsymbol{Y}) \\ &= \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \ . \end{aligned} \tag{4.4}$$

The variance of $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ can be calculated with ease:

$$\begin{aligned} \text{Var}(\hat{\boldsymbol{\beta}}_{\text{GLS}}) &= \text{Var}((\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}) \\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\text{Var}(\boldsymbol{Y})((\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})^T \\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}((\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})^T \\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}) \\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1} \ . \end{aligned} \tag{4.5}$$

Generally, one uses the generalised least squares estimator for $\boldsymbol{\beta}$ if $\boldsymbol{\gamma}$ is fully known. However, this is not always the case.

4.1.2 Variance Components Are Not Fully Known - Maximisation of the Likelihood

If $\boldsymbol{\gamma}$ is unknown, one approach is to estimate it through the maximisation of the likelihood, as done by Einbeck (2023) [10],

$$L^*(\boldsymbol{\beta}, \boldsymbol{\gamma}) \propto \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\bigg(-\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\bigg) \ . \tag{4.6}$$

Where we say that $\boldsymbol{\Sigma}$ is evaluated at $\boldsymbol{\gamma}$ since we do not fully know the variance components. We then use the $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ (4.1) but evaluate it with respect to the unknown variance components $\boldsymbol{\gamma}$. Hence, we have

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\gamma}) = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\boldsymbol{Y} \ . \tag{4.7}$$

One can then substitute in this estimate for $\boldsymbol{\beta}$ into $L^*(\boldsymbol{\beta}, \boldsymbol{\gamma})$, as done by Einbeck (2023) [10], giving $L(\boldsymbol{\gamma}) = L^*(\hat{\boldsymbol{\beta}}(\boldsymbol{\gamma}), \boldsymbol{\gamma})$ which no longer depends directly on $\boldsymbol{\beta}$. Maximising $L(\boldsymbol{\gamma})$ with respect to $\boldsymbol{\gamma}$ yields

$$\hat{\boldsymbol{\gamma}}_{ML} = \arg\max_{\boldsymbol{\gamma}} L(\boldsymbol{\gamma}) \ . \tag{4.8}$$

This is a good start in the process of estimation. However, Einbeck (2023) [10] highlights that "$\hat{\boldsymbol{\gamma}}_{ML}$ is biased for $\boldsymbol{\gamma}$, due to a "loss" in degrees of freedom" when estimating $\boldsymbol{\beta}$. Hence, we seek to find a method that avoids this issue.

4.1.3 The Restricted Maximisation Likelihood Estimation (REML) Method

In this section we will discuss the Restricted Maximisation Likelihood Estimation (REML) method. The theory introduced is referenced from Einbeck (2023) [10], LaMotte (2007) [11] and Verbeke and Molenberghs (2000) ([3] , p44-47). The issue with the theory discussed in Section 4.1.2 was the dependence on $\boldsymbol{\beta}$. The fundamental idea of REML is to multiply the model $\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Zu} + \boldsymbol{\epsilon}$ by an $n \times (n-p)$ dimensional matrix $\boldsymbol{A}$ which is orthogonal to $\boldsymbol{X}$, so that one has $\boldsymbol{A^TX} = 0$. So, one has

$$\begin{aligned}\boldsymbol{A}^T\boldsymbol{Y} &= \boldsymbol{A}^T\boldsymbol{X\beta} + \boldsymbol{A}^T\boldsymbol{Zu} + \boldsymbol{A}^T\boldsymbol{\epsilon} \\ &= \boldsymbol{A}^T\boldsymbol{Zu} + \boldsymbol{A}^T\boldsymbol{\epsilon} \ .\end{aligned}$$

This means that one can find a likelihood of $\boldsymbol{A}^T\boldsymbol{Y}$ that does not depend on $\boldsymbol{\beta}$. Using the property of multivariate normal distributions, we can say that the distribution of $\boldsymbol{A}^T\boldsymbol{Y}$ is as follows:

$$\boldsymbol{A}^T\boldsymbol{Y} \sim N(\boldsymbol{0}, \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}) \ .$$

So it is straightforward to calculate the REML likelihood as

$$\begin{aligned}L_{\text{REML}} &= \frac{1}{(2\pi)^{n-p}}\frac{1}{|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}|^{\frac{1}{2}}}\exp\bigg(-\frac{1}{2}(\boldsymbol{A}^T\boldsymbol{Y}-\boldsymbol{0})^T(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{Y}-\boldsymbol{0})\bigg) \\ &= \frac{1}{(2\pi)^{n-p}}\frac{1}{|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}|^{\frac{1}{2}}}\exp\bigg(-\frac{1}{2}\boldsymbol{Y}^T\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{Y}\bigg) \ .\end{aligned} \tag{4.9}$$

We can rewrite the L_{REML} in an expression that does not contain $\boldsymbol{A}$. This is summarised in Theorem 4.2 below.

Theorem 4.2
One can rewrite the REML likelihood given in (4.9) independent of $\boldsymbol{A}$, in the following form:

$$L_{REML} \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}}|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-\frac{1}{2}}\exp\bigg(-\frac{1}{2}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\bigg) \ . \tag{4.10}$$

Which we can then say

$$L_{REML} \propto |\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-\frac{1}{2}}L(\boldsymbol{\gamma})$$

Proof. We will heavily refer to the work of LaMotte (2007) [11] for the proof of Theorem 4.2, adding in further explanation and working where necessary. First, a few assumptions are required. We assume that $\boldsymbol{\Sigma}$ is an $n \times n$ positive-definite matrix. The first step of the proof is showing that

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1} + \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T \ .$$

It is noted that

$$\begin{aligned}
(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^{-1}\boldsymbol{\Sigma}^{-1}((\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T)^{-1} &= [(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})]^{-1} \\
&= \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\boldsymbol{A} \\ \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{X} & \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} \end{pmatrix}^{-1} \\
&= \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} \end{pmatrix}^{-1} \\
&= \begin{pmatrix} (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1} \end{pmatrix} \ .
\end{aligned}$$

LaMotte (2007) [11] then rewrites $\boldsymbol{\Sigma}^{-1}$ to include the expression above. Essentially, LaMotte adds the inverse of the outside matrices so they cancel out. One can see that

$$\begin{aligned}
\boldsymbol{\Sigma}^{-1} &= (\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^{-1}\boldsymbol{\Sigma}^{-1}((\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T)^{-1}(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T \\
&= (\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A}) \begin{pmatrix} (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1} \end{pmatrix} (\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T \\
&= (\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}, \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1})(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}, \boldsymbol{A})^T \\
&= \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1} + \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T \ .
\end{aligned}$$

The expression above can be rearranged as

$$\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1} \ , \tag{4.11}$$

and one can rewrite the following as such:

$$\begin{aligned}
\boldsymbol{Y}^T\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{Y} &= \boldsymbol{Y}^T\Big(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\Big)\boldsymbol{Y} \\
&= \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} \\
&= \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} \ ,
\end{aligned}$$

where the GLS estimate of $\boldsymbol{\beta}$ is used. LaMotte (2007) [11] then claims that the expression above is equal to $(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$ without proof. We show equivalence here:

$$\begin{aligned}
(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) &= \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} \\
&= \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Y}) \ .
\end{aligned}$$

Now note that

$$\begin{aligned}\hat{\beta}^T \boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Y}) &= \hat{\beta}^T \boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}) \\ &= \hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{X}\boldsymbol{X}^{-1}\boldsymbol{\Sigma}(\boldsymbol{X}^T)^{-1}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{Y} - \boldsymbol{Y}) \\ &= \hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{Y}) \\ &= \boldsymbol{0} \ .\end{aligned}$$

Hence one has

$$\boldsymbol{Y}^T \boldsymbol{A}(\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A})^{-1}\boldsymbol{A}^T \boldsymbol{Y} = (\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) \ . \tag{4.12}$$

The final step of the proof is to justify the relation

$$|\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A}| = C|\boldsymbol{\Sigma}||\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X}| \ ,$$

where C is a constant. To do this, it is satisfactory to show that the derivatives of the logarithms of $|\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A}|$ and $|\boldsymbol{\Sigma}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X}|$ are equal. For the ease of notation, allow

$$\boldsymbol{W} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1} \ .$$

To calculate the derivatives, LaMotte (2007) [11] relies on the Jacobi's formula [12], which states that, for any square matrix $\boldsymbol{P}$, one has

$$\frac{\partial}{\partial \boldsymbol{\theta}}|\boldsymbol{P}| = |\boldsymbol{P}|\mathrm{tr}(\boldsymbol{P}^{-1}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}}) \ , \tag{4.13}$$

where tr( ) denotes the trace of a matrix, the sum of the diagonal entries of the matrix. So when tackling the derivative of the logarithm of the determinant of a matrix $\boldsymbol{P}$, the chain rule can be applied as such:

$$\frac{\partial}{\partial \boldsymbol{\theta}}\log|\boldsymbol{P}| = \frac{1}{|\boldsymbol{P}|}|\boldsymbol{P}|\mathrm{tr}(\boldsymbol{P}^{-1}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}}) = \mathrm{tr}(\boldsymbol{P}^{-1}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}}) \ .$$

Denote $\boldsymbol{\Sigma_i} = \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\theta}_i}$. Now, calculating the derivatives, we have

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_i}\log|\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A}| &= \mathrm{tr}\big[(\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A})^{-1}\boldsymbol{A}^T \boldsymbol{\Sigma}_i \boldsymbol{A}\big] \\ &= \mathrm{tr}\big[\boldsymbol{A}(\boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A})^{-1}\boldsymbol{A}^T \boldsymbol{\Sigma}_i\big] \\ &= \mathrm{tr}(\boldsymbol{W}\boldsymbol{\Sigma}_i) \ .\end{aligned}$$

We can rearrange the expression in the second line as so since $\mathrm{tr}(\boldsymbol{AB}) = \mathrm{tr}(\boldsymbol{BA})$. The second derivative requires slightly more work. First, note that

$$\log|\boldsymbol{\Sigma}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X}| = \log|\boldsymbol{\Sigma}||\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X}| = \log|\boldsymbol{\Sigma}| + \log|\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{X}| \ .$$

Hence,

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_i}\log|\boldsymbol{\Sigma}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}| &= \frac{\partial}{\partial \boldsymbol{\theta}_i}\big(\log|\boldsymbol{\Sigma}| + \log|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|\big)\\ &= \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) - \mathrm{tr}((\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i\boldsymbol{\Sigma}^{-1}\boldsymbol{X})\\ &= \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) - \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i)\\ &= \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) - \mathrm{tr}((\boldsymbol{\Sigma}^{-1} - \boldsymbol{W})\boldsymbol{\Sigma}_i)\\ &= \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) - \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) + \mathrm{tr}(\boldsymbol{W}\boldsymbol{\Sigma}_i)\\ &= \mathrm{tr}(\boldsymbol{W}\boldsymbol{\Sigma}_i)\ .\end{aligned}$$

Hence, the proof is complete. Equivalence of the derivatives of the logarithms implies

$$|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}| = C|\boldsymbol{\Sigma}||\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|\ ,$$

which in turn implies that the REML likelihood can be written without $\boldsymbol{A}$. Using this result, one has

$$L_{REML} \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}}|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-\frac{1}{2}}\exp\bigg(-\frac{1}{2}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\bigg)\ .$$

□

We follow Theorem 4.2 with an immediate corollary, which states that we can find a closed form expression of the REML likelihood which does not contain $\boldsymbol{A}$. This was introduced by Harville (1974) [13].

Corollary 4.3
If $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, then the REML likelihood (4.9) can be expressed in a closed form as

$$L_{REML} = (2\pi)^{-\frac{(n-p)}{2}}|\boldsymbol{X}^T\boldsymbol{X}|^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-\frac{1}{2}}\exp\bigg(-\frac{1}{2}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\bigg)\ . \quad (4.14)$$

Proof. LaMotte (2007) [11] proves this by proving the following. Assuming $\boldsymbol{\Sigma}$ is an $n \times n$ symmetric, positive definite matrix and $(\boldsymbol{X}, \boldsymbol{A})$ is an $n \times n$ matrix with $\boldsymbol{A}^T\boldsymbol{X} = \boldsymbol{0}$, then

$$|\boldsymbol{\Sigma}| = \frac{|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{X}|}{|\boldsymbol{A}^T\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|}\ .$$

LaMotte (2007) [11] notes that

$$\begin{aligned}|\boldsymbol{A}^T\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{X}||\boldsymbol{\Sigma}| &= \left|\begin{pmatrix}\boldsymbol{X}^T\\ \boldsymbol{A}^T\end{pmatrix}\boldsymbol{\Sigma}(\boldsymbol{X},\boldsymbol{A})\right|\\ &= \begin{vmatrix}\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{A}\\ \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{X} & \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}\end{vmatrix}\ ,\end{aligned}$$

by expanding the expression in the first line. LaMotte (2007) [11] then makes use of the calculation of the determinant of a 2×2 matrix. This makes the determinant of the matrix

$$\begin{aligned}\begin{vmatrix} \boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{A} \\ \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{X} & \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} \end{vmatrix} &= |\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X}||\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}| - |\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{X}| \\ &= |\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{X}| \\ &= |\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T[\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}]\boldsymbol{X}| \ .\end{aligned}$$

LaMotte (2007) [11] notes that using the equivalence previously proved (4.11) then

$$\begin{aligned}\boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma} &= \boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})\boldsymbol{\Sigma} \\ &= \boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma} \\ &= \boldsymbol{\Sigma} - \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T \ .\end{aligned}$$

Hence, $\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T$. So, one has the following:

$$\begin{aligned}|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T[\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}]\boldsymbol{X}| &= |\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}| \\ &= |\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T||\boldsymbol{X}^T||\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-1} \\ &= \frac{|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{X}|^2}{|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|} \ .\end{aligned}$$

Hence, using this, LaMotte (2007) [11] proves

$$|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}| = \frac{|\boldsymbol{\Sigma}||\boldsymbol{A}^T\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|}{|\boldsymbol{X}^T\boldsymbol{X}|} \ ,$$

which can be substituted into the REML likelihood equation (4.9) as follows:

$$\begin{aligned}L_{REML} &= \frac{1}{(2\pi)^{n-p}}\left[\frac{|\boldsymbol{\Sigma}||\boldsymbol{A}^T\boldsymbol{A}||\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|}{|\boldsymbol{X}^T\boldsymbol{X}|}\right]^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})\right) \\ &= \frac{1}{(2\pi)^{n-p}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}|\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}|^{-\frac{1}{2}}|\boldsymbol{X}^T\boldsymbol{X}|^{\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})\right) \ ,\end{aligned}$$

since $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$. □

We are interested in the REML estimation method as this is used in `R` to calculate the fixed effects coefficients in functions such as `lmer` and `lme` which we have already used in Chapter 3 to create the mixed effects models for the `ChickWeight` data.

4.1.4 Minimum Norm Quadratic Unbiased Estimator (MINQUE) for σ^2

When we perform maximum likelihood estimation for the LME model, we must assume that the distributions of the error term and the random effects are both normal. Demidenko (2013) [8] claims this

assumption means that 'the MLE may not be robust to the distribution specification.' A potential solution to this issue is quadratic estimation of the variance parameters since it does not require distribution assumptions and produces distribution-free estimates for the parameters. In this section we will discuss the Minimum Norm Quadratic Unbiased Estimator (MINQUE), as highlighted by Demidenko (2013). When choosing within the class of quadratic unbiased estimators, Rao (1973) [14] suggests minimising the norm of the variance matrix since minimising the variance directly involves third and fourth moments, which would lead to complications. We will follow the development of the theory as done by Demidenko (2013) ([8] p.157-166) as MINQUE theory can be applied to linear regression models and easily extended to linear mixed effects models.

Demidenko (2013) ([8] p.158) first applies MINQUE theory to the standard linear regression model. One has

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \qquad \text{with} \quad \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}, \quad \text{Cov}(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{I}$$

Here $\boldsymbol{y}$ is a $n \times 1$ vector of observations of the response variable, $\boldsymbol{X}$ is a $n \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown parameters and $\boldsymbol{\epsilon}$ is a $n \times 1$ error term vector. We do not assume that $\boldsymbol{X}$ is of full rank. We want to find a quadratic estimator for σ^2 in the linear regression model. Demidenko (2013) ([8] p.158) says if one has bmA, an $n \times n$ matrix, then a quadratic estimator for σ^2 is defined as

$$\hat{\sigma}^2 = \boldsymbol{y}^T \boldsymbol{A} \boldsymbol{y} \ .$$

Demidenko (2013) ([8] p.158) makes a couple of assumptions without loss of generality: $\boldsymbol{A}$ is a symmetric matrix and, to ensure $\hat{\sigma}^2$ is always non-negative, $\boldsymbol{A}$ is non-negative. Demidenko (2013) ([8] p.158) then introduces the formula $\mathbb{E}(\boldsymbol{\epsilon}^T \boldsymbol{A} \boldsymbol{\epsilon} = \text{tr}(\boldsymbol{A}\text{Cov}(\boldsymbol{\epsilon}))$ without proof. The proof is briefly included below. We can say that

$$\boldsymbol{\epsilon}^T \boldsymbol{A} \boldsymbol{\epsilon} = \sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} \epsilon_i \epsilon_j \ .$$

Hence, taking expectations we have

$$\mathbb{E}(\boldsymbol{\epsilon}^T \boldsymbol{A} \boldsymbol{\epsilon}) = \sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} \mathbb{E}(\epsilon_i \epsilon_j) \ .$$

Howeve, we also know that $\text{Cov}(\epsilon_i, \epsilon_j) = \mathbb{E}(\epsilon_i \epsilon_j) - \mathbb{E}(\epsilon_i)\mathbb{E}(\epsilon_j)$ and we also know the error terms have zero mean. Hence,

$$\begin{aligned} \mathbb{E}(\boldsymbol{\epsilon}^T \boldsymbol{A} \boldsymbol{\epsilon}) &= \sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} \text{Cov}(\epsilon_i, \epsilon_j) \\ &= \text{tr}(\boldsymbol{A}\text{Cov}(\boldsymbol{\epsilon})) \ . \end{aligned}$$

Since $\text{Cov}(\boldsymbol{\epsilon})$ only has entries on the diagonal, the sum above is equal to calculating the trace of the product of the matrices since we are only getting non zero entries on the diagonal. Hence, we have proved the formula. Using this in the calculation of the expectation of $\hat{\sigma}^2$ yields the following result:

$$\begin{aligned}\mathbb{E}(\hat{\sigma}^2) &= \mathbb{E}(\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y}) = \mathbb{E}\{(\boldsymbol{\epsilon}^T + \beta^T\boldsymbol{X}^T)\boldsymbol{A}(\boldsymbol{X}\beta + \boldsymbol{\epsilon})\} \\ &= \mathbb{E}(\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T\boldsymbol{A}\boldsymbol{\epsilon}) \\ &= \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \mathrm{tr}(\boldsymbol{A}\mathrm{Cov}(\boldsymbol{\epsilon})) = \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \sigma^2\mathrm{tr}(\boldsymbol{A}) \ .\end{aligned}$$

To ensure that the estimator is unbiased, one must have $\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} = 0$ and $\mathrm{tr}(\boldsymbol{A}) = 1$. The aim of MINQUE is to find the quadratic unbiased estimator that minimises the norm of $\boldsymbol{A}$. One must minimise

$$\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^T) \ , \quad \text{given} \quad \boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} = 0 \quad \text{and} \quad \mathrm{tr}(\boldsymbol{A}) = 1 \ .$$

Demidenko (2013) ([8] p.159) solves this problem by using a Lagrange multiplier. One can use a $m \times m$ Lagrange multiplier matrix $\boldsymbol{L_1}$ and a scalar value l_2, so the Lagrange function can be written as

$$\mathcal{L}(\boldsymbol{A}, \boldsymbol{L}_1, l_2) = \frac{1}{2}\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^T) + \mathrm{tr}(\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{L}_1{}^T) + (1 - \mathrm{tr}(\boldsymbol{A}))l_2 \ .$$

The Lagrange function must then be differentiated with respect to $\boldsymbol{A}$. Demidenko (2013) ([8] p.159) makes use of the following three formulae:

$$\frac{\partial \mathrm{tr}(\boldsymbol{A})}{\partial \boldsymbol{A}} = \boldsymbol{I}, \qquad \frac{\partial \mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^T)}{\partial \boldsymbol{A}} = 2\boldsymbol{A}, \qquad \frac{\partial \mathrm{tr}(\boldsymbol{C}\boldsymbol{A}\boldsymbol{B})}{\partial \boldsymbol{A}} = \boldsymbol{C}^T\boldsymbol{B}^T \ .$$

One then must differentiate the Lagrange function and set it equal to 0, such that

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{X}\boldsymbol{L_1}\boldsymbol{X}^T - l_2\boldsymbol{I} = 0 \ .$$

We can find L_1 by rearranging and using the first condition:

$$\begin{aligned}\boldsymbol{A} + \boldsymbol{X}\boldsymbol{L_1}\boldsymbol{X}^T &= l_2\boldsymbol{I} \\ \boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{L_1}\boldsymbol{X}^T\boldsymbol{X} &= l_2\boldsymbol{X}^T\boldsymbol{X} \\ \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{L_1}\boldsymbol{X}^T\boldsymbol{X} &= l_2\boldsymbol{X}^T\boldsymbol{X} \\ \boldsymbol{L_1} &= l_2(\boldsymbol{X}^T\boldsymbol{X})^\dagger \ ,\end{aligned}$$

where $(\boldsymbol{X}^T\boldsymbol{X})^\dagger$ denotes the pseudo inverse of the product of the matrices, which is a generalisation of the inverse matrices which are not necessarily square or invertible. Therefore, one obtains

$$\boldsymbol{A} = l_2(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^\dagger\boldsymbol{X}^T) = l_2(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger) \ .$$

To satisfy $\mathrm{tr}(\boldsymbol{A}) = 1$, Demidenko (2013) ([8] p.159) notes that $(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger)(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger) = \boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger - \boldsymbol{X}\boldsymbol{X}^\dagger + \boldsymbol{X}\boldsymbol{X}^\dagger = \boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger$ hence $(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger)$ is an idempotent matrix. This means its rank and trace are equal. So we can set

$$l_2 = \frac{1}{\mathrm{rank}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger)}$$

Hence

$$\mathrm{tr}(\boldsymbol{A}) = \frac{1}{\mathrm{rank}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger)}\mathrm{tr}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger) = 1$$

Hence one has the following MINQUE for $\hat{\sigma}^2$ in a standard linear regression as

$$\hat{\sigma}^2 = \frac{1}{n - rank(\boldsymbol{X})} \boldsymbol{y}^T (\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\dagger)\boldsymbol{y}$$

Using this theory developed in the simple case for a standard linear regression model, the MINQUE for σ^2 in a linear mixed effects model can be found by first considering the case of a mixed effects model and then making slight adjustments. Consider the mixed effects model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\nu}, \quad \mathbb{E}(\boldsymbol{\nu}) = \boldsymbol{0}, \quad \text{Cov}(\boldsymbol{\nu}) = \sigma^2 \boldsymbol{I} + \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T$$

Then

$$\begin{aligned}
\mathbb{E}(\hat{\sigma}^2) &= \mathbb{E}\big((\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\nu})^T \boldsymbol{A}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\nu})\big) \\
&= \boldsymbol{\beta}^T \boldsymbol{X}^T \boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \mathbb{E}(\boldsymbol{\nu}^T \boldsymbol{A}\boldsymbol{\nu}) \\
&= \boldsymbol{\beta}^T \boldsymbol{X}^T \boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \text{tr}(A\text{Cov}(\boldsymbol{\nu})) \\
&= \boldsymbol{\beta}^T \boldsymbol{X}^T \boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} + \sigma^2 \text{tr}(\boldsymbol{A}) + \text{tr}(\boldsymbol{Q}\boldsymbol{Z}^T \boldsymbol{A}\boldsymbol{Z})
\end{aligned}$$

Again, the aim is to make $\hat{\sigma}^2$ unbiased. For this, one needs $\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} = 0$, $\boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{Z} = 0$ and $\text{tr}(\boldsymbol{A}) = 1$. Demidenko (2013) ([8], p.160) combines $\boldsymbol{X}$ and $\boldsymbol{Z}$ into a single concatenated matrix $\boldsymbol{W} = [\boldsymbol{X}; \boldsymbol{W}]$ so the first two conditions can become $\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} = 0$. We omit the proof of equivalence between these conditions. So one must minimise $\text{tr}(\boldsymbol{A}\boldsymbol{A}^T)$ under the restrictions $\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} = 0$ and $\text{tr}(\boldsymbol{A}) = 1$. This is exactly the same problem that was faced in the standard linear regression case. Hence,

$$\hat{\sigma}^2 = \frac{1}{n - \text{rank}(\boldsymbol{W})} \boldsymbol{y}^T (\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger)\boldsymbol{y} \ .$$

Finally, the MINQUE theory can be applied to the LME model as in Equation 3.5. The dimension of $\boldsymbol{A}$ is $N_T \times N_T$ where N_T is the total number of observations $\sum_{i=1}^N n_i$ where n_i is the number of observations for the i^{th} individual. For the matrix $\boldsymbol{W}$, Demidenko (2013) ([8], p.160) has

$$\boldsymbol{y}^T (\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger)\boldsymbol{y} = \min_{\boldsymbol{\beta}, \boldsymbol{b_1}, \ldots, \boldsymbol{b_N}} \sum_{i=1}^N ||\boldsymbol{y}_i - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{b_i}||^2 = S_{\min} \ .$$

So one has the MINQUE estimator for $\hat{\sigma}^2$ in the linear mixed effects model as

$$\hat{\sigma}^2_{\text{MINQUE}} = \frac{S_{\min}}{N_T - \text{rank}(\boldsymbol{W})} \ .$$

4.2 Prediction of the Random Effects

Whilst it is most common just to estimate the fixed effects and variance components, one may find it useful to also estimate the random effects, $\boldsymbol{u}_i$. This will tell us how each subject differs from the population average. There are multiple methods which maybe used to do this. This section will follow a proof provided by Einbeck (2023) [10]. We first consider the Bayesian approach. We will estimate $\boldsymbol{u}_i$ by taking its posterior mean. From Bayes theorem, we know that

$$f(\boldsymbol{u}_i|\boldsymbol{y}_i) = \frac{f(\boldsymbol{y}_i|\boldsymbol{u}_i) f(\boldsymbol{u}_i)}{\int f(\boldsymbol{y}_i|\boldsymbol{u}_i) f(\boldsymbol{u}_i) d\boldsymbol{u}_i} \ ,$$

and we are interested in calculating

$$\hat{u}_i(\boldsymbol{\theta}) = E(\boldsymbol{u}_i|\boldsymbol{Y}_i = \boldsymbol{y}_i) \ .$$

To do this, a general result for multivariate normal distributions is used. If one has

$$\begin{pmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{pmatrix} \sim N\Bigg(\begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix} \Bigg) \ ,$$

then

$$\boldsymbol{y}_1|\boldsymbol{y}_2 \sim N(\boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{y}_2 - \boldsymbol{\mu}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \ .$$

It is clear that with the correct implementation that this will provide our posterior distribution for $\boldsymbol{u}_i$. Denote $\boldsymbol{C} = \mathrm{Cov}(\boldsymbol{u}, \boldsymbol{Y})$. Since

$$\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{Y}\end{pmatrix} \sim N\Bigg(\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{X\beta}\end{pmatrix}, \begin{pmatrix}\boldsymbol{Q} & \boldsymbol{C}\\ \boldsymbol{C}^T & \boldsymbol{\Sigma}\end{pmatrix} \Bigg) \ ,$$

then

$$\boldsymbol{u}|\boldsymbol{Y} \sim N(\boldsymbol{C\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X\beta}), \boldsymbol{Q} - \boldsymbol{C\Sigma}^{-1}\boldsymbol{C}^T) \ .$$

All there is left to do is to calculate $\mathrm{Cov}(\boldsymbol{u}, \boldsymbol{Y})$. Using the total law of covariance, for any random vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, one has

$$\mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b}) = \mathbb{E}(\mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b})|\boldsymbol{c}) + \mathrm{Cov}(\mathbb{E}(\boldsymbol{a}|\boldsymbol{c}), \mathbb{E}(\boldsymbol{b}|\boldsymbol{c})) \ .$$

Setting $\boldsymbol{c} = \boldsymbol{b}$,

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b}) &= \mathbb{E}(\mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b})|\boldsymbol{b}) + \mathrm{Cov}(\mathbb{E}(\boldsymbol{a}|\boldsymbol{b}), \mathbb{E}(\boldsymbol{b}|\boldsymbol{b}))\\ &= \mathrm{Cov}(\mathbb{E}(\boldsymbol{a}|\boldsymbol{b}), \boldsymbol{b}) \ .\end{aligned}$$

So $\mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{u})$ is calculated as

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{u}) &= \mathrm{Cov}(\mathbb{E}(\boldsymbol{Y}, \boldsymbol{u})|\boldsymbol{u})\\ &= \mathrm{Cov}(\boldsymbol{X\beta} + \boldsymbol{Zu}, \boldsymbol{u})\\ &= \mathrm{Cov}(\boldsymbol{X\beta}, \boldsymbol{u}) + \mathrm{Cov}(\boldsymbol{Zu}, \boldsymbol{u})\\ &= \boldsymbol{0} + \boldsymbol{Z}\mathrm{Cov}(\boldsymbol{u}, \boldsymbol{u})\\ &= \boldsymbol{ZQ} \ .\end{aligned}$$

Therefore $\boldsymbol{C} = \boldsymbol{QZ}^T$ since $\boldsymbol{C}^T = \mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{u})$ and $\boldsymbol{Q}$ is symmetric. Hence, one has the estimate for $\boldsymbol{u}$, which we have specified to be its posterior mean, as

$$\hat{\boldsymbol{u}}_i(\boldsymbol{\theta}) = \boldsymbol{QZ}^T\Sigma^{-1}(\boldsymbol{Y} - \boldsymbol{X}\beta) \ . \tag{4.15}$$

We can also find the expectation and the variance of the estimate of the random effects vector, assuming that $\boldsymbol{\beta}$ is the generalised least squares estimate for the vector of fixed effects $\hat{\boldsymbol{\beta}}$. Recall that $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\Sigma^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\Sigma^{-1}\boldsymbol{Y}$.

$$\begin{aligned}\mathbb{E}(\hat{\boldsymbol{u}}_i(\boldsymbol{\theta})) &= \mathbb{E}(\boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}))\\ &= \boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\mathbb{E}(\boldsymbol{Y})-\boldsymbol{X}\mathbb{E}(\boldsymbol{\beta}))\\ &= \boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{X}\boldsymbol{\beta})\\ &= \boldsymbol{0}\end{aligned}$$

$$\begin{aligned}\mathrm{Var}(\hat{\boldsymbol{u}}_i(\boldsymbol{\theta})) &= \mathrm{Var}(\boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}))\\ &= \boldsymbol{Q}\boldsymbol{Z}^T\mathrm{Var}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}))\boldsymbol{Z}\boldsymbol{Q}\\ &= \boldsymbol{Q}\boldsymbol{Z}^T(\boldsymbol{\Sigma}^{-1}\mathrm{Var}(\boldsymbol{Y})\boldsymbol{\Sigma}-1-\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\mathrm{Var}(\boldsymbol{\beta})\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})\boldsymbol{Z}\boldsymbol{Q}\\ &= \boldsymbol{Q}\boldsymbol{Z}^T(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}-\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})\boldsymbol{Z}\boldsymbol{Q}\\ &= \boldsymbol{Q}\boldsymbol{Z}^T(\boldsymbol{\Sigma}^{-1}-\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1})\boldsymbol{Z}\boldsymbol{Q}\end{aligned}$$

When we substitute $\hat{\boldsymbol{\beta}}$ into the $\hat{\boldsymbol{u}}_i(\theta)$ equation (4.15) like we did in the calculations of the expectation and variance, this gives us the BLUP (best linear unbiased predictor).

4.3 Criterion for Relevance of Random Effects

A question that may arise is: in the model, are the random effects relevant? We can rewrite $\boldsymbol{Q} = \sigma^2\boldsymbol{D}$ for ease of the notation. Demidenko (2013) ([8] p.74-75) claims that we can argue that the random effects are relevant in the model if the maximum likelihood estimate of $\boldsymbol{D}$, $\hat{\boldsymbol{D}}_{ML}$, is non-zero. Demidenko (2013) ([8], p.74) proves the following theorem which provides a criterion for $\hat{\boldsymbol{D}}_{ML} \neq \boldsymbol{0}$. We define the following before introducing the theorem. One has the ordinary least squares (OLS) estimate of $\boldsymbol{\beta}$ as such:

$$\boldsymbol{\beta}_{\mathrm{OLS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \ . \tag{4.16}$$

The OLS estimate is a special case of the GLS estimate for when $\boldsymbol{D} = \boldsymbol{0}$. The error vector $\hat{\boldsymbol{e}}$ is defined to be $\hat{\boldsymbol{e}}_i = \boldsymbol{Y}_i - \boldsymbol{X}_{\boldsymbol{i}}\boldsymbol{\beta}_{\mathrm{OLS}}$, with $\hat{\boldsymbol{e}}_i \in \mathbb{R}^{n_i\times 1}$ and the OLS variance is defined as such:

$$\sigma^2_{OLS} = \frac{\sum_i ||\hat{\boldsymbol{e}}_i||^2}{N_T}$$

The following theorem is introduced by Demidenko (2013) ([8], p.74):

Theorem 4.4 - Relevance of the Random Effects
If the matrix

$$\sum_i \boldsymbol{Z}_i^T\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^T\boldsymbol{Z}_i - \sigma^2{}_{\mathrm{OLS}}\sum_i \boldsymbol{Z}_i^T\boldsymbol{Z}_i \tag{4.17}$$

is non-zero and positive semidefinite, the maximum likelihood estimate of $\boldsymbol{D}$ is nonzero, and hence the random effects are considered significant.

Proof. Demidenko (2013) ([8] p.74-75) provides a brief proof which we will expand on. It is argued that it is sufficient to prove that there exists a positive semidefinite matrix, $\boldsymbol{M}$, such that $\max_{\boldsymbol{\beta},\sigma^2} l(\boldsymbol{\beta},\sigma^2,\boldsymbol{M}) > l(\boldsymbol{\beta}_{\text{OLS}},{\sigma^2}_{\text{OLS}},0)$. This is simply saying we can find a matrix $\boldsymbol{M}$ which means the log-likelihood dependent on $\boldsymbol{D}=\boldsymbol{M}$ is greater than the log-likelihood dependent on $\boldsymbol{D}=\boldsymbol{0}$, hence the MLE of $\boldsymbol{D}\neq 0$.

The first step is to find the derivative of the log-likelihood. Since we know that $\boldsymbol{Q}=\sigma^2\boldsymbol{D}$, we can say that $\boldsymbol{\Sigma}_i=\sigma^2\boldsymbol{V}_i$, where $\boldsymbol{V}_i=\boldsymbol{I}+\boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^T$. We need to find $\frac{\partial l}{\partial \boldsymbol{D}}$,

$$\frac{\partial}{\partial \boldsymbol{D}}\bigg(-\frac{N}{2}\log(2\pi)-\frac{1}{2}\log|\sigma^2\boldsymbol{V}_i|-\frac{1}{2}{\boldsymbol{e}_i}^T(\sigma^2\boldsymbol{V}_i)^{-1}\boldsymbol{e}_i\bigg),$$

where $\boldsymbol{e}_i=\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}$. To be able to calculate the expression, [8] makes use of Jacobi's formula [12] (Equation 4.13), which gives

$$\frac{\partial}{\partial \boldsymbol{D}}\log|\boldsymbol{V}_i|={\boldsymbol{Z}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{Z}_i\in\mathbb{R}^{k\times k}\ . \tag{4.18}$$

We can also calculate the derivative of ${\boldsymbol{e}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{e}_i$ by using the chain rule which gives

$$\frac{\partial {\boldsymbol{e}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{e}_i}{\partial \boldsymbol{D}}=-{\boldsymbol{Z}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{e}_i{\boldsymbol{e}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{Z}_i\in\mathbb{R}^{k\times k}\ . \tag{4.19}$$

Using equations 4.18 and 4.19, one gets

$$\frac{\partial l}{\partial \boldsymbol{D}}=-\frac{1}{2}\sum_{i=1}^{N}\left({\boldsymbol{Z}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{Z}_i-\frac{1}{\sigma^2}{\boldsymbol{Z}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{e}_i{\boldsymbol{e}_i}^T{\boldsymbol{V}_i}^{-1}\boldsymbol{Z}_i\right)\ .$$

Using a result from multivariate calculus, Demidenko (2013) ([8] p.75) states, if $F(\boldsymbol{x})$ is a function taking a vector argument and $\boldsymbol{g}=\frac{\partial F(\boldsymbol{x}=\boldsymbol{0}}{\partial \boldsymbol{x}}\neq=\boldsymbol{0}$, then there exists a scalar $\lambda>0$ such that, for $\boldsymbol{z}=\lambda\boldsymbol{g}$, one has $F(\boldsymbol{z})>F(\boldsymbol{0})$. The next step follows from a property of positive semidefinite matrices: if a matrix $\boldsymbol{M}$ is positive semidefinite then for any scalar value a, the matrix $a\boldsymbol{M}$ is also positive semidefinite. Hence, if (4.17) is positive semidefinite, then

$$\left.\frac{\partial l}{\partial \boldsymbol{D}}\right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\text{OLS}},\sigma^2=\hat{\sigma}^2_{\text{OLS}},\boldsymbol{D}=0}=-\frac{1}{2}\sum_{i=1}^{N}\boldsymbol{Z}_i{\boldsymbol{Z}_i}^T+\frac{1}{2\sigma^2_{\text{OLS}}}{\boldsymbol{Z}_i}^T\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^T\boldsymbol{Z}_i \tag{4.20}$$

is also a non-zero positive semidefinite, since the matrix in (4.20) is just the matrix in (4.17) multiplied by $-2\hat{\sigma}^2_{\text{OLS}}$, which is a scalar. Therefore, using the fact from multivariate calculus, there exists a matrix $\boldsymbol{M}$ for which

$$\max_{\boldsymbol{\beta},\sigma^2} l(\boldsymbol{\beta},\sigma^2,\boldsymbol{M})\geq l(\hat{\boldsymbol{\beta}}_{\text{OLS}},\hat{\sigma}^2_{\text{OLS}},\boldsymbol{M})>l(\hat{\boldsymbol{\beta}}_{\text{OLS}},\hat{\sigma}^2_{\text{OLS}},\boldsymbol{0})=\max_{\boldsymbol{\beta},\sigma^2} l(\boldsymbol{\beta},\sigma^2,\boldsymbol{0}),$$

hence the MLE of $\boldsymbol{D}$ is nonzero, which implies the random effects are relevant in the model. □

4.4 The Expectation-Maximisation Algorithm

The Expectation-Maximisation (EM) algorithm is an iterative method for finding the maximum likelihood estimates of parameters where the model depends on unobserved latent variables. It involves iteratively cylcing through two steps: an expectation step (E-step) and a maximisation step (M-step). The E-step computes the expectation of the log-likelihood of a dataset conditional on current parameters $\boldsymbol{\theta}$ and the observed data. The M-step involves maximising the conditional expectation from the E-step over values of $\boldsymbol{\theta}$. In the version of the EM Algorithm for linear mixed effects models, the variance components are iteratively updated until they converge. This section references ideas that were introduced by Demidenko (2013) ([8] p.88-92).

Let us rewrite $\boldsymbol{Q} = \sigma^2\boldsymbol{D}$, so that we can take a factor of σ^2 outside the expression in our variance. We then have $\boldsymbol{Y} \sim N(\boldsymbol{X\beta}, \sigma^2(\boldsymbol{ZDZ}^T + \boldsymbol{I}_N))$. The following formulae are used to update the components. We will then derive the explicit formulae needed to perform the EM Algorithm for LME models. To calculate the formulae, we must find the partial derivatives of the log-likelihood with respect to σ^2 and $\boldsymbol{D}$.

$$\sigma^2_{s+1} - \sigma^2_s = \frac{2\sigma^4_s}{N_T}\frac{\partial l}{\partial \sigma^2} \qquad\qquad \boldsymbol{D}_{s+1} - \boldsymbol{D}_s = \frac{2}{N}\boldsymbol{D}_s\frac{\partial l}{\partial \boldsymbol{D}}\boldsymbol{D}_s \tag{4.21}$$

We recall the method to calculate the log likelihood using our new $\boldsymbol{D}$.

$$\begin{aligned} l &= \log\Big(\prod_{i=1}^{N}\frac{1}{(2\pi)^{\frac{n_i}{2}}}\frac{1}{|\sigma^2(\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + \boldsymbol{I}_N)|^{\frac{1}{2}}}\exp\big(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T(\sigma^2(\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + \boldsymbol{I}_N))^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\big)\Big) \\ &= -\frac{N_T}{2}\log(2\pi) + \sum_{i=1}^{N}\left[\log((\sigma^2)^{-\frac{n_i}{2}} - \frac{1}{2}\log(|\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + I_N|) - \frac{1}{2\sigma^2}(\boldsymbol{y}_i - \boldsymbol{X}_i\beta)^T(\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T)^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\beta)\right] \\ &= -\frac{1}{2}\left\{N_T\log(\sigma^2) + \sum_{i=1}^{N}\left[\log(|\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + \boldsymbol{I}_N|) + \frac{1}{\sigma^2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T(\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + \boldsymbol{I}_N)^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right]\right\} \end{aligned}$$

If we denote $\boldsymbol{V}_i = \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^T + \boldsymbol{I}_N$ for ease of notation, we obtain the log-likelihood to be

$$l = -\frac{1}{2}\left\{N_T\log(\sigma^2) + \sum_{i=1}^{N}\left[\log(|\boldsymbol{Z}_i\boldsymbol{D}{\boldsymbol{Z}_i}^T + \boldsymbol{I}_N|) + \frac{1}{\sigma^2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right]\right\} .$$

We must now calculate the partial derivatives of the log likelihood to substitute into the equations in 4.21:

$$\begin{aligned} \frac{\partial l}{\partial \sigma^2} &= -\frac{1}{2}\left\{\frac{N_T}{\sigma^2} - \frac{1}{\sigma^4}\sum_{i=1}^{N}\left[(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right]\right\} \\ &= \frac{1}{2}\left\{-\frac{N_T}{\sigma^2} + \frac{1}{\sigma^4}\sum_{i=1}^{N}\left[(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right]\right\} . \end{aligned}$$

To calculate $\frac{\partial l}{\partial \boldsymbol{D}}$ we need to make use of two identities:

$$\frac{\partial}{\partial \boldsymbol{D}} \log(|\boldsymbol{V}_i|) = \boldsymbol{Z}_i{}^T \boldsymbol{V}_i{}^{-1} \boldsymbol{Z}_i \quad \text{and}$$

$$\frac{\partial}{\partial \boldsymbol{D}} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T \boldsymbol{V}_i{}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}) = -\boldsymbol{Z}_i^T \boldsymbol{V}_i{}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T \boldsymbol{V}_i{}^{-1} \boldsymbol{Z}_i \ .$$

Hence, one has that

$$\frac{\partial l}{\partial \boldsymbol{D}} = -\frac{1}{2} \sum_{i=1}^{N} \left[\boldsymbol{Z}_i{}^T \boldsymbol{V}_i{}^{-1} \boldsymbol{Z}_i - \frac{1}{\sigma^2} \boldsymbol{Z}_i^T \boldsymbol{V}_i{}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T \boldsymbol{V}_i{}^{-1} \boldsymbol{Z}_i \right] \ .$$

Looking back to the iterative formulae (4.21) introduced at the start of this section, we can now substitute the expressions we found in for the partial derivatives to give explicit formulae to calculate the variance components:

$$\begin{aligned} \sigma^2_{s+1} &= \sigma^2_s + \frac{2\sigma^4_s}{N_T} \left(\frac{1}{2} \left(-\frac{N_T}{\sigma^2_s} + \frac{1}{\sigma^4_s} \sum_{i=1}^{N} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)^T \boldsymbol{V}_{is}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s) \right) \right) \\ &= \sigma^2_s - 1 + \frac{1}{N_T \sigma^2_s} \sum_{i=1}^{N} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)^T \boldsymbol{V}_{is}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s) \quad \text{and} \end{aligned}$$

$$\begin{aligned} \boldsymbol{D}_{s+1} &= \boldsymbol{D}_s + \frac{2}{N} \boldsymbol{D}_s \left\{ -\frac{1}{2} \sum_{i=1}^{N} \left[\boldsymbol{Z}_i{}^T \boldsymbol{V}_{is}{}^{-1} \boldsymbol{Z}_i - \frac{1}{\sigma^2} \boldsymbol{Z}_i^T \boldsymbol{V}_{is}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)^T \boldsymbol{V}_{is}{}^{-1} \boldsymbol{Z}_i \right] \right\} \boldsymbol{D}_s \\ &= \boldsymbol{D}_s - \frac{1}{N} \sum_{i=1}^{N} \left[\boldsymbol{D}_s \boldsymbol{Z}_i{}^T \boldsymbol{V}_{is}{}^{-1} \boldsymbol{Z}_i \boldsymbol{D}_s - \frac{1}{\sigma^2} \boldsymbol{D}_s \boldsymbol{Z}_i^T \boldsymbol{V}_{is}^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_s)^T \boldsymbol{V}_{is}{}^{-1} \boldsymbol{Z}_i \boldsymbol{D}_s \right] \ . \end{aligned}$$

The $\boldsymbol{\beta}$s are calculated using the generalised least squares estimate formula:

$$\boldsymbol{\beta}_s = \left[\sum_{i=1}^{N} \boldsymbol{X}_i^T \boldsymbol{V}_i^{-1} \boldsymbol{X}_i \right]^{-1} \sum_{i=1}^{N} \boldsymbol{X}_i \boldsymbol{V}_i^{-1} \boldsymbol{y}_i \ .$$

To perform the EM algorithm, one must cycle through the iterations until the components both converge.

5 Non-linear Mixed Effect Models

In this section, we will consider non-linear mixed effects models (NLMEMs) which consider the case when the random effects enter the model in a non-linear fashion. Due to this, the likelihood of the model cannot be obtained in a closed form. NLMEMs can be extremely effective and allow the range of possible applications to extend to cases in which we want to model non-linear longitudinal data. Throughout the past 50 years, researchers have come up with different methods to obtain an expression for the marginal distribution of the response variable. We will highlight some of these methods after considering the model itself.

5.1 Building The Framework

In this section, we will discuss theory introduced by Demidenko (2013) ([8], p.434-437) and Davidian (2009) ([4], p.116-122). As in the linear case, the idea of considering the structure of the model in two stages (Verbeke and Molenberghs, 2000 ([3], p.20)) can be applied here. For the first stage of the NLMEM, we consider N non-linear regression models each with random subject specific parameters:

$$\boldsymbol{y}_i = \boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i) + \boldsymbol{\epsilon}_i \ , \tag{5.1}$$

where $\boldsymbol{y}_i$ is an $n_i \times 1$ vector of the response variable, $\boldsymbol{f}_i$ is a non-linear $n_i \times 1$ vector function and $\boldsymbol{a}_i$ is a $k \times 1$ vector of unknown subject specific parameters with unknown $k \times k$ covariance matrix $\boldsymbol{Q}$. $\boldsymbol{f}_i$ is such that

$$\boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i) = (\boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i; \boldsymbol{x}_{i1}), \boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i; \boldsymbol{x}_{i2}), ..., \boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i; \boldsymbol{x}_{in_i}))^T \ ,$$

where $\boldsymbol{x}_{ij}$ is a vector of covariates, $\boldsymbol{\gamma}$ is a $q \times 1$ vector of unknown deterministic parameters and $\boldsymbol{\epsilon}_i$ is an $n_i \times 1$ error vector with $E(\boldsymbol{\epsilon}_i) = 0$ and $\text{Cov}(\boldsymbol{\epsilon}_i) = \sigma^2 I$, just like in the linear case.

The second stage of the NLMEM specifies the structure of $\boldsymbol{a}_i$ with

$$\boldsymbol{a}_i = \boldsymbol{A}_i \beta + \boldsymbol{b}_i \ , \tag{5.2}$$

where $\boldsymbol{A}_i$ is a $k \times m$ design matrix, β is an $m \times 1$ vector of population averaged parameters and $\boldsymbol{b}_i$ is a $k \times 1$ vector of random effects with $E(\boldsymbol{b}_i) = 0$ and $\text{Cov}(\boldsymbol{b}_i) = \boldsymbol{Q} = \sigma^2 \boldsymbol{D}$. We assume that the random effects and the error terms are independent. Putting both stages together, the NLMEM can now be written as

$$\boldsymbol{y}_i = \boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{A}_i \beta + \boldsymbol{b}_i) + \boldsymbol{\epsilon}_i \ .$$

The main difference between the NLMEM and the marginal effects model is that the non-linear function is acting upon the random parameters, meaning that it is extremely difficult to obtain properties of estimators in a closed form even when variances are known. This is a common pitfall of non-linear models and a common topic of discussion among mathematicians working on NLMEMs today.

We will calculate the maximum likelihood estimator of the NLME model. If it is assumed that our random terms are normally distributed, Demidenko ([8], p.435) writes the NLMEM as such:

$$\boldsymbol{y}_i | \boldsymbol{a}_i \sim N(\boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i), \sigma^2 \boldsymbol{I}) \quad \text{and} \quad \boldsymbol{a}_i \sim N(\boldsymbol{A}_i \beta, \sigma^2 \boldsymbol{D}) \ ,$$

with both $\boldsymbol{y}_i|\boldsymbol{a}_i$ and $\boldsymbol{a}_i$ having multivariate normal distributions. We will start by explicitly giving the expressions for both probability density functions which we denote by $p()$.

$$p(\boldsymbol{y}_i|\boldsymbol{a}_i) = \frac{1}{(2\pi)^{\frac{n_i}{2}}}\frac{1}{(\sigma^2)^{\frac{n_i}{2}}}\exp\Big\{ -\frac{1}{2\sigma^2}(\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i))^T(\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i))\Big\}$$

Here we have our dimension equal to n_i, and note that the determinant of the identity matrix is equal to 1. We will express $(\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i))^T(\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i))$ as $||\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i)||^2$. We also have

$$p(\boldsymbol{a}_i) = \frac{1}{(2\pi)^{\frac{k}{2}}}\frac{1}{(\sigma^2)^{\frac{k}{2}}}\frac{1}{|\boldsymbol{D}|^{\frac{1}{2}}}\exp\Big\{ -\frac{1}{2\sigma^2}(\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}(\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})\Big\} \ .$$

Here we have the dimension equal to k. We multiply the two probability density functions and then take the product of that to give us the likelihood of the NLME model.

$$\mathcal{L}(\gamma,\beta,\theta,\boldsymbol{a}) = \prod_{i=1}^{N}\frac{1}{(2\pi)^{\frac{1}{2}(n_i+k)}}\frac{1}{(\sigma^2)^{\frac{1}{2}(n_i+k)}}\frac{1}{|\boldsymbol{D}|^{\frac{1}{2}}}\exp\Big\{-\frac{1}{2\sigma^2}\Big[||\boldsymbol{y}_i-\boldsymbol{f_i}(\gamma,\boldsymbol{a}_i)||^2+(\boldsymbol{a}_i-\boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}^{-1}(\boldsymbol{a}_i-\boldsymbol{A}_i\boldsymbol{\beta})\Big]\Big\}$$

However, we do not want our likelihood to be conditional on $\boldsymbol{a}$, so we must integrate this out over $\mathbb{R}^k$.

$$\mathcal{L}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\theta}) = \prod_{i=1}^{N}\frac{1}{(2\pi)^{\frac{1}{2}(n_i+k)}}\frac{1}{(\sigma^2)^{\frac{1}{2}(n_i+k)}}\frac{1}{|\boldsymbol{D}|^{\frac{1}{2}}}\int_{\mathbb{R}^k}\exp\Big\{-\frac{1}{2\sigma^2}\Big[||\boldsymbol{y}_i-\boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i)||^2+(\boldsymbol{a}_i-\boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}^{-1}(\boldsymbol{a}_i-\boldsymbol{A}_i\boldsymbol{\beta})\Big]\Big\}d\boldsymbol{a}$$

We now take the logarithm of our likelihood. We then use the property of logarithms that the logarithm of a product of values is the sum of the logarithms of those values. We use this property multiple times across different steps. We have

$$\begin{aligned} l(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\theta}) &= \log\Big(\prod_{i=1}^{N}\frac{1}{(2\pi)^{\frac{1}{2}(n_i+k)}}\Big) + \log\Big(\prod_{i=1}^{N}\frac{1}{(\sigma^2)^{\frac{1}{2}(n_i+k)}}\Big) + \log\Big(\prod_{i=1}^{N}\frac{1}{|\boldsymbol{D}|^{\frac{1}{2}}}\Big) \\ &\quad + \log\Big(\prod_{i=1}^{N}\int_{\mathbb{R}^k}\exp\Big\{ -\frac{1}{2\sigma^2}\Big[||\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i)||^2 + (\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}^{-1}(\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})\Big]\Big\}d\boldsymbol{a}\Big) \ . \end{aligned}$$

We now calculate the products where possible. Denote $N_T = \sum n_i$.One has

$$\begin{aligned} l(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\theta}) &= \log\Big(\frac{1}{(2\pi)^{\frac{1}{2}(N_T+Nk)}}\Big) + \log\Big(\frac{1}{(\sigma^2)^{\frac{1}{2}(N_T+Nk)}}\Big) + \log\Big(\frac{1}{|\boldsymbol{D}|^{\frac{N}{2}}}\Big) \\ &\quad + \sum_{i=1}^{N}\log\Big(\int_{\mathbb{R}^k}\exp\Big\{ -\frac{1}{2\sigma^2}\Big[||\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i)||^2 + (\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}^{-1}(\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})\Big]\Big\}d\boldsymbol{a}\Big) \\ &= -\frac{1}{2}\Big[(N_T+Nk)\log(2\pi) + (N_T+Nk)\log(\sigma^2) + N\log(|\boldsymbol{D}|)\Big] \\ &\quad + \sum_{i=1}^{N}\log\Big(\int_{\mathbb{R}^k}\exp\Big\{ -\frac{1}{2\sigma^2}\Big[||\boldsymbol{y}_i - \boldsymbol{f_i}(\boldsymbol{\gamma},\boldsymbol{a}_i)||^2 + (\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})^T\boldsymbol{D}^{-1}(\boldsymbol{a}_i - \boldsymbol{A}_i\boldsymbol{\beta})\Big]\Big\}d\boldsymbol{a}\Big) \quad (5.3) \end{aligned}$$

The integral in the log-likelihood makes the maximum likelihood procedure difficult since it means one does not have a closed-form expression for the likelihood. The next section will highlight a method to obtain an estimate for $\boldsymbol{\beta}$ and $\boldsymbol{Q} = \sigma^2\boldsymbol{D}$ in the non-linear framework.

5.2 Estimation of Parameters in the NLMEM Framework

This section will explore a method for computing an estimate for the coefficient vector $\boldsymbol{\beta}$ and the random effects covariance matrix $\boldsymbol{Q}$. This method is introduced by Demidenko (2013) ([8] p.472-475) and works around the issue of the integral in the log-likelihood of the non-linear mixed effects model, as highlighted in Section 5.1. The overall idea is to compute an estimate of $\boldsymbol{a}_i$ and its covariance matrix and then maximise the likelihood to find an estimate for $\boldsymbol{\beta}$.

The first step is finding a value of $\boldsymbol{a}_i$ which minimises

$$||\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i)||^2 \ .$$

Denote this value ${\boldsymbol{a}_i}^*$. One also must find an estimate of the covariance matrix of ${\boldsymbol{a}_i}^*$, $\text{Cov}({\boldsymbol{a}_i}^*|\boldsymbol{b}_i)$. One can do this by using theory from multivariate calculus, by viewing $\boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i)$ as a surface which we want to approximate using a tangent plane evaluated at a point, ${\boldsymbol{a}_i}^*$ (Ruckstuhl, 2020 [15], p.8). The result claims

$$\boldsymbol{f}_i(\boldsymbol{\gamma}, \boldsymbol{a}_i) \approx \boldsymbol{f}_i(\boldsymbol{\gamma}, {\boldsymbol{a}_i}^*) + \boldsymbol{R}_i({\boldsymbol{a}_i}^*)(\boldsymbol{a}_i - {\boldsymbol{a}_i}^*) \ , \tag{5.4}$$

where $\boldsymbol{R}_i({\boldsymbol{a}_i}^*)$ is the $n_i \times k$ derivative matrix of $\boldsymbol{f}_i$ evaluated at the point ${\boldsymbol{a}_i}^*$. Ruckstuhl (2020) ([15] p.15) then shows that if we substitute 5.4 into 5.1 and rearrange to get

$$\tilde{\boldsymbol{y}_i} = \boldsymbol{y}_i - \boldsymbol{f}_i({\boldsymbol{a}_i}^*) = \boldsymbol{R}_i({\boldsymbol{a}_i}^*)(\boldsymbol{a}_i - {\boldsymbol{a}_i}^*) + \boldsymbol{\epsilon}_i \ , \tag{5.5}$$

this can be viewed as a linear regression problem. Then following from theory of asymptotic distributions of the least squares estimators, one has

$${\boldsymbol{a}_i}^*|\boldsymbol{b}_i \sim N(\boldsymbol{a}_i, \sigma^2(\boldsymbol{R}_i(\boldsymbol{a}_i)^T \boldsymbol{R}_i(\boldsymbol{a}_i))^{-1}) \ ,$$

where one can estimate the covariance with $\boldsymbol{C}_i = \hat{\sigma}^2({\boldsymbol{R}_i(\boldsymbol{a}_i)^*}^T \boldsymbol{R}_i(\boldsymbol{a}_i)^*)^{-1})$ where $\hat{\sigma}^2$ is the pooled variance as such:

$$\hat{\sigma}^2 = \frac{1}{\sum_i (n_i - k)} \sum_{i=1}^{N} ||\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{\gamma}, {\boldsymbol{a}_i}^*)||^2 \ .$$

Then using a result from multivariate statistics, which says, if $\boldsymbol{Y}$ is a vector random variable dependent on a random variable $\boldsymbol{u}$, then

$$\text{Cov}(\boldsymbol{Y}) = \text{Cov}[\mathbb{E}(\boldsymbol{Y}|\boldsymbol{u})] + \mathbb{E}[\text{Cov}(\boldsymbol{Y}|\boldsymbol{u})] \ .$$

Now note that $\mathbb{E}(\boldsymbol{a}_i|\boldsymbol{b}_i) = \boldsymbol{A}_i\boldsymbol{\beta} + \boldsymbol{b}_i$ and hence $\text{Cov}[\mathbb{E}(\boldsymbol{a}_i|\boldsymbol{b}_i)] = \text{Cov}(\boldsymbol{b}_i)$. Therefore one has

$$\text{Cov}({\boldsymbol{a}_i}^*) = \text{Cov}({\boldsymbol{a}_i}^*|\boldsymbol{b}_i) + \text{Cov}(\boldsymbol{b}_i) \approx \boldsymbol{C}_i + \boldsymbol{Q} \ .$$

Now an estimate for the covariance of the estimate of $\boldsymbol{a}_i$ has been obtained, Demidenko (2013) ([8] p.445) assumes that ${\boldsymbol{a}_i}^*$ is normally distributed and we can explicitly calculate the parameters.

$${\boldsymbol{a}_i}^* \sim (\boldsymbol{A}_i\boldsymbol{\beta}, \boldsymbol{C}_i + \boldsymbol{Q}) \tag{5.6}$$

This gives a log-likelihood that can be maximised to calculate an estimate for $\boldsymbol{\beta}$. The likelihood for this distribution is of the form

$$L(\boldsymbol{\beta}, \boldsymbol{Q}) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{C}_i + \boldsymbol{Q}|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2} (\boldsymbol{a}_i{}^* - \boldsymbol{A}_i\boldsymbol{\beta})^T (\boldsymbol{C}_i + \boldsymbol{Q})^{-1} (\boldsymbol{a}_i{}^* - \boldsymbol{A}_i\boldsymbol{\beta}) \Big) \ ,$$

and hence the log-likelihood which we want to maximise over is of the form

$$l(\boldsymbol{\beta}, \boldsymbol{Q}) = -\frac{1}{2} \sum_{i=1}^{N} \Big(\log(|\boldsymbol{C}_i + \boldsymbol{Q}|) + (\boldsymbol{a}_i{}^* - \boldsymbol{A}_i\boldsymbol{\beta})^T (\boldsymbol{C}_i + \boldsymbol{Q})^{-1} (\boldsymbol{a}_i{}^* - \boldsymbol{A}_i\boldsymbol{\beta}) \Big) \ .$$

This expression can be maximised using a variety of maximisation algorithms to calculate an estimate for $\boldsymbol{\beta}$.

5.3 Modelling the `Theophylline` Data

The following section will attempt to apply a non-linear model framework to the `Theophylline` data previously introduced in Section 2.2. It is clear in Figure 2.2 that the relationship between theophylline concentration and time is non-linear. We can also see that the growth curve for each individual has the same overall shape, however there is variability in how quickly the concentration increases and decreases. We aim to model the `Theophylline` dataset using non-linear mixed effects models which account for the unknown variability between individuals.

5.3.1 Applying a Non-Linear Model to the `Theophylline` Data

Before fitting a non-linear mixed effects model, Chiquet [16] fits a non-linear model of the form

$$y_{ij} = f(t_{ij}, \boldsymbol{\psi}) + \epsilon_{ij}, \qquad 1 \leq i \leq N, \ \ 1 \leq j \leq n_i \ ,$$

where y_{ij} is the measurement for individual i and time j. We denote $\boldsymbol{\psi}$ as the vector of parameters. The `Theophylline` dataset is often modelled using a pharmacokinetic model (for example by Davidian (2009) ([4], p.111-114) and by Chiquet [16]) f to model the data which is of the form

$$f(t_{ij}, \psi) = \frac{D k_a}{V(k_a - k_e)} (e^{-k_e t} - e^{-k_a t}) \ ,$$

where $\boldsymbol{\psi} = (k_a, V, k_e)$ are parameters of the model and D is the amount of theophylline given to each patient. Each patient was administered 320mg of the drug so $D = 320$ for all individuals. The model is then ready to be fitted using the `nls` function, from the `stats` package [17], in `R` (Chiquet [16]) and the resulting fitted model is of the form

$$f(t_{ij}, \psi) = 10.081(e^{-0.0793t} - e^{-1.5798t}) \ ,$$

where $\psi = (1.5798, 33.4218, 0.0793)$. The `nls` function fits this model by aiming to minimise the least squares estimate of ψ using initial values $\psi_{\text{init}} = (k_a = 1, V = 40, k_e = 0.1)$. The plot of the predicted concentration is shown by the lime green line in Figure 5.1.

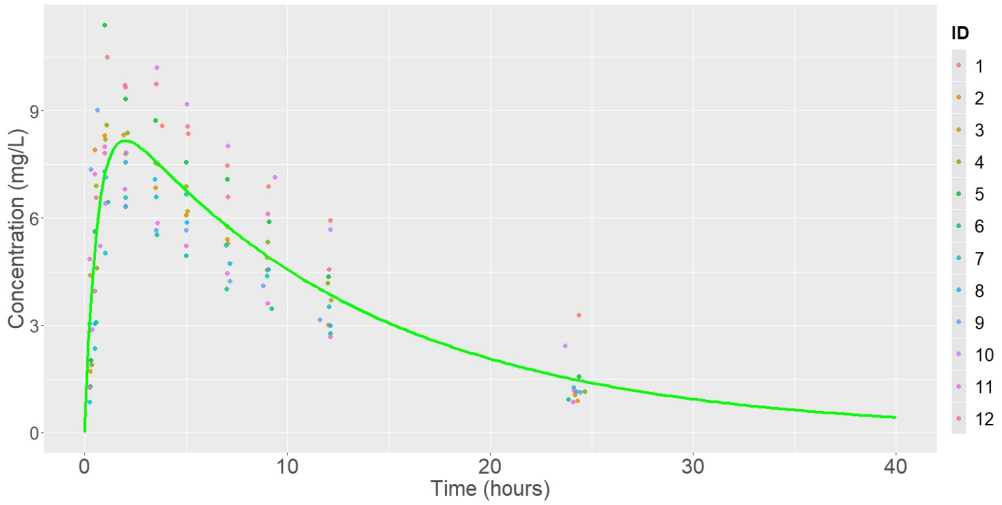


Figure 5.1: Pharmacokinetic Model Fitted to `Theophylline`

This allows us to make estimates for the average individual within the sample of patients, however this fits one model to the entire dataset. Hence, we learn little about the specific individuals and do not account for the variability between subjects. A solution to this is to fit a different non-linear model to each individual. This gives us a different set of estimated parameters, ψ_i, for each individual. We can see how the different models predict the concentration of theophylline in Figure 5.2.

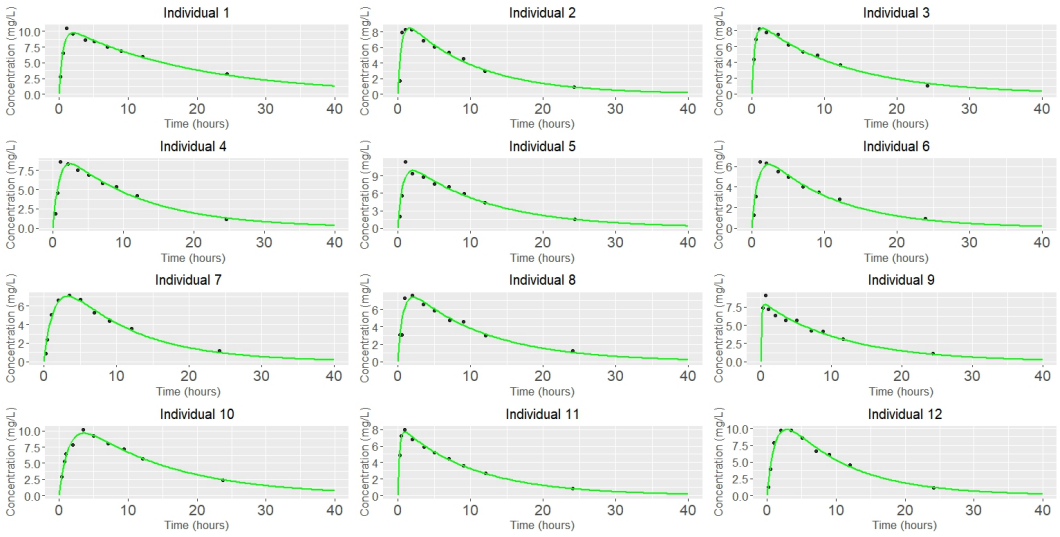


Figure 5.2: Individual Pharmacokinetic Model Fitted to Each Individual

We can see that the trends for each individual are captured by their respective models. However,

fitting n separate models for n individuals is not the most robust approach. This does not account for the rest of the data and for the unknown sources of variation that leads to correlation between subjects. Hence, we must apply a non-linear mixed effects model.

5.3.2 Applying a NLMEM to the `Theophylline` Data

We showed in Section 5.2.1 that one can fit a non-linear model to a dataset using the `nls` function within `R` by calculating the least squares estimate of the parameter vector, $\boldsymbol{\psi}$. In this section, we will look to account for the unknown variability, which leads to correlation between subjects, by fitting a non-linear mixed effects model to the dataset, using theory outlined in Section 5.1. The NLMEM has the structure

$$y_{ij} = f(t_{ij}, \boldsymbol{\psi}_i) + \epsilon_{ij} \ ,$$

where our parameter vector, $\boldsymbol{\psi}_i$, is now going to be decomposed into a fixed part and a random part as such

$$\boldsymbol{\psi}_i = \boldsymbol{\psi}_{\text{pop}} + \boldsymbol{b}_i \ ,$$

where $\boldsymbol{\psi}_{\text{pop}}$ is the population parameter (fixed effect) and $\boldsymbol{b}_i$ is the random effect for each individual parameter. It is assumed $\epsilon_{ij} \sim N(0, \sigma^2)$ and $\boldsymbol{b}_i \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{D})$. Hence

$$y_{ij} \sim N(f(t_{ij}, \boldsymbol{\psi}_i), \sigma^2) \quad \text{and} \quad \boldsymbol{\psi}_i \sim N(\boldsymbol{\psi}_{\text{pop}}, \sigma^2 \boldsymbol{D}).$$

We follow Chiquet (2020) [16] by using the `saemix` package (Comets et al. 2017 [18]) to fit the model. The package uses the stochastic approximation expectation maximisation (SAEM) algorithm to compute the parameters of the model. We include a brief explanation of the method. It was shown in Section 5.1 that the likelihood cannot be expressed in a closed form, hence a stochastic approximation is required. Comets et al. (2017)([18] p.4) wants to maximise the likelihood using an altered EM algorithm. In non-linear mixed effects models, the E-step of the algorithm of iteration k computes the conditional expectation of the log-likelihood $l_k(\boldsymbol{\theta}) = \mathbb{E}(l(\boldsymbol{y}, \boldsymbol{\psi}; \boldsymbol{\theta}) | \boldsymbol{y}, \boldsymbol{\theta}_{k-1})$. The M-step consists of finding the $\boldsymbol{\theta}_k$ that maximises $l_k(\boldsymbol{\theta})$.

It is explained by Comets et al. (2017 [18]), that in the case where the function f does not depend linearly on the random effects, the E-step of the EM algorithm must be replaced by a stochastic counterpart method. We explain the steps of the procedure for the k^{th} iteration of the SAEM iteration.

Step 1 (Simulation): draw $\boldsymbol{\psi}^{(k)}$ from the conditional distribution $p(.|\boldsymbol{y}, \boldsymbol{\theta}_k)$.

Step 2 (Stochastic Approximation): update $l_k(\boldsymbol{\theta})$ as such:

$$l_k(\boldsymbol{\theta}) = l_{k-1}(\boldsymbol{\theta}) + \eta_k(l(\boldsymbol{y}, \boldsymbol{\psi^{(k)}}; \boldsymbol{\theta}) - l_{k-1}(\boldsymbol{\theta})) \ .$$

Step 3 (Maximisation): update $\boldsymbol{\theta}_k$ as such:

$$\boldsymbol{\theta}_{k+1} = \underset{\boldsymbol{\theta}}{\text{argmax}} \ l_k(\boldsymbol{\theta}) \ .$$

The $\boldsymbol{\psi}^{(k)}$s in Step 1 are simulated at each iteration via an Monte Carlo Markov Chain (MCMC) method from the conditional distribution of the individual parameters. Note that in Step 2, the η_ks are a sequence which control the converge of the SAEM algorithm. They can be thought of as a

learning rate for the algorithm.

We specify that the algorithm should find individual estimates for each parameter and it should estimate the Fisher information matrix. The fitted model had statistics shown in Table 5.1.

Estimates for Fixed Effects			
Parameter	Estimate	Standard Error	CV(%)
k_a	1.8280	0.34060	18.63
V	32.6824	1.61419	4.94
k_e	0.0874	0.00616	7.05
Variance of Random Effects			
Parameter	Estimate	Standard Error	CV(%)
$\text{Var}(k_a)$	1.20162	5.47×10^{-1}	45.5
$\text{Var}(V)$	23.23684	11.5	49.1
$\text{Var}(k_e)$	0.00019	1.66×10^{-4}	87.3
Statistical Criteria			
-2LL	345.0071		
AIC	359.0071		
BIC	362.4015		

Table 5.1: Results of NLMEM Fitted to `Theophylline` Using `saemix`

We can also obtain individual parameter estimates using the `saemix` package. These estimates can be used to plot individual predictions shown in Figure 5.3.

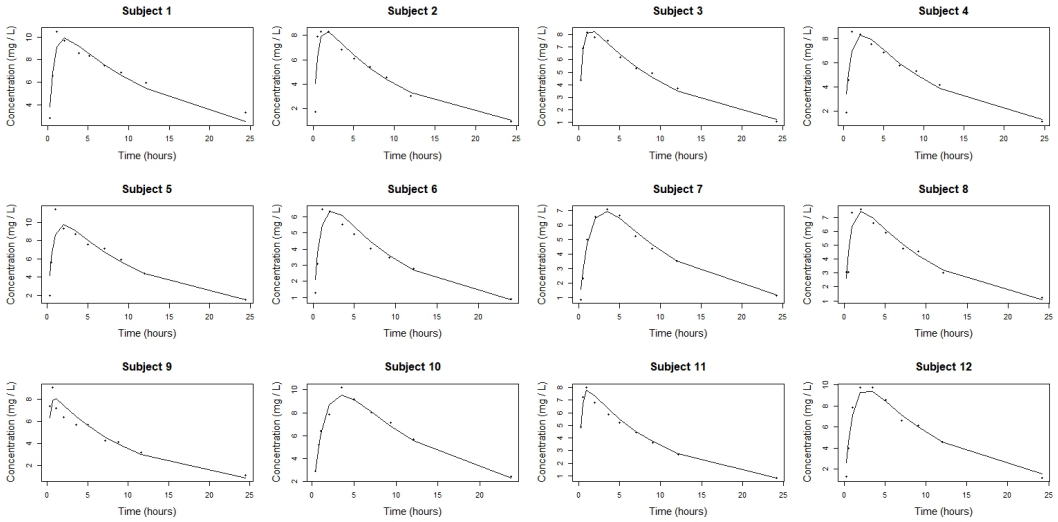


Figure 5.3: Predictions for Each Individual Computed Using the NLMEM

One can also fit a model with covariates, where the covariates can be used to explain the variability of the non-observed individual parameters $\boldsymbol{\psi}_i$. In the `Theophyilline` dataset, we have another variable titled `weight`, which is the weight of the individual in kilograms. We look to include this in our model. Chiquet [16] considers a model where the volume parameter V_i follows a normal distribution and is a function of the weight w_i as such:

$$V_i = V_{\text{pop}} + \beta(w_i - w_{\text{pop}}) + b_{V,i}$$

The model includes a centred weight and the predicted volume for an average individual will be V_{pop} as expected. Using the average within sample weight, 69.5kg, allows us to fit a NLMEM containing covariates. The model's statistics are contained in Table 5.2.

Estimates for Fixed Effects			
Parameter	Estimate	Standard Error	CV(%)
k_a	1.6025	0.30631	19.11
V	32.7236	1.39187	4.25
β	0.3336	0.13977	41.90
k_e	0.0859	0.00607	7.06
Variance of Random Effects			
Parameter	Estimate	Standard Error	CV(%)
Var(k_a)	0.3894	0.1733	44.5
Var(V)	15.5818	8.2225	52.8
Var(k_e)	0.0257	0.0213	82.9
Statistical Criteria			
-2LL		333.2529	
AIC		349.2529	
BIC		353.1321	

Table 5.2: Statistics of an NLMEM Containing Covariates Fitted Using `saemix`

One can note that the statistical criteria of our second model are all smaller than those of the first model, suggesting that including the covariates in the fitting of the model improves the overall fit. The p-value of the test $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$ is 0.017 which suggests that the volume increases as the weight increases.

6 Missing Data in Longitudinal Datasets

6.1 Introduction to Missing Data

6.1.1 Motivations

Consider setting up a longitudinal data study. Ideally, you would collect data on every individual at the initial and follow up times you desire. In practice, this 'perfect' longitudinal data study is rare. Many studies have missing observations due to a variety of reasons. These reasons can be split into three groups, which we will consider later.

An individual's response can be missing at a particular time. For example, an individual may not be able to attend a study one day due to other commitments. This does not necessarily mean they will not attend the next time. On the other hand, longitudinal data studies often suffer from the problem of dropout, where the individual stops participating in the study at some point of its duration and does not provide any more responses. In this chapter, we will explore models that deal with both of these issues.

When longitudinal data is incomplete, there are many consequences to its analysis. How does the accuracy of our prediction change when we incur a loss of information? When there is missing data, the validity of any analysis is based on the assumptions that are made about the reasons for missingness, which is referred to as the missing data mechanism.

6.1.2 Structure of Missing Data Mechanisms

We will continue to use notation previously introduced. Recall that in longitudinal data we have N individuals who we take n_i measurements from individual i. An individual with a complete set of responses will have response vector $Y_i = (y_{i1}, ..., y_{in_i})^T$. We also have our matrix of fixed covariates $\boldsymbol{X}_i$ for each of our $\boldsymbol{Y}_i$. We now introduce our first new notation of this chapter. Let $\boldsymbol{R}_i \in \mathbb{R}^{n_i \times 1}$ be a vector of response indicators. We define R_{ij} as so:

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

Given our $\boldsymbol{R}_i$, we can partition our $\boldsymbol{Y}_i$, our set of intended responses, into two disjoint sets, $\boldsymbol{Y}_i{}^o$ and $\boldsymbol{Y}_i{}^m$. $\boldsymbol{Y}_i{}^o$ denotes the vector of the observed responses and $\boldsymbol{Y}_i{}^m$ denotes the vector of those which are missing. The missing data mechanism describes the probability that a response is missing at any one point. It specifies a probability model for the distribution of $\boldsymbol{R}_i$, conditional on $\boldsymbol{X}_i$, $\boldsymbol{Y}_i{}^o$ and $\boldsymbol{Y}_i{}^m$.

6.1.3 Types of Missing Data

We will now introduce the three types of missing data and their mathematical definitions.

We will first consider data that is *Missing Completely At Random (MCAR)*. Longitudinal data is MCAR when $\boldsymbol{R}_i$ is independent of both $\boldsymbol{Y}_i{}^o$ and $\boldsymbol{Y}_i{}^m$, the observed and missing components of the response. For example, consider a salary questionnaire getting lost in the mail. The fact the response is missing

is unrelated to the questionnaire or the outcomes of the questionnaire. Mathematically, we say data is MCAR when

$$\mathbb{P}(\boldsymbol{R}_i|\boldsymbol{Y}_i^o, \boldsymbol{Y}_i^m, \boldsymbol{X}_i) = \mathbb{P}(\boldsymbol{R}_i) \ .$$

We say data is *Missing At Random (MAR)* when the probability that the response is missing is dependent on the observed responses but is unrelated to the responses that are unobserved. For example, someone may not fill out a salary questionnaire due to privacy concerns. This is not related to the nature of the questionnaire. Mathematically, we say data is MAR when

$$\mathbb{P}(\boldsymbol{R}_i|\boldsymbol{Y}_i^o, \boldsymbol{Y}_i^m, \boldsymbol{X}_i) = \mathbb{P}(\boldsymbol{R}_i|\boldsymbol{Y}_i^o, \boldsymbol{X}_i) \ .$$

If we assume that the data is MAR, there are several consequences. The individuals who have provided 'complete' data are a biased subset of the entire population. Therefore an analysis restricted to this subset is considered invalid. With MAR, the observed data cannot be considered a random sample of the target population.

We say the data are *Missing Not At Random (MNAR)* when the probability that the response is missing is dependent on the observed and unobserved responses. For example, someone who is wanting to evade tax would be unlikely to fill out a salary questionnaire. The missing data is directly related to the outcome of the survey. Mathematically, we say data is MNAR when

$$\mathbb{P}(\boldsymbol{R}_i|\boldsymbol{Y}_i^o, \boldsymbol{Y}_i^m, \boldsymbol{X}_i) = \mathbb{P}(\boldsymbol{R}_i|\boldsymbol{Y}_i^o, \boldsymbol{Y}_i^m, \boldsymbol{X}_i) \ .$$

An MNAR missing data mechanism is referred to as *non-ignorable* since the missing data mechanism cannot be ignored when we are making inferences about the distribution of the complete data.

The `lmer` function in `R` can still fit a linear mixed effects model when there are missing observations, which are assumed to be missing at random (MAR). It does this by ignoring the missing values and just considering the observed values, in a technique called complete case analysis. We randomly took different amounts of data out of the `ChickWeight` dataset and fit a random intercept and slope model, as in Section 3.2, to the data. We then measured the residual sum of squares (RSS) of the models applied to the `ChickWeight` dataset.

$$\text{RSS} = \sum_{i=1}^{50} \sum_{j=1}^{n_i} (y_{ji} - \hat{y_{ji}})^2$$

We sum from $i = 1, .., 50$ since we have 50 chicks each having n_i measurements. The result is shown in Figure 6.1. One can see that as the percentage of missing data increases, so does the residual sum of squares. This suggests that the model is a worse fit of the data, since this means that our predicted values are venturing further away from our true values. This is what we intuitively expect. Hence, we need a method to be able to impute data so we do not have to deal with the problem of fitting a model to a dataset containing missing values.

Figure 6.1: RSS of LMEMs Fitted with Percentages of Data Missing

6.2 Imputation

One way we can obtain a data set on which complete data methods can be deployed is by filling in (imputing) the missing values, instead of deleting subjects who have missing data or who have dropped out. Imputation is extremely popular in survey methods. However, one must be careful when performing imputation (Verbeke and Molenberghs, 2000) ([3] p.224). Dempster and Rubin (1983) [19] write:

"The idea of imputation is both seductive and dangerous. It is seductive because it can lull the user into the pleasurable state of believing that the data are complete after all, and it is dangerous because it lumps together situations where the problem is sufficiently minor that it can be legitimately handled in this way and situations where standard estimates applied to the real and imputed data have substantial biases."

There are several issues that could arise from imputation. If the imputation is wrong, then the estimates would be biased. We would also be hiding the uncertainty resulting from missingness.

6.2.1 Simple Imputation Methods

We will first consider 'Last Observation Carried Forward' (LOCF) imputation. The idea is simple - wherever we have an individual with a missing measurement, we replace it with the most recently observed measurement from that individual. Often unrealistic assumptions have to be made for this method. We have to assume that the subject's measurement stays the same from the point they dropout or whilst they are missing from the study. We can immediately see that this is a extremely bold assumption. In the context of our `ChickWeight` data, this would mean assuming that for those chicks who dropped out, if they were not to have dropped out, that their weight would have never changed from that point onward.

Now consider single mean imputation, where we replace a missing value with the average of the observed values at that time point over the other subjects for which this variable is non-missing. We

will use this method for our `ChickWeight` data to show some potential flaws with this method. The original data is shown in Table 6.1.

Chick ID	Diet	Weight (g))											
		t=0	t=2	t=4	t=6	t=8	t=10	t=12	t=14	t=16	t=18	t=20	t=21
8	A	42	50	61	71	84	93	110	116	126	134	125	NA
15	A	41	49	56	64	68	68	67	68	NA	NA	NA	NA
16	A	41	45	49	51	57	51	54	NA	NA	NA	NA	NA
18	A	39	35	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
44	D	42	51	65	86	103	118	127	138	145	146	NA	NA

Table 6.1: Original Data for all Dropout Subjects

Above we see all recorded measurements for the subjects which did not complete the trial. We denote their dropout time as the point where they first do not record a measurement, or the first time NA is reported in the table above. Table 6.2 contains the data once mean substitution has been performed.

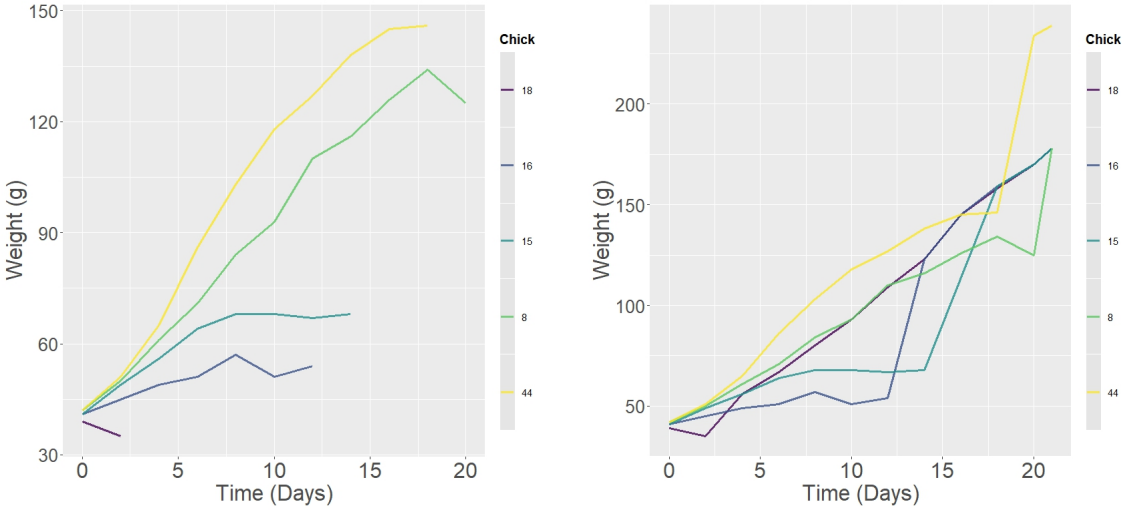
Chick ID	Diet	Weight (g))											
		t=0	t=2	t=4	t=6	t=8	t=10	t=12	t=14	t=16	t=18	t=20	t=21
8	A	42	50	61	71	84	93	110	116	126	134	125	178
15	A	41	49	56	64	68	68	67	68	145	159	170	178
16	A	41	45	49	51	57	51	54	123	145	159	170	178
18	A	39	35	56	67	80	93	109	123	145	159	170	178
44	D	42	51	65	86	103	118	127	138	145	146	234	239

Table 6.2: Data for all Dropout Subjects after Mean Substitution (Imputed Measures Marked in Red)

Here we have fully imputed our dropout values. Figure 6.2 visualises the imputation 'before and after'. According to our imputation, this is what we would have seen if the chicks had not dropped out. One could argue there are a few issues with this approach. First, we see that in Figure 6.2b for chicks $8, 15, 16, 18$ that since they all follow the same diet, their mean imputed values have been calculated using the same samples, hence output the same values. The likelihood of four chickens out of twenty all completing the trial at the same weight, and some weighing the same for the final few observation points, is unlikely. We can see that chick 44 and chick 15 had sudden increases in weight between occasions, which also leads us to question the validity of this approach for modelling dropout data. Figure 6.2a suggests that chicks often lose weight before dropping out, which suggests that dropouts may be a result of death. This creates the question of how to approach dropout of this kind, as it may not be valid to impute values of a subject if the subject can no longer be measured.

6.2.2 Multiple Imputation

Multiple imputation (MI) was introduced by Rubin (1978) [20] and has since become a key tool used in the analysis of incomplete data. It involves generating multiple values for each missing value and then performing analysis on each complete dataset imputed and then combining all analysis to decide on a single imputation. Verbeke and Molenberghs (2005) ([21] p. 512) split multiple imputation into



(a) Original Data for Dropout Chicks

(b) Data After Single Mean Imputation

Figure 6.2: Plots Showing the Original Data 6.2a and the Single Mean Imputed Data 6.2b

three stages, making references to Rubin (1978) [20]:

Step 1: The incomplete values are imputed M times, creating M sets of complete data.

Step 2: Analysis is performed on each of the M datasets.

Step 3: The results from the analyses are combined into a final imputation, giving a single complete dataset.

First considerations would tell us that multiple imputation would give us a better answer than any form of single imputation, since we are giving ourselves the option to compare multiple different possibilities. Verbeke and Molenberghs (2005) ([21] p.512) and Ogundimu (2023) ([22] p.22) both highlight a theoretical justification of multiple imputation which we will refer to.

Our aim is to estimate the parameter vector $\boldsymbol{\theta}$ from the data to tell us about the distribution of $\boldsymbol{Y}$. One uses multiple imputation to impute the missing data $\boldsymbol{Y}^m$ using the observed data $\boldsymbol{Y}^o$, M times, and then the results from the M complete datasets are used to estimate $\boldsymbol{\theta}$. Ideally, one would want to sample from the true distribution $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta})$ to draw values of $\boldsymbol{Y}_i{}^m$ which can be used for imputation. Since $\boldsymbol{\theta}$ is unknown, it must be estimated from the complete data available, giving an estimate of $\hat{\boldsymbol{\theta}}$. However $\boldsymbol{\theta}$ is a random variable so must have a certain amount of uncertainty. It is assumed that $\boldsymbol{\theta}$ is a random variable with mean $\hat{\boldsymbol{\theta}}$, the estimate taken from the complete data. Ogundimu (2023) [22] claims one then draws a random $\boldsymbol{\theta}^*$ from the distribution of $\boldsymbol{\theta}$. One then substitutes this $\boldsymbol{\theta}^*$ into the conditional distribution to draw a $\boldsymbol{Y}_i{}^m$ from $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta}^*)$.

We will now describe the multiple imputation algorithm as mentioned by Ogundimu (2023) ([22] p.23)

and Verbeke and Molenberghs (2005) ([21] p.513).

Step 1: Draw $\boldsymbol{\theta}^*$ from the posterior distribution of $\boldsymbol{\theta}$

Step 2: Draw the imputed data $\boldsymbol{Y}_i{}^{m*}$ from the conditional distribution $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta}^*)$

Step 3: Calculate the estimate of a parameter of interest, for example $\boldsymbol{\beta}$, and its estimated variance, using the now-complete data $(\boldsymbol{Y}^o, \boldsymbol{Y}^{m*})$:

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\boldsymbol{Y}^o, \boldsymbol{Y}^{m*})$$

The within-imputation variance is $\boldsymbol{U} = \hat{\text{Var}}(\boldsymbol{\beta})$

Step 4: Repeat steps 1,2 and 3 M times, producing estimates $\hat{\boldsymbol{\beta}}^m$ and $\boldsymbol{U}^m$ each time for $m = 1, 2, ..., M$.

After performing the algorithm, M inferences need to be analysed to combine into a single conclusion. One can take the average of all M parameter estimates for $\hat{\beta}^m$ to give the pooled parameter estimate of $\boldsymbol{\beta}$.

$$\boldsymbol{\beta}^* = \frac{1}{M}\sum_{m=1}^{M} \hat{\boldsymbol{\beta}}^{(m)}$$

One has the average within imputation variance as

$$\boldsymbol{W} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{U}^{(m)} \ ,$$

and the between imputation variance as

$$\boldsymbol{B} = \frac{1}{M-1}\sum_{m=1}^{M} (\hat{\boldsymbol{\theta}}^{(m)} - \boldsymbol{\beta}*)^2 \ ,$$

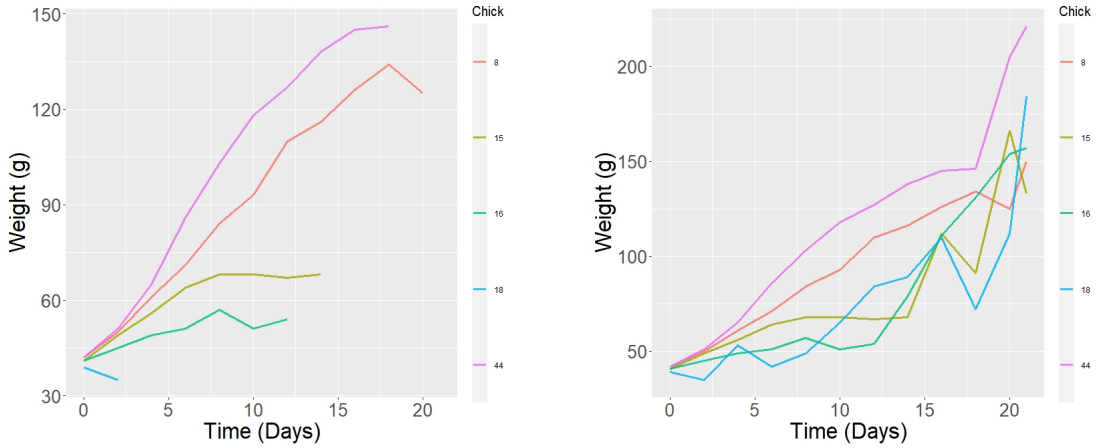
which gives the total variance, $\boldsymbol{T}$, for $(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^*)$ as

$$\boldsymbol{T} = \boldsymbol{W} + \big(1 + \frac{1}{M}\big)\boldsymbol{B} \ .$$

This allows us to make inferences for $\boldsymbol{\beta}$ based on

$$(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^*) \sim N(0, \boldsymbol{T}) \ .$$

We will now perform multiple imputation on the `ChickWeight` dataset using the `mice` function from the `mice` [23] package in `R`. For the `mice` function to work we need to set 'missing' values to `NA` so we had to amend the `ChickWeight` dataset to include missing values, instead of having the rows containing dropout values removed from the dataset. This allows `R` to recognise the missingness. The results of the multiple imputation are displayed in Figure 6.3b. We can see that, unlike the single mean imputation in Figure 6.2b, multiple imputation gives different imputed values for chicks on the same diet. We can still argue against how realistic our imputed values are. We have seen, in Figure



(a) Original Data for Dropout Chicks

(b) Data After Multiple Imputation

Figure 6.3: Plots Showing the Original Data (6.3a) and Multiple Imputed Data (6.3b)

2.1, that the weight of the chicks rarely decreased over the 21 days they were measured. However, in our imputed data, it is common for the weight to decrease between measurements and then increase again. This again questions the validity of imputation of dropout data, especially for individuals who dropped out in the earlier stages of a trial.

6.2.3 Multiple Imputation using Chained Equations (MICE)

If there are missing values in more than one variable, one can implement Multiple Imputation using Chained Equations. We start by introducing the algorithm, as included highlighted by Ogundimu (2023) ([22] p.41) and presented in Algorithm 1.

Algorithm 1 Multiple Imputation using Chained Equations (MICE)

for j in $1, ..., p$ **do** ▷ Initialising the Imputed Values
  Specify the imputation model for variable $Y_j : f({Y_j}^{\text{mis}} | {Y_j}^{\text{obs}}, Y_{-j}, R)$
  Fill in initial imputations $\dot{Y}_j^0$ by taking random draws from ${Y_j}^{\text{obs}}$
end for

for n in $1, ..., N$ **do** ▷ Looping through iterations
  for j in $1, ..., p$ **do** ▷ Looping through variables
    Take the currently complete dataset except Y_j
    $\dot{Y}_{-j}^t = \left(\dot{Y}_1^t, ..., \dot{Y}_{j-1}^t, \dot{Y}_{j+1}^{t-1}, ..., \dot{Y}_p^{t-1}\right)$
    Draw parameters $\dot{\theta}_j^t \sim f({\theta_j}^t | {Y_j}^{\text{obs}}, {Y_{-j}}^t, R)$
    Draw imputations $\dot{Y}_j^t \sim f({Y_j}^{\text{mis}} | \dot{Y}_{-j}^t, R, \dot{\theta}_j^t)$
  end for
end for

We show how an iteration of this algorithm works in practice on a simulated dataset. Suppose we have data from six individuals about their hearing ability and sight ability (both a score out of 100) and their age. The data is summarised in Table 6.3a. We randomly draw Y_j^o for each covariate by randomly picking one of the observed values for each covariate. This step is shown in Table 6.3b.

Age	Hearing	Sight
12	80	NA
16	75	65
20	NA	70
NA	70	65
45	60	50
60	40	35

(a) Iteration 0: Initial Data

Age	Hearing	Sight
12	80	50
16	75	65
20	60	70
16	70	65
45	60	50
60	40	35

(b) Iteration 0: Starting Imputations

Table 6.3: Iteration 0: Initial Data (a) and Initial Data with Initial Imputations (b)

Let us assume very basic imputation models:

$$\text{Age} = \beta_a + \beta_b \text{Hearing} + \beta_c \text{Sight}$$
$$\text{Hearing} = \beta_i + \beta_k \text{Age} + \beta_l \text{Sight}$$
$$\text{Sight} = \beta_x + \beta_y \text{Age} + \beta_z \text{Hearing}$$

Now that the data is complete, Iteration 1 can be started for $j = 1, 2, 3$. For $j = 1$, we remove our imputed value (Age = 16) and fit the model to the remaining data. This gives us the equation $\text{Age} = 120.1094 - 0.9331\text{Height} - 0.5690\text{Sight}$. We predict the missing value and update it in Table 6.4a. We then do the same for $j = 2$, but fit the model using the updated data in Table 6.4a. This gives the model $\text{Hearing} = 89.9292 - 0.7446\text{Age} - 0.0467\text{Sight}$. We then predict the missing Hearing value and update the data to give Table 6.4b. We then do the same for $j = 3$, giving us the model $\text{Sight} = 41.1972 - 0.3987\text{Age} + 0.4493\text{Height}$. We then predict the missing Sight value and update the data to give Table 6.4c.

Age	Hearing	Sight
12	80	50
16	75	65
20	60	70
17.8074	70	65
45	60	50
60	40	35

(a) Iteration 1: Step j=1

Age	Hearing	Sight
12	80	50
16	75	65
20	71.7682	70
17.8074	70	65
45	60	50
60	40	35

(b) Iteration 1: Step j=2

Age	Hearing	Sight
12	80	72.3569
16	75	65
20	71.7682	70
17.8074	70	65
45	60	50
60	40	35

(c) Iteration 1: Step j=3

Table 6.4: Iteration 1 of the MICE Algorithm Applied to Simulated Data

This completes one iteration of the MICE algorithm. One must continue through iterations repeating the same process until all imputed values have converged.

6.3 Selection Models and Pattern Mixture Models

The following section discusses theory and examples mentioned in Longitudinal Data Analysis (Fitzmaurice et al, 2009)([4] p.417-422). However, we will expand on these ideas and give further explanation of results. We will now consider the case where we cannot ignore missingness and examine models for the joint distribution of $\mathbf{R}_i$ and $\mathbf{Y}_i$.

6.3.1 Selection Models

Selection models describe the joint distribution of $\mathbf{R}_i$ and $\mathbf{Y}_i$ through a combination of models for the marginal distribution of $\mathbf{Y}_i$ and the conditional distribution of $\mathbf{R}_i$ given $\mathbf{Y}_i$, in the form

$$f(\boldsymbol{R}_i, \boldsymbol{Y}_i|\boldsymbol{X}_i, \boldsymbol{\gamma}, \boldsymbol{\psi}) = f_{\boldsymbol{Y}}(\boldsymbol{Y}_i|\boldsymbol{X}_i, \boldsymbol{\gamma}) f_{\boldsymbol{R}|\boldsymbol{Y}}(\boldsymbol{R}_i|\boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{\psi}) \ .$$

Note that in the selection model, if we assume that the data is MAR and we can ignore missingness, we obtain the likelihood. Now let us consider a simple selection model for just two repeated measures with MNAR dropouts. We will then extend this to a more general case.

Say we have (Y_{i1}, Y_{i2}) are observed and $\boldsymbol{R}_i = 1$ for $i = 1, ..., m$, and Y_{i1} is observed, Y_{i2} is missing and $\boldsymbol{R}_i = 0$ for $i = m+1, ..., N$. Let $\mu = (\mu_1, \mu_2)$ with $\mu_j = E(Y_{ij})$ and $\boldsymbol{\Sigma}$ be the covariance matrix of $(Y_{i1}, Y_{i2}$. Let our normal selection model be as follows:

$$(Y_{i1}, Y_{i2}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \text{and} \quad (R_i|Y_{i1}, Y_{i2}) \sim \text{Bern}(P(\phi(Y_{i1}, Y_{i2})))$$

$$\text{logit}(P(\phi(Y_{i1}, Y_{i2}))) = \phi_0 + \phi_1 Y_{i1} + \phi_2 Y_{i2}$$

Where $\text{Bern}(\boldsymbol{\gamma})$ represents the Bernoulli distribution with probability γ. Does this make intuitive sense? We are using the Bernoulli distribution here as we are modelling binary response and using logistic regression to output the probability of observing the response.

We will now walk through how we find the likelihood for this selection model. We know that

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\phi}|\boldsymbol{R}, \boldsymbol{Y}^o) = \prod_{i=1}^{N} f(\boldsymbol{Y}_i, \boldsymbol{R}_i) = \prod_{i=1}^{N} f(\boldsymbol{Y}_i) f(\boldsymbol{R}_i|\boldsymbol{Y}_i) \ ,$$

with the right hand part of our equation coming from the law of total probability. Now we are going to partition our expression into $i = 1, ..., m$ and $i = m+1, ..., N$ such that

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\phi}|\boldsymbol{R}, \boldsymbol{Y}^o) = \prod_{i=1}^{m} f(\boldsymbol{Y}_i) f(\boldsymbol{R}_i|\boldsymbol{Y}_i) \times \prod_{i=m+1}^{N} f(\boldsymbol{Y}_i) f(\boldsymbol{R}_i|\boldsymbol{Y}_i) \ .$$

Now comes the more difficult step. For $i = 1, ..., m$, we have Y_{i1} and Y_{i2} fully observed, so we have $f(\boldsymbol{R}_i|Y_{i1}, Y_{12} = P(\phi(Y_{i1}, Y_{i2}))$. For $i = m+1, .., N$, we have Y_{i2} unobserved so have $1 - f(\boldsymbol{R}_i|Y_{i1}, Y_{12} = P(phi(Y_{i1}, Y_{i2}))$. Note for our likelihood we are also conditioning on $\boldsymbol{Y}^o$ so, in this product, we must integrate over our unobserved data. Hence we obtain our full final likelihood as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\phi}|\boldsymbol{R}, \boldsymbol{Y}^o) &= \prod_{i=1}^{m} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\boldsymbol{Y}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{Y}_i - \boldsymbol{\mu})) P(\phi(Y_{i1}, Y_{i2})) \\ &\times \prod_{i=m+1}^{N} \sigma_{11}^{-\frac{1}{2}} \int \exp(-\frac{1}{2}(\boldsymbol{Y}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{Y}_i - \boldsymbol{\mu}))(1 - P(\phi(Y_{i1}, Y_{i2}))) dY_{i2} \ . \end{aligned} \tag{6.1}$$

We have our σ_{11} in the second part of our expression since we have only observed our Y_{i1} hence we do not want our full $\boldsymbol{\Sigma}$. The large exponential term comes from the multivariate normal equation. Using this likelihood, we can now find our maximum likelihood estimates for our parameters however this requires an iterative method.

We can easily extend this model for n repeated measures $Y_i = (Y_{i1}, ..., Y_{in})$ with Y_{in} subject to dropout, with $R_i = 1$ if Y_{in} is observed and $R_i = 0$ if Y_{in} is unobserved. $\boldsymbol{X}_i$ is our fully observed matrix of covariates. The generalised model is

$$(\boldsymbol{Y}_i|\boldsymbol{X}_i) \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{\Sigma}) \ ,$$

$$(R_i|\boldsymbol{Y}_i, \boldsymbol{X}_i) \sim \text{Bern}(P(\phi(\boldsymbol{Y}_i, \boldsymbol{X}_i)) \ ,$$

$$\text{logit}(P(\phi(\boldsymbol{Y}_i, \boldsymbol{X}_i))) = \phi_0 + \phi_1 Y_{i1} + ... + \phi_n Y_{in} + \phi_{n+1}^{'}\boldsymbol{X}_i$$

We see our familiar $\boldsymbol{Y}_i|\boldsymbol{X}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{\Sigma})$ here as we have encountered in previous chapters. We add our $\phi_{n+1}^{'}\boldsymbol{X}_i$ term here as this allows our mechanism to depend on our covariates as well as our responses, in the hope of making our mechanism as accurate as possible.

Following this theory, we aim to create a package in `R` which contains functions which fit different selection models to longitudinal datasets. We have created a function to fit the first model discussed in this section, a normal model for two repeated measures with MNAR dropout, where the 2^{nd} measurement is subject to dropout. The objective of the function is to feed the data into the likelihood (as in Equation 6.1) and jointly maximise the likelihood to obtain a set of optimal distribution parameters. We maximised the log-likelihood as this makes it easier to work with in `R`.

To tackle the integral, we created a function with one input, x, that returned

$$f(x) = \exp(-\frac{1}{2}(\boldsymbol{Y_i} - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{Y_i} - \boldsymbol{\mu}))(1 - P(\phi(\boldsymbol{Y_i}))$$

where $\boldsymbol{Y_i} = (Y_{i1}, x)$. We use x to represent our missing Y_{i2} so then we can integrate over x to obtain the required expression. However, the `integrate` function, from the `stats` package [17], in `R` struggled to converge to a value so we used Monte Carlo Importance Sampling (MCIS) (Anderson, 2009) [24] to approximate the integral. We briefly explain the theory of MCIS below.

It is commonly known that if a random variable X has a probability density function (PDF) $f_X(x)$ which is defined on a set of values $\mathcal{X}$, the expectation of $g(X)$, where g is a function, is

$$\mathbb{E}(g(X)) = \int_{x\in\mathcal{X}} g(x)f_X(x)dx \ ,$$

if X is a continuous variable. Using the theory discussed by Anderson (2009) [24], one can obtain the Monte Carlo estimate of the expectation by taking a sample of n Xs $(x_1, ..., x_n)$ and taking the mean of $g(x)$ over the sample:

$$\tilde{g_n}(x) = \frac{1}{n}\sum_{i=1}^{n} g(x_i) \ .$$

One can show that this estimator is unbiased. MCIS follows a similar idea. Let $h(x)$ be the probability distribution function (PDF) of the random variable X taking values in $\mathcal{X}$ such that $\int_{x\in\mathcal{X}} h(x)dx = 1$. Then

$$\int_{x\in\mathcal{X}} g(x)dx = \int_{x\in\mathcal{X}} g(x)\frac{h(x)}{h(x)}dx = \int_{x\in\mathcal{X}} \frac{g(x)}{h(x)}h(x)dx = \mathbb{E}\bigg(\frac{g(X)}{h(X)}\bigg),$$

assuming $h(x) \neq 0$ for any $x \in \mathcal{X}$ for which $g(x) \neq 0$. The Monte Carlo estimator is as follows:

$$\tilde{g_n}^h(X) = \frac{1}{n}\sum_{i=1}^{n}\frac{g(X_i)}{h(X_i)} \quad \text{with} \quad X_i \sim h(x) \ .$$

The function into which we substitute sampled x_is into is of the form

$$\frac{1}{h(x_i)}\bigg(\exp\bigg(-\frac{1}{2}\begin{pmatrix}Y_{i1}-\mu_1\\ x_i-\mu_2\end{pmatrix}^T\Sigma^{-1}\begin{pmatrix}Y_{i1}-\mu_1\\ x_i-\mu_2\end{pmatrix}\bigg)\bigg(1-P(\phi(Y_{i1},x_i))\bigg) \ .$$

So, we are left to choose a PDF $h(x)$. Initially, it seemed sensible to use a standard normal distribution. However, we would then be sampling values close to 0: when plugging these values into our function, this would cause the expression inside the exponential function to be extremely small if μ_2 is far away from 0. Our `R` function struggled with this and often would not be able to converge. Hence, we slightly altered our approach by choosing $h(x)$ to be the PDF of a normal distribution with $X \sim N(\hat{\mu_2}, 1)$, where $\hat{\mu_2}$ is an estimate of μ_2. Hence, our MCIS estimator was of the form:

$$\tilde{g}_n^h(X) = \frac{1}{n}\sum_{i=1}^{n}\frac{\exp\bigg(-\frac{1}{2}\begin{pmatrix}Y_{i1}-\mu_1\\ x_i-\mu_2\end{pmatrix}^T\Sigma^{-1}\begin{pmatrix}Y_{i1}-\mu_1\\ x_i-\mu_2\end{pmatrix}\bigg)\bigg(1-P(\phi(Y_{i1},x_i))\bigg)}{\frac{1}{\sqrt{2\pi}}\exp\bigg(-\frac{1}{2}(x_i-\hat{\mu}_2)^2\bigg)} \quad \text{with} \quad X_i \sim N(\hat{\mu}_2, 1)$$

We can easily extend this function to n repeated measurements, where the n^{th} measurement is subject to dropout. The theory we use is a generalisation of the two-measurement case. The likelihood is of the form

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\phi}|\boldsymbol{R},\boldsymbol{Y}^o) &= \prod_{i=1}^{m}|\Sigma|^{-\frac{1}{2}}\exp(-\frac{1}{2}(\boldsymbol{Y_i}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i}-\boldsymbol{\mu}))P(\phi(Y_{i1},...,Y_{in}))\\ \times\prod_{i=m+1}^{N}|\boldsymbol{\Sigma}_{-n}|^{-\frac{1}{2}}&\int\exp(-\frac{1}{2}(\boldsymbol{Y_i}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i}-\boldsymbol{\mu}))(1-P(\phi(Y_{i1},...,Y_{in})))dY_{in} \ ,\end{aligned} \tag{6.2}$$

where $\boldsymbol{\Sigma}_{-n}$ is the covariance matrix without the n^{th} row and n^{th} column. Importance sampling works in a similar manner, however we are sampling for the n^{th} measurement so set the mean of the normal distribution as the sample mean of the observed n^{th} measurements. To show the performance of our function (see Code 6.1 in Appendix D), `selectmultMNAR`, we simulated 100 multivariate normal samples, with distribution parameters as such:

$$\boldsymbol{\mu} = (100, 200, 300, 400)^T \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix}20 & 15 & 10 & 5\\ 15 & 20 & 15 & 10\\ 10 & 15 & 20 & 15\\ 5 & 10 & 15 & 20\end{pmatrix} \ . \tag{6.3}$$

We then set 30 of the fourth measurement equal to `NA`, applied our function to the incomplete dataset and obtained the following results:

$$\hat{\boldsymbol{\mu}} = (100.475, 200.748, 302.178, 401.570) \qquad \hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 20.295 & 15.557 & 11.768 & 4.775 \\ 16.331 & 19.036 & 14.090 & 9.486 \\ 10.962 & 10.400 & 19.330 & 13.614 \\ 3.727 & 10.122 & 13.755 & 18.330 \end{pmatrix}$$

$$\hat{\boldsymbol{\phi}} = (-2.122, 0.379, -0.607, 0.188, -1.749)$$

An initial run of the function shows that it obtains estimates close to the true parameters. We want to compare the performance of our function to the performance of other methods used to calculate the mean and covariance of the dataset. The other methods we used consisted of performing different imputation techniques to the dataset and then fitting a linear mixed effects model with a random slope and intercept to the completed dataset. The four methods used were complete case analysis (since `lmer` is able to fit a LMEM to a dataset containing missing values), 'last observation carried forward', single mean imputation and multiple imputation.

We randomly sampled 30% of the individuals and set their final measurement equal to `NA`. We then applied the five methods to the dataset and calculated the mean and covariance parameter estimates. We performed this entire process 85 times, using a different random sample of individuals each time. For each method, we then calculated the average estimate for each of the mean and covariance parameter and calculated the absolute estimation error compared to the true parameters values.

	Average Absolute Estimation Error				
Method	μ_1	μ_2	μ_3	μ_4	Mean
LMEM - Complete Case Analysis	0.1546	0.0625	0.0295	0.1216	0.0921
LMEM - Last Observation Carried Forward	5.0720	2.6758	10.4236	18.1714	9.0857
LMEM - Single Mean Imputation	0.1329	0.0734	0.0138	0.0457	0.0665
LMEM - Multiple Imputation	0.1471	0.0663	0.0144	0.0951	0.0807
Selection Model Function	0.1603	0.1430	0.3123	0.8218	0.3594

Table 6.5: Average Absolute Estimation Errors of Each Model for the Calculation of the Mean Parameters

We used the absolute estimation error in Table 6.5 so that we can take averages. We see that our model performs better than LOCF and, on average, has an absolute estimation error for a mean parameter of 0.3594. The imputation methods, followed by fitting a LMEM, seem to give closer estimates for the mean parameters. We also compared the estimation errors for the covariance parameters. Figure 6.4 shows a violin plot for each method of the distribution of the estimation errors for the covariance parameters. Whilst we see that our function has the largest random of estimation errors, we see that it has the smallest interquartile range only behind the complete case analysis method. Our model seems to perform well compared to other methods and a key advantage is that our model copes well with non-linear data, however LMEMs will struggle to calculate parameter estimates for data which does not display a linear trend.

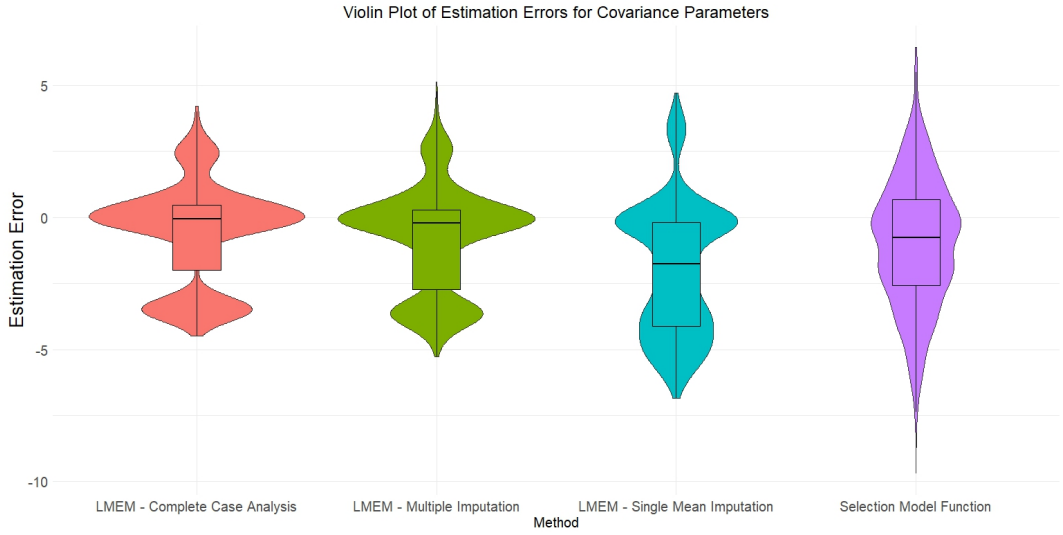


Figure 6.4: Violin Plot Demonstrating the Distribution and Interquartile Range of the Estimation Errors of the Covariance Parameters

However, the case where only the final measurement is subject to dropout is rare. It is more common for dropout to occur at any point in the longitudinal data study. Building on the theory that was introduced in *Longitudinal Data Analysis* (Fitzmaurice et al. 2009) [4], we created a function which fits a selection model to a dataset in which every measurement is subject to dropout and then attempted to extend the model to the case of intermittent missingness. We adapted the theory in the original MNAR selection model discussed at the beginning of this section. The basic concept is the same, however we now will have multivariate integrals in which we are integrating out over the variables which are missing. In our function, we perform multivariate importance sampling, which samples from multivariate distributions, to approximate these integrals. To perform multivariate importance sampling in `R` we adapted Python code introduced by Zhao (2021) [25]. If we have N individuals measured at k time points and we assume that an individual may drop out at any point, we order the individuals in terms of their time of dropout. We put those who did not drop out at all first, then those who dropped out at the n^{th} measurement, then those who dropped out at the $(n-1)^{\text{th}}$ measurement etc. until we have ordered all individuals.

We demonstrate the method used to calculate the likelihood function for this scenario. Consider we have 8 individuals who are measured over 5 different time points. 5 of the individuals dropped out of the trial at some point, giving the following dataset, as shown in Table 6.6a. Our function begins by ordering the dataset in terms of missingness and relabling the individuals, as shown in Table 6.6b.

	Measurement Value Y_{ij}				
Individual i	Y_{i1}	Y_{i2}	Y_{i3}	Y_{i4}	Y_{i5}
1	a_1	a_2	NA	NA	NA
2	b_1	b_2	b_3	b_4	b_5
3	c_1	c_2	c_3	c_4	NA
4	d_1	d_2	d_3	d_4	d_5
5	e_1	e_2	e_3	e_4	NA
6	f_1	f_2	f_3	NA	NA
7	g_1	g_2	g_3	g_4	g_5
8	h_1	h_2	h_3	NA	NA

(a) Step 1: Original Dataset

	Measurement Value $Y_{i'j}$				
Individual i'	$Y_{i'1}$	$Y_{i'2}$	$Y_{i'3}$	$Y_{i'4}$	$Y_{i'5}$
1	b_1	b_2	b_3	b_4	b_5
2	d_1	d_2	d_3	d_4	d_5
3	g_1	g_2	g_3	g_4	g_5
4	c_1	c_2	c_3	c_4	NA
5	e_1	e_2	e_3	e_4	NA
6	f_1	f_2	f_3	NA	NA
7	h_1	h_2	h_3	NA	NA
8	a_1	a_2	NA	NA	NA

(b) Step 2: Rearrange dataset in order of missingness and relabel individuals to correspond with row number

Table 6.6: Demonstration of how `selectmultMNARevery` Prepares the Input Data

Table 6.6 demonstrates how `selectmultMNARevery` permutes the dataset to make the calculation of the likelihood function easier. Continuing with the example, we would calculate the likelihood as such:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\phi}|\boldsymbol{R},\boldsymbol{Y}^{\text{o}}) = &\prod_{i'=1}^{3}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp(-\frac{1}{2}(\boldsymbol{Y_i'}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i'}-\boldsymbol{\mu}))P(\phi(\boldsymbol{Y}_{i'}))\\ &\times\prod_{i'=4}^{5}|\boldsymbol{\Sigma}_{-5}|^{-\frac{1}{2}}\int_{\mathbb{R}}\exp(-\frac{1}{2}(\boldsymbol{Y_i'}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i'}-\boldsymbol{\mu}))(1-P(\phi(\boldsymbol{Y}_{i'})))dY_{i'5}\\ &\times\prod_{i'=6}^{7}|\boldsymbol{\Sigma}_{-(4,5)}|^{-\frac{1}{2}}\int_{\mathbb{R}^2}\exp(-\frac{1}{2}(\boldsymbol{Y_i'}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i'}-\boldsymbol{\mu}))(1-P(\phi(\boldsymbol{Y}_{i'})))d(Y_{i'4},Y_{i'5})\\ &\times\prod_{i'=8}^{8}|\boldsymbol{\Sigma}_{-(3,4,5)}|^{-\frac{1}{2}}\int_{\mathbb{R}^3}\exp(-\frac{1}{2}(\boldsymbol{Y_i'}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y_i'}-\boldsymbol{\mu}))(1-P(\phi(\boldsymbol{Y}_{i'})))d(Y_{i'3},Y_{i'4},Y_{i'5})\ .\end{aligned} \tag{6.4}$$

Note that $\boldsymbol{\Sigma}_{-(a,b)}$ denotes the covariance matrix with the a^{th} and b^{th} column and row removed (so if we orginally had a $n\times n$ matrix we would now have a $(n-2)\times(n-2)$ matrix). The example likelihood (6.4) demonstrates how we calculate the likelihood of the joint distribution of a dataset in which dropout can occur at any time point. Our function, `selectmultMNARevery` (see Code 6.2 in Appendix D), uses this method to calculate the log-likelihood, using multivariate importance sampling to approximate the integrals, of which we then use the `optim` function, from the `stats` package [17], to find parameter estimates. Using the dataset previously used with parameters (6.3), we randomly selected 30 of the individuals, of whom we set 10 of them to drop out at the second measurement, 10 of them to drop out at the third measurement and 10 of them to drop out at the fourth (and final) measurement. We obtained the following parameter estimates:

$$\hat{\boldsymbol{\mu}} = (100.979, 198.638, 302.028, 400.109) \qquad \hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 18.779 & 14.436 & 11.345 & 5.923 \\ 14.066 & 20.311 & 13.901 & 8.864 \\ 9.933 & 15.447 & 20.838 & 17.792 \\ 5.537 & 10.925 & 14.789 & 21.583 \end{pmatrix}$$

$$\hat{\boldsymbol{\phi}} = (-0.597, -1.372, 0.841, 0.100, -1.964)$$

We see that our model outputs mean and covariance parameter estimates close to the true values, and we do not see a significant drop in performance compared to first case we considered in which dropout only occurred at the last measurement. This suggests the form of the likelihood we proposed allows us to correctly model the joint distribution. We also created a function, `selectmultMNARany` (Code 6.3 in Appendix D), to fit a selection model to a dataset which contains intermittent missingness, in which we calculate the likelihood using a similar method.

6.3.2 Pattern Mixture Models

Pattern mixture models describe the joint distribution of $\mathbf{R}_i$ and $\mathbf{Y}_i$ through combination of the marginal distribution of $\mathbf{R}_i$ and the conditional distribution of $\mathbf{Y}_i$ given $\mathbf{R}_i$.

$$f(\boldsymbol{R}_i, \boldsymbol{Y}_i | \boldsymbol{X}_i, \boldsymbol{\gamma}, \boldsymbol{\psi}) = f_{\boldsymbol{R}}(\boldsymbol{R}_i | \boldsymbol{X}_i, \boldsymbol{\gamma}) f_{\boldsymbol{Y}|\boldsymbol{R}}(\boldsymbol{Y}_i | \boldsymbol{X}_i, \boldsymbol{R}_i, \boldsymbol{\psi})$$

Let us consider the same scenario we had earlier, with two repeated measures with MNAR dropouts. Here we have a pattern mixture model of the form

$$(Y_{i1}, Y_{i2} | R_i = k) \sim N(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)})$$

$$R_i \sim \text{Bern}(\gamma)$$

We will now see where the term 'pattern mixture' comes from. If we calculate the mean of (Y_{i1}, Y_{i2}), using the total law of probability we get

$$E[\boldsymbol{Y}] = E[\boldsymbol{Y} | R_i = 0] P(R_i = 0) + E[\boldsymbol{Y} | R_i = 1] P(R_i = 1) \ ,$$

$$E[\boldsymbol{Y}] = \boldsymbol{\mu}^{(0)}(1 - \gamma) + \boldsymbol{\mu}^{(1)}\gamma \ ,$$

where we have just used the properties of our distribution to go from the first to the second line. So we can see that our marginal mean is a mixture of the two means from the different distributions.

6.3.3 Sensitivity Analysis for the Models

Despite the models discussed in Section 6.3.1 and Section 6.3.2 seeming to be robust ways to model missing data, we are still left with an important question: given a dataset, how do we decide whether missingness is MAR or MNAR? It is formally shown by Fitzmaurice et al. (2009) ([4], p.520-522) that every MNAR model fit to an observed dataset can be reproduced by an MAR model. Although an MAR and MNAR model may have the exact same fit for the observed data, they often will have vastly different predictions for the unobserved data. This issue highlights the need for sensitivity analysis, which is defined as a tool in which several models are fitted simultaneously to the data and the results are analysed thereafter. The more similar estimates are across all models, the higher the degree of confidence one would have that the estimate is robust.

Theorem 6.1: Every fit to the observed data, obtained from fitting an MNAR model to a set of incomplete data, is exactly reproducible from an MAR model.

Proof. Theorem 6.1 is introduced by Fitzmaurice et al. (2009) ([4], p520-522) with a brief proof which we look to expand on. The first step of the proof is to consider an MNAR model fitted to the observed set of data. We assume that the model is of a selection model form. The likelihood is as such:

$$\begin{aligned} L &= \prod_i \int f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m, \boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \\ &= \prod_i \int f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m|\boldsymbol{\theta}) f(\boldsymbol{r}_i|\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \ . \end{aligned}$$

We have gone from the first to the second line simply by using the laws of conditional probability. One can express the density in a different form using laws from conditional probability:

$$\begin{aligned} f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m, \boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) &= f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m|\boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) \\ &= f(\boldsymbol{y}_i{}^o|\boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) \ . \end{aligned}$$

Again laws of conditional probability have been used to rearrange the expression into a pattern mixture model format. We are only considering the observed data in this theory, so one can note that the middle term in the second line of the expression above, $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi})$, is not identified from the observed data due to it depending on $\boldsymbol{y}_i{}^m$. We can replace this factor with the appropriate MAR factor, using the lemma that in the pattern mixture model framework, the missing data mechanism is MAR if and only if $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{r}_i, \boldsymbol{\theta}) = f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta})$. Fitzmaurice et al. (2009) ([4] p.522) uses the lemma to replace $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi})$ with $f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta}, \boldsymbol{\psi})$. These steps can then be put together to obtain the following results, showing that every fit to the observed data obtained from fitting an MNAR model to a set of incomplete data, is exactly reproducible from an MAR model.

$$\begin{aligned} L &= \prod_i \int f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m, \boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \\ &= \prod_i \int f(\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m|\boldsymbol{\theta}) f(\boldsymbol{r}_i|\boldsymbol{y}_i{}^o, \boldsymbol{y}_i{}^m, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \\ &= \prod_i \int f(\boldsymbol{y}_i{}^o|\boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \\ &= \prod_i \int f(\boldsymbol{y}_i{}^o|\boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{y}_i{}^m|\boldsymbol{y}_i{}^o, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) d\boldsymbol{y}_i{}^m \\ &= \prod_i f(\boldsymbol{y}_i{}^o|\boldsymbol{r}_i, \boldsymbol{\theta}, \boldsymbol{\psi}) f(\boldsymbol{r}_i|\boldsymbol{\theta}, \boldsymbol{\psi}) \ . \end{aligned}$$

□

This shows it is extremely difficult to differentiate between MNAR models and MAR models without making unverifiable assumptions. This motivates the need for sensitivity analysis so we can make informed decisions about which missing data mechanism we should assume. When multiple models are fitted to the data, one can then compare performance using different metrics such as prediction error.

7 Summary and Conclusions

The initial problem highlighted in this report was the multiple issues one faces when attempting to model longitudinal data. Namely, simple models do not account for variability between measurements of an individual and variability between individuals. We have explained that sources of variability are not always obvious and often unknown. A key focus of this report has been a class of models which acknowledge the random, potentially unknown, sources of variability in a dataset: linear and non-linear mixed effects models.

Chapter 3 discussed linear mixed effects models as a potential solution to our problem. Linear mixed effects models are of the general form

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon} \ ,$$

where $\boldsymbol{Y}$ is the vector of responses, $\boldsymbol{X}$ and $\boldsymbol{Z}$ are design matrices, $\boldsymbol{\beta}$ is a vector of fixed unknown regression coefficients, $\boldsymbol{u}$ is a vector of random effects and $\boldsymbol{\epsilon}$ is a vector of residual errors. The inclusion of random effects aim to account for the random sources of variability between measurements and individuals. We showed that the response in a linear mixed effects model has a multivariate normal distribution as such:

$$\boldsymbol{Y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{Z}\boldsymbol{Q}^T\boldsymbol{Z} + \sigma^2\boldsymbol{I}_N) \ ,$$

where $\boldsymbol{Q}$ is the covariance matrix of the random effects. We displayed the effectiveness of the linear mixed effects model by fitting a random intercept model and a random slope and intercept model to the `ChickWeight` dataset, one of two datasets introduced in Chapter 2. We saw that the random slope and intercept model managed to capture the individual trends in the data extremely well, importantly much better than a linear regression model. The remainder of Chapter 3 (Section 3.3) discussed multiple model selection criteria, the AIC and BIC, and some alternate versions of the classical criteria which are more appropriate for linear mixed effects models. The 'Hybrid BIC' was introduced by Chen and González [9] as a more robust criterion as it relaxes the assumption of independent and identically distributed measurements.

Chapter 4 focused on deeper theory related to linear mixed effects models. Section 4.1 discussed different methods related to inference for the fixed effects in an LMEM. The method for estimating $\boldsymbol{\beta}$ depends on how much is known about the variance components, $\boldsymbol{\gamma} = \{\boldsymbol{Q}, \sigma^2\}$. It is rare for the variance components to be fully known, so we highlighted the REML method (Section 4.1.3) as a method to find a likelihood that can be jointly maximised which does not depend directly on $\boldsymbol{\beta}$. It was shown that the likelihood has the form

$$L_{REML} = \frac{1}{(2\pi)^{n-p}}|\Sigma|^{-\frac{1}{2}}|X^T\Sigma^{-1}X|^{-\frac{1}{2}}|X^TX|^{\frac{1}{2}}\exp\left(-\frac{1}{2}(Y - X\hat{\beta})^T\Sigma^{-1}(Y - X\hat{\beta})\right) \ .$$

The REML method is used in statistical software, such as `R`, to calculate the parameter estimates of a linear mixed effects model. One assumption that could be perceived as naive in linear mixed effects models is the assumption of normally distributed random effects and residual errors. Section 4.1.4 focused on theory which allowed us to avoid this assumption by using the minimum norm quadratic unbiased estimator (MINQUE) as an estimator for σ^2. Section 4.3 discussed a criterion which allows us to state whether the random effects in a linear mixed effects model are relevant. The idea

behind this is finding a criterion which ensures that the maximum likelihood estimator of $\boldsymbol{Q}$, the covariance matrix of the random effects, is non-zero. This in-turn tells us that the random effects are relevant in our model. The final section of Chapter 4 highlighted a version of the EM algorithm, as introduced in [8]. This algorithm aims to find the maximum likelihood estimator of model parameters and the version we include iteratively updates the variance components, σ^2 and $\boldsymbol{D}$, until they converge.

In Chapter 5, we focused on non-linear mixed effects models. In practice, it is common for the response measurement in a longitudinal dataset to change in a non-linear fashion as time increases. A key issue with these models is that we cannot obtain a closed-form expression for the likelihood, so one must use approximation methods. We followed the method discussed by Chiquet [16] to fit a non-linear mixed effects model to the `Theophylline` dataset using the `saemix` package. This packages fits a non-linear mixed effects model by performing the stochastic approximation expectation maximisation (SAEM) algorithm. We saw that the fitted model performed well and seemed to capture the majority of the trends within the data.

In the final of the report, Chapter 6, we focused on missing data. It is extremely common for longitudinal datasets to contain missing values, where either an individual is missing certain measurements or they have dropped out of the study altogether, meaning from that point onwards all of their measurements are missing. Section 6.1.3 discussed different missing data mechanisms, missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR). These mechanisms differed in what the missingness depended on. A key topic in Chapter 6 was imputation, the idea of 'filling in' the missing data. Multiple single imputation methods were highlighted, but ultimately shown to be rather unrealistic in practice: for example, 'Last Observation Carried Forward' (LOCF) would be a naive assumption if we knew that the response variable tended to increase over time. This motivated the need for a more robust imputation technique. Multiple imputation generates multiple complete imputed datasets and then performs analysis before pooling conclusions together to obtain a single imputed value for each missing data point. A key area of focus in Chapter 6, and indeed in the report as a whole, was the contents of Section 6.3. This focused on selection and pattern mixture models, which jointly model the response variable $\boldsymbol{Y}_i$ and the missingness, $\boldsymbol{R}_i$. We included the description and showcased the performance of the `R` function we developed to fit a selection model to an n dimensional longitudinal dataset containing dropout.

A potential area for research would be to develop theory which considers a joint model between a non-linear mixed effects model and the missingness within a datset, in a similar method to the selection models mentioned in Section 6.3.1. One could also further explore methods to estimate parameters in a non-linear mixed effects model which tackle the issue of the likelihood being unable to be expressed in a closed form. Going forward, we plan to continue work on the `R` package we created and highlighted in Section 6.3.1 in the hopes of making it user friendly so others can use it to fit a selection model. A key focus is reducing the time taken for our functions to produce an output, with the aim of users being able to apply the function to high-dimensional datasets within `RStudio` in a reasonable time. An immediate next step is to write a vignette for our package with the aim of submitting our work to journals for publication by the start of Autumn 2024.

Bibliography

[1] National Institute of Health: Office of Scientific Policy. "The Framingham Heart Study: Laying the Foundation for Preventative Health Care". URL: https://www.framinghamheartstudy.org/files/2021/07/FHS-Laying-the-Foundation-from-NIH.pdf.

[2] N. Laird and J. Ware. "Random Effects Models for Longitudinal Data". In: *Biometrics* 38.4 (1982), pp. 963–974.

[3] G. Verbeke and G. Molenberghs. *Linear Mixed Models for Longitudinal Data*. Springer, 2000.

[4] G. Fitzmaurice et al. *Longitudinal Data Analysis*. Chapman Hall/CRC, 2009.

[5] B. Douglas et al. "Fitting Linear Mixed-Effects Models Using lme4". In: *Journal of Statistical Software* 67.1 (2015), pp. 1–48. DOI: 10.18637/jss.v067.i01.

[6] Daniel Lüdecke. *sjPlot: Data Visualization for Statistics in Social Science*. R package version 2.8.15. 2023. URL: https://CRAN.R-project.org/package=sjPlot.

[7] H. Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. DOI: 10.1109/TAC.1974.1100705.

[8] E. Demidenko. *Mixed Models - Theory and Applications with R: Second Edition*. Wiley, 2013.

[9] N. Shen and B. González. "Bayesian Information Criterion for Linear Mixed-Effects Models". In: *arXiv* 2104.14725 (2021).

[10] J. Einbeck. "Advanced Statistical Modelling - Section 15: Linear Mixed Models". Lecture Note - Durham University.

[11] L.R. LaMotte. "A direct derivation of the REML likelihood function". In: *Statistical Papers* 48 (2007), pp. 321–327.

[12] The Free Encyclopedia Wikipedia. "Jacobi's Formula". URL: https://en.wikipedia.org/wiki/Jacobi%27s_formula.

[13] D.A. Harville. "Bayesian Inference for Variance Components Using Only Error Contrasts". In: *Biometrika* 61.2 (1974), pp. 383–385.

[14] C.R. Rao. *Linear Statistical Inference and its Applications: Second Edition*. Wiley, 1973. DOI: 10.1002/9780470316436.

[15] A. Ruckstuhl. "Introduction into Nonlinear Regression". Lecture Note, IDP Institute of Data Analysis and Process Design. 2020. URL: https://ethz.ch/content/dam/ethz/special-interest/math/statistics/sfs/Education/Advanced%20Studies%20in%20Applied%20Statistics/course-material-1921/Robust/nlreg20E.pdf.

[16] J. Chiquet. "MAP566 - Stats in Action: Nonlinear Mixed Effects Models". GitHub Page. URL: https://jchiquet.github.io/MAP566/docs/mixed-models/map566-lecture-nonlinear-mixed-model.html.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2023. URL: https://www.R-project.org/.

[18] Emmanuelle Comets, Audrey Lavenu, and Marc Lavielle. "Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm". In: *Journal of Statistical Software* 80.3 (2017), pp. 1–41. DOI: 10.18637/jss.v080.i03.

[19] A. Dempster and D.B Rubin. “Incomplete Data in Sample Surveys, Vol II: Theory and Annotated Bibliography”. In: *New York: Academic Press* (1983).

[20] D.B Rubin. “Multiple Imputations in Sample Surveys - A Phenomenological Bayesian Approach to Nonresponse”. In: *JSM Proceeding, Survey Research Methods Section. American Statistical Association* (1978), pp. 20–29. URL: http://www.asasrms.org/Proceedings/papers/1978_004.pdf.

[21] G. Molenberghs and G. Verbeke. *Models for Discrete Longitudinal Data*. Springer, 2005.

[22] E. Ogundimu. “Advanced Statistical Modelling - Missing Data Analysis”. Lecture Note, Durham University 2023.

[23] S. van Buuren and K. Groothuis-Oudshoorn. “mice: Multivariate Imputation by Chained Equations in R”. In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03.

[24] E.C. Anderson. “Monte Carlo Methods and Importance Sampling”. Lecture Note, Berkeley 1999. URL: https://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf.

[25] B. Zhao. “Monte Carlo Integration in Python Over Univariate and Multivariate Functions”. GitHub page. 2021. URL: https://boyangzhao.github.io/posts/monte-carlo-integration.

The following appendices contain extracts of code used to produce results in the main body of the report. We include an appendix for each chapter. We will include explanation of the code where necessary through the use of comments within the code. We assume that any built-in `R` packages are already installed and called in the environment. This code is also available on my **GitHub** page.

A - Chapter 2 Code

Figure 2.1 Code

```
library(ggplot2)
library(gridExtra)
DietA = ChickWeight[1:220,]
DietB = ChickWeight[221:340,]
DietC = ChickWeight[341:460,]
DietD = ChickWeight[461:578,]

par(mfrow=c(2,2))
p1 =ggplot(data=DietA, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(title='Diet A', x = 'Time (Days)', y = 'Weight (g)')+
  theme(legend.position='none', plot.title=element_text(size=15, hjust=0.5),
  axis.title = element_text(size=15), axis.text = element_text(size=15))
p2 =ggplot(data=DietB, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(title='Diet B', x = 'Time (Days)', y = 'Weight (g)')+
  theme(legend.position='none',plot.title=element_text(size=15, hjust=0.5),
  axis.title = element_text(size=15), axis.text = element_text(size=15))
p3=ggplot(data=DietC, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(title='Diet C', x = 'Time (Days)', y = 'Weight (g)')+
  theme(legend.position='none',plot.title=element_text(size=15, hjust=0.5),
  axis.title = element_text(size=15), axis.text = element_text(size=15))
p4=ggplot(data=DietD, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(title='Diet D', x = 'Time (Days)', y = 'Weight (g)')+
  theme(legend.position='none',plot.title=element_text(size=15, hjust=0.5),
  axis.title = element_text(size=15), axis.text = element_text(size=15))
#Call this comment to plot results
#gridExtra::grid.arrange(p1,p2,p3,p4)
```

Table 2.2 Code

```r
CW= ChickWeight[-c(166:182, 195:196, 85:95, 497:506),]
###here we omit all chicks who dropped out
finalweight = CW[as.vector(which(CW$Time==21)),]

##calculating mean final weights
for(i in 1:4){
    print(mean(finalweight[as.vector(which(finalweight$Diet==i)),1]))
}

# Calculating maximum weights
for (i in 1:4){
    print(max(finalweight[as.vector(which(finalweight$Diet==i)),1]))
}

# Calculuating minimum weights
for (i in 1:4){
    print(min(finalweight[as.vector(which(finalweight$Diet==i)),1]) )
}

# Calculating mean change between occasions (for all chicks)

r = matrix(NA, nrow=50, ncol=2)
for (i in 1:50){
  t = ChickWeight[as.vector(which(ChickWeight$Chick==i)),]
  diet = (t$Diet)[1]
  j=1
  n = length(t[,1])
  x = rep(0, n-1)
  if (n==12){
    while (j<n){
      x[j] = t[j+1,1] - t[j,1]
      j = j+1
    }
    s = x[11]
    x[11] = 2*s
    r[i,1]=mean(x)
    r[i,2]=diet
  }
  else{
    while(j<n){
      x[j] = t[j+1,1]-t[j,1]
      j=j+1
    }
    r[i,1]=mean(x)
    r[i,2]=diet
```

```
  }
}



# Here we have multiplied our last step if we have full growth by 2 since we measure
# day 20 and 21 so half the distance than the rest of them.

mean(r[1:20,1])
mean(r[21:30,1])
mean(r[31:40,1])
mean(r[41:50,1])
```

Figure 2.2 Code

```
require(ggplot2)
Theophyllin = read.csv('Theophyllin.csv')
Theophyllin = Theophyllin[1:120,]
Theo = Theophyllin
Theo$ID = as.factor(Theo$ID)
names(Theo) = c('id', 'time', 'concentration', 'weight')

plot = ggplot(data=Theo, aes(x=time, y=concentration, col=id))+
  geom_line(size=1.2, alpha=0.8)+
  geom_point(size=1)+
  labs(x='Time (Hours)', y='Concentration (mg/L)', col='ID')+
  theme(axis.title = element_text(size=25), axis.text = element_text(size=25),
  legend.key.height = unit(1.4, 'cm'), legend.title=element_text(size=25),
  legend.text = element_text(size=15))

# Call plot to actually plot the data
```

Table 2.2 Code

```
Theophyllin = read.csv('Theophyllin.csv')
Theophyllin = Theophyllin[1:120,]
Theo = Theophyllin
Theo$ID = as.factor(Theo$ID)

names(Theo) = c('id', 'time', 'concentration', 'weight')
serum = matrix(NA, nrow=1, ncol=12)
for (i in 1:12){
```

```
  Theosplit = Theo[which(Theo$id==i),]
  serum[i] = max(Theosplit[,3])
}
min(serum)
max(serum)

timev = matrix(NA, nrow=1, ncol=12)
for (i in 1:12){
  Theosplit = Theo[which(Theo$id==i),]
  mw = max(Theosplit[,3])
  timev[i] = Theosplit[which(Theosplit[,3]==mw),2]
}
mean(timev)

difr = matrix(NA, nrow=1, ncol=12)
for(i in 1:12){
  Theosplit = Theo[which(Theo$id==i),]
  difr[i] = Theosplit[1,3] - Theosplit[10,3]
}
difr
```

B - Chapter 3 Code

Figure 3.1 Code

```
library(ggplot2)
ChickWeight$Chick = as.factor(ChickWeight$Chick)
#below code plots Figure 3.1
ggplot(data=ChickWeight, aes(x=Time, y=weight, col=Chick))+
  geom_point(size=1.2, alpha=0.8)+
  geom_line(aes(y=predict(lm(weight~Time, data=ChickWeight)), x=Time),col='red',
  size=1, alpha=0.8)+
  labs(x='Time (Days)', y='Weight (g)')+
  theme(legend.position='theme',
  axis.title = element_text(size=25), axis.text = element_text(size=25))
```

Code 3.2

```
library(lme4)
# Random intercept model fitted to ChickWeight
fitcwri = lmer(weight ~ Time + Diet + (1|Chick), data=ChickWeight)

# Obtaining all statistics
```

```
summary(fitcwri)

# Plotting the model

ggplot(data=ChickWeight, aes(x=Time, y=weight, col=Chick))+
  geom_point(size=1.2, alpha=0.8)+
  geom_line(aes(y=predict(fitcwri, ChickWeight), x=Time), size=1, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col = 'Chick')+
  theme(axis.title=element_text(size=25), axis.text=element_text(size=25),
  legend.key.height = unit(1, 'cm'), legend.title = element_text(size=25))
```

Code 3.3

```
library(lme4)
# Random intercept and slope model fitted to ChickWeight
fitcwrs = lmer(weight ~ Time + Diet + (Time|Chick), data=ChickWeight)

# Obtaining model statistics
summary(fitcwrs)

# Plotting the model

ggplot(data=ChickWeight, aes(x=Time, y=weight, col=Chick))+
  geom_point(size=1.2, alpha=0.8)+
  geom_line(aes(y=predict(fitcwrs, ChickWeight), x=Time), size=1, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col = 'Chick')+
  theme(axis.title=element_text(size=25), axis.text=element_text(size=25),
  legend.key.height = unit(1, 'cm'), legend.title = element_text(size=25))
```

Figure 3.4 Code

```
require(sjPlot)
library(lme4)
install.packages('TMB')
fitcwri = lmer(weight ~ Time + Diet + (1|Chick), data=ChickWeight)
fitcwrs = lmer(weight ~ Time + Diet + (Time|Chick), data=ChickWeight)
# Setting aesthetic requirements

set_theme(title.size = 1.4,axis.title.size = 1.2, axis.textsize.x = 1.2, title.align = 'center')

# Creating the plots
r1 = plot_model(fitcwri, type='re', transform=NULL, show.p=TRUE,
```

```
title='Random Effects for Random Intercept Model')
r2 = plot_model(fitcwrs, type='re', transform=NULL, show.p=TRUE,
title='Random Effects for Random Intercept and Slope Model')

# Print the plot
gridExtra::grid.arrange(r1,r2, nrow=1)
```

Table 3.1 Code

```
library(lme4)
library(nlme)

# Function to calculate 'Improved BIC'

BICimp = function(model, p){
  corr = cov2cor(vcov(model))
  ne = sum(solve(corr))
  loglik = summary(model)$logLik
  return((-2*loglik) + (p*log(ne)))
}

# Function to calculate 'Hybrid BIC'

BIChybrid = function(model, r, f){
  loglik = summary(model)$logLik
  N = dim(unique(summary(model)$groups))[1]
  n = dim(summary(model)$groups)[1]
  return((-2*loglik)+(r*log(N))+(f*log(n)))
}

# Model 1
fit.t = lme(fixed = weight ~ Time, data=ChickWeight, random=~1|Chick )

# BICs for Model 1
bic1 = BIChybrid(fit.t, 2,3)   ## 5646.301
bicimp1 = BIC(fit.t) ## 5644.822
bichyb1 = BICimp(fit.t, 5) ## 5625.605

# Model 2
fit.td = lme(fixed = weight ~ Time + Diet, data=ChickWeight, random=~1|Chick )

# BICs for Model 2
bic2 = BIC(fit.td) ## 5628.46
bicimp2 = BICimp(fit.td, 8) ## 5607.036
```

```
bichyb2 = BIChybrid(fit.td, 2, 6) ##5629.986

# Model 3
fit.rst = lme(fixed = weight~ Time, data=ChickWeight, random=~Time|Chick)

#BICs for Model 3
bic3 = BIC(fit.rst) ##4865.636
bicimp3 = BICimp(fit.rst, 7) ##4846.7
bichyb3 = BIChybrid(fit.rst, 4,3)  ##4862.226

# Model 4
fit.rstd = lme(fixed= weight~Time + Diet, data=ChickWeight, random=~Time|Chick)

#BICs for Model 4
bic4 = BIC(fit.rstd)  ## 4860.912
bicimp4 = BICimp(fit.rstd, 10) ## 4841.654
bichyb4 = BIChybrid(fit.rstd, 4,6) ##4857.559
```

C - Chapter 5 Code

Figure 5.1 Code

```
library(ggplot2)

Theophyllin = read.csv('Theophyllin.csv')
Theophyllin = Theophyllin[1:120,]
Theo = Theophyllin
Theo$ID = as.factor(Theo$ID)
names(Theo) = c('id', 'time', 'concentration', 'weight')

f1 = function(psi, t){
  D  = 320; ka = psi[1]; V  = psi[2]; ke = psi[3]
  f  = D*ka/V/(ka-ke)*(exp(-ke*t)-exp(-ka*t))
  return(f)
}

dplot = data.frame(time=seq(0,40,by=0.2))
model_all = nls(concentration ~ f1(psi, time), start=list(psi=c(ka=1, V=40, ke=0.1)), data=Theo)
dplot$predall = predict(model_all, newdata=dplot)

subjectallplot = ggplot(data=Theo, aes(x=time, y=concentration, col=id))+
  geom_point(size=2, alpha=0.8)+
  labs(x='Time (hours)', y='Concentration (mg/L)', color='ID')+
  theme(axis.title=element_text(size=25), axis.text = element_text(size=20),
```

```
  legend.key.height = unit(1.3,'cm'), legend.text = element_text(size=20),
  legend.title= element_text(size=20))

# Print the plot in Figure 5.1
subjectallplot + geom_line(data=dplot, aes(x=time, y=predall), col='green',linewidth=1.5)
```

Figure 5.2 Code

```
plottt = function(i){
  dum = subset(Theo, id==i)
  model = nls(concentration~f1(psi, time), start=list(psi=c(ka=1, V=40, ke=0.1)), data=dum)
  dplot$pred = predict(model, newdata=dplot)
  gg = ggplot(data=dum, aes(x=time, y=concentration)) +
    geom_point(size=1.5, alpha=0.8)+labs(x = 'Time (hours)', y='Concentration (mg/L)',
    title=paste('Individual',i))+
    theme(plot.title = element_text(size=15))
  print(gg + geom_line(data=dplot, aes(x=time, y=pred), col='green', linewidth=1))
}

# Plot Figure 5.2

plots = lapply(unique(Theo$id), plottt)
gridExtra::grid.arrange(grobs = plots, ncol=3)
```

Table 5.1 Code

```
install.packages('saemix')
library(tidyverse)
library(saemix)
saemix_theo = saemixData(name.data = Theo,
                         name.group = 'id',
                         name.predictors = 'time',
                         name.response = 'concentration')


model1_nlme = function(psi,id,x) {
  D   = 320
  t   = x[,1]
  ka  = psi[id,1]
  V   = psi[id,2]
  ke  = psi[id,3]
  fpred = D*ka/(V*(ka-ke))*(exp(-ke*t)-exp(-ka*t))
```

```
  return(fpred)
}


saemix_model = saemixModel(model=model1_nlme,
                           psi0 = c(ka=1, V=20, ke=0.5))


saemix_options = list(map=TRUE, fim=TRUE, ll.is=FALSE, displayProgress=TRUE, save=FALSE,
seed=123123)
saemix_fit1 = saemix(saemix_model, saemix_theo, saemix_options)
saemix_fit1@results
```

Figure 5.3 Code

```
psi = psi(saemix_fit1)
psi
saemix_fit = saemix.predict(saemix_fit1)
saemix_fit
saemix.plot.fits(saemix_fit1, xlab='Time (hours)', ylab='Concentration (mg / L)',
cex.lab=1.4, cex.main=1.6)
```

Table 5.2 Code

```
mean(unique(Theo$weight))

Theo$w695 = Theo$weight - 69.5
theo_data1 = saemixData(name.data = Theo,
                        name.group = c('id'),
                        name.predictors = c('time'),
                        name.response = c('concentration'),
                        name.covariates=c('w695'))

saemix_modelw = saemixModel(model = model1_nlme,
                            psi0 = c(ka=1, V=20, ke=0.5),
                            transform.par = c(1,0,1),
                            covariate.model = c(0,1,0))
saemix_fitw = saemix(saemix_modelw, theo_data1, saemix_options)
saemix_fitw@results
```

D - Chapter 6 Code

Figure 6.1 Code

```
x = seq(0,1,0.05)

## After trying the data we saw that it was sufficient just to go up to taking a maximum
# of 60% of data out of ChickWeight
## STEP 1 - Make an 1x13 empty matrix to put our RSS values in
rss = matrix(NA, nrow=1, ncol=13)

## STEP 2 - We randomly choose a percentage of the response values to set equal to NA
## We then fit a Linear Mixed Effects Random Slope model to the data using lmer.
## We then obtained the RSS of each model and input that value into our empty 'rss' matrix
for (i in 1:13){
  mis = round(runif((578*x[i]), min=0, max=578))
  ChickWeightNA = ChickWeight
  ChickWeightNA[mis,1] = NA
  chick.lmm = lmer(weight ~ Time +(1|Chick), data=ChickWeightNA)
  chick.lmm.pred = predict(chick.lmm, ChickWeight)
  rss[i] = sum((chick.lmm.pred - ChickWeight[,1])^2)
}

##Then checked our RSS
rss
# Call this command to check how your RSS changes with different amounts
# of missing data
# plot(x[1:13], rss)
# Note - Step 1 and Step 2 may need to be repeated multiple times before a full set
# of RSS values is obtained. This is due to some chicks having very
# few response values before dropping out. As we take more data, the chance of all response
# values of a chick being set to NA increases. R then cannot
# fit a model to the dataset.

## Once a full set of RSS values is obtained, save this as rssperf
rssperf = rss

## Multiplying x by 100 to give us percentage values
xperc = 100*x

## Only taking up to 60% of data so only need x values 1:13
xperc = xperc[1:13]

## Making xperc a matrix so we can created a combined dataframe which ggplot can understand
xperc = as.matrix(xperc, nrow=1)
```

```
## Creating a combined dataframe of percentages and rss values
plotdf = as.data.frame(cbind(xperc, t(rssperf)))

## Changing the names of the columns so we can refer to them in ggplot
names(plotdf) = c('Percentage', 'RSS')

## Plotting Percentage vs RSS value. See an overall positive trend.
ggplot(plotdf, aes(x=Percentage, y=RSS, xlab='Percentage of Missing Data'),
ylab='Residual Sum of Squares')+
  geom_point(col='red',size=2.5)+
  geom_smooth(method='lm', se=FALSE)+
  labs(x='Percentage of Missing Data', y='Residual Sum of Squares')+
  theme(axis.title = element_text(size=25), axis.text = element_text(size=20))
```

Table 6.2 Code

```
DietA = ChickWeight[1:220,]
DietB = ChickWeight[221:340,]
DietC = ChickWeight[341:460,]
DietD = ChickWeight[461:578,]

#Here we do mean imputation

mean(DietA[as.vector(which(DietA$Time==4)),1])  # 56.47368
mean(DietA[as.vector(which(DietA$Time==6)),1])  # 66.78947
mean(DietA[as.vector(which(DietA$Time==8)),1]) # 79.68421
mean(DietA[as.vector(which(DietA$Time==10)),1]) # 93.05263
mean(DietA[as.vector(which(DietA$Time==12)),1]) # 108.5263
mean(DietA[as.vector(which(DietA$Time==14)),1]) # 123.3889
mean(DietA[as.vector(which(DietA$Time==16)),1]) # 144.6471
mean(DietA[as.vector(which(DietA$Time==18)),1]) # 158.9412
mean(DietA[as.vector(which(DietA$Time==20)),1]) # 170.4118
mean(DietA[as.vector(which(DietA$Time==21)),1]) # 177.75

mean(DietD[as.vector(which(DietD$Time==20)),1]) # 233.8889
mean(DietD[as.vector(which(DietD$Time==21)),1]) # 238.5556
```

Figure 6.2 Code

```
# Making a dataframe containing only the chicks which dropped out
drop = ChickWeight[c(as.vector(which(ChickWeight$Chick==8)),
as.vector(which(ChickWeight$Chick==15)),
as.vector(which(ChickWeight$Chick==16)),
```

```
as.vector(which(ChickWeight$Chick==18)),
as.vector(which(ChickWeight$Chick==44))),]

# Call plot to plot the incomplete data (Figure 6.2a)

plot = ggplot(data=drop, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col='Chick')+
  theme(axis.text = element_text(size=20), axis.title = element_text(size=25),
  legend.key.height = unit(3, 'cm'), legend.title=element_text(size=15))

# Making a dataframe containing all imputed values

t = data.frame(weight=c(178, 159, 170, 178, 123, 145, 159, 170, 178, 56, 67, 80, 93
, 109, 123, 145, 158, 170, 178, 234, 239),
Time=c(21, 18, 20, 21, 14, 16, 18, 20, 21, 4, 6, 8, 10,12, 14, 16, 18, 20,21, 20, 21),
Chick=c(8, 15, 15, 15, 16, 16, 16, 16, 16, 18, 18, 18, 18, 18, 18, 18,18, 18, 18, 44, 44),
Diet=c(1, 1, 1, 1, 1 ,1, 1,1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4))

Chickweightt = ChickWeight
complete = rbind(Chickweightt, t)

drop = complete[c(as.vector(which(complete$Chick==8)),
as.vector(which(complete$Chick==15)),
as.vector(which(complete$Chick==16)),as.vector(which(complete$Chick==18)),
as.vector(which(complete$Chick==44))),]

#Call plot to plot the imputed data (Figure 6.2b)

plot = ggplot(data=drop, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col='Chick')+
  theme(axis.text = element_text(size=20), axis.title = element_text(size=25),
  legend.key.height = unit(3, 'cm'), legend.title=element_text(size=15))
```

Figure 6.3 Code

```
library(mice)
library(ggplot2)
Chickfull = read.csv('Chickfull.csv')
Chickfull = Chickfull[,-1]
names(Chickfull) = c('weight', 'Time', 'Chick', 'Diet')
Chickfull$Chick = as.factor(Chickfull$Chick)
Chickfull$Diet = as.factor(Chickfull$Diet)
```

```
Chickdrop = Chickfull[c(which(Chickfull$Chick==8),which(Chickfull$Chick==15),
which(Chickfull$Chick==16),which(Chickfull$Chick==18),which(Chickfull$Chick==44)),]

compchick  =complete(mice(Chickfull, m=20))
# we tried specifying method='2l.lmer' however this did not give good imputed values

comppp = compchick[c(which(compchick$Chick==8),which(compchick$Chick==15),
which(compchick$Chick==16),which(compchick$Chick==18),which(compchick$Chick==44)),]

# Call plot1 and plot2 to plot the data

plot1 = ggplot(data=comppp, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col='Chick')+
  theme(axis.text = element_text(size=20), axis.title = element_text(size=25),
  legend.key.height = unit(3, 'cm'), legend.title=element_text(size=15))
plot2 = ggplot(data=Chickdrop, aes(x=Time, y=weight, col=Chick))+
  geom_line(size=1.2, alpha=0.8)+
  labs(x = 'Time (Days)', y='Weight (g)', col='Chick')+
  theme(axis.text = element_text(size=20), axis.title = element_text(size=25),
  legend.key.height = unit(3, 'cm'), legend.title=element_text(size=15))
```

Table 6.3 and 6.4 Code - MICE Algorithm Example

```
## going to make a step by step guide for the MICE algorithm using dummy data.


dummy = matrix(c(12,16,20,30,45,60,80,75,65,70,60,40,85,65,70,65,50,35), ncol=3)
dummy = as.data.frame(dummy)
names(dummy) = c('Age', 'Hearing', 'Sight')
dummy[4,1] = NA
dummy[3,2] = NA
dummy[1,3] = NA
dummy


round(runif(1, min=1, max=5))

#iteration 1

dummy1 = dummy
dummy1[4,1] = 16
dummy1[3,2] = 60
dummy1[1,3] = 50
```

```
dummy1[4,1]=NA
dummy1


lm(Age ~ Hearing+Sight, data=dummy1[-4,])
##120.1094 - 0.9331H - 0.569S
dummy1[4,1] = 17.8074
lm(Hearing~Age+Sight, data=dummy1[-3,])
## 89.9292 - 0.7446A -0.0467S
dummy1[3,2] = 71.7682
lm(Sight ~ Age+ Hearing, data=dummy1[-1,])
## 41.1973 - 0.3987A + 0.4493H

dummy1[1,3] = 72.3569
dummy1
#iteration 2

lm(Age~ Hearing + Sight, data=dummy2[-4,])
## Age = 112.9239 - 0.673H - 0.6644S
lm(Hearing ~ Age + Sight, data=dummy2[-3,])
## Hearing = -44.1922 + 0.4111A + 1.7029S
lm(Sight ~ Age + Hearing, data=dummy2[-1,])
## Sight = 45.6905 - 0.4363A + 0.3980H

dummy3 = dummy2

#iteration 3
dummy3[4,1] = 22.6276
dummy3[3,2] = 83.6738
dummy3[1,3] = 72.2949


lm(Age~ Hearing + Sight, data=dummy3[-4,])
## 105.0181 + 0.9411H -2.3634S
lm(Hearing ~ Age + Sight, data=dummy3[-3,])
## 12.1174 - 0.0658A + 0.9560S
lm(Sight ~ Age + Hearing, data=dummy3[-1,])
## 36.0270 -0.3359A + 0.4863H
```

Code 6.1 - `selectmultMNAR`

```
selectmultMNAR = function(data){
  # Calculating the number of observations per subject
  n_obs = as.numeric(dim(data)[2])
```

```
# Calculating the number of individuals in the dataset
nsub = as.numeric(dim(data)[1])
# Calculating the total number of observations
N = n_obs*nsub
# The function does not assume that the inputted data is ordered in terms of
# missingness (ie all individuals with no missing observations above the
# individuals who have the last observation missing). This part of the code
# rearranges the dataset to give a new dataset which has the individuals
# with fully observed measurements at the top of the dataset, then the
# individuals only missing the last observation, then the individuals missing
# the last two observations etc...

# This also allows to generalise when we have more than the final observation
# missing.
data1=data
empty = matrix(NA, nrow=0, ncol=n_obs)
empty = as.data.frame(empty)
for (i in 2:n_obs){
  if (identical(which(is.na(data1[,i])), integer(0))){
    empty = empty
  }else{
  mis = data1[which(is.na(data1[,i])),]
  data1 = data1[-which(is.na(data1[,i])),]
  empty = rbind(mis,empty)
  }
}
empty = rbind(data1, empty)
data = empty

m = as.numeric(min(which(is.na(data[,n_obs])==TRUE)))

data = as.matrix(data)


# Now creating a function representing the log likelihood as in Equation 6.1
loglikkk = function(inits){
  mu = matrix(c(inits[1:n_obs]),nrow=1, ncol=n_obs)
  Sigma = matrix(c(inits[(n_obs+1):(n_obs+(n_obs*n_obs))]), nrow=n_obs, ncol=n_obs, byrow=TRUE)
  phiv = matrix(c(inits[(n_obs+(n_obs*n_obs)+1):length(inits)]), nrow=1, ncol=(n_obs+1))
  detsig = det(Sigma)
  invsig = as.matrix(solve(Sigma))
  ll = 0
  for (i in 1:(m-1)){
    yone = as.matrix(cbind(1, t(data[i,])))
    logis = phiv%*%t(yone)
    P = exp(logis)/(1+exp(logis))
```

```
    difff = data[i,] - mu
    mat1 = difff%*%invsig
    mat1 = mat1%*%t(difff)
    mat1 = as.numeric(mat1)
    ll = ll + (-1/2)*log(detsig) - (1/2)*mat1 + log(P)
  }
  ll2=0
  for (i in m:nsub){
  # We perform importance sampling to approximate the integral
  # Numerator function
    gfunc = function(ql){
      yone = as.matrix(cbind(1,t(data[i,1:(n_obs-1)]),ql))
      logis = phiv%*%t(yone)
      P = exp(logis)/(1+exp(logis))
      difff = cbind(t(data[i,1:(n_obs-1)]), ql) - mu
      mat1 = difff%*%invsig
      mat1 = mat1%*%t(difff)
      mat1 = as.numeric(mat1)
      return(exp(-(1/2)*mat1)*(1-P))
    }
    # Denominator function
    hfunc = function(x){
      # The distribution we sample from is a shifted standard normal, with the
      # mean equal to the sample mean of the last column
      return((1/sqrt(2*pi))*exp(((x-mu[n_obs])^2)*(-1/2)))
    }
    # Performing importance sampling
    impsample = matrix(NA, nrow=1, ncol=100)
    for (q in 1:100){
      r = rnorm(1, mean=mu[n_obs], sd=1)
      impsample[q] = (gfunc(r)/hfunc(r))
    }
    impsample = as.vector(impsample)
    # Finding the average of the importance sample values to give an estimate
    # for the integral
    intapprox = mean(impsample)
    covminn = as.matrix(Sigma[-n_obs,-n_obs])
    detcovminn = det(covminn)
    d = (1/2)*log(detcovminn)
    p = log(intapprox)
    ll2 = ll2 - d +p
  }
  return(ll+ll2)
}
```

```
  # Here we grab some initial values
  # Obtain the inital mean values by performing a complete case analysis and
  # calculating the mean from that.
  Ycomp = data[-(m:nsub),]
  meanvec = matrix(NA, nrow=1, ncol=n_obs)
  for (l in 1:n_obs){
    meanvec[l] = mean(Ycomp[,l])
  }
  meanvec = as.vector(meanvec)

  # Estimating a covariance matrix
  mumat = matrix(meanvec, nrow=(m-1), ncol=n_obs, byrow=TRUE)
  difmu = as.matrix(Ycomp - mumat)
  sampcov = as.matrix((1/(m-1))*t(difmu)%*%difmu)

  # Initialise the phi vector with (0.01, ..., 0.01, 0.05)
  # This allows for convergence
  phiv = cbind(matrix(0.01, nrow=1, ncol=n_obs),0.05)
  phiv = as.vector(phiv)
  initial_vals = list(meanvec, sampcov, phiv)
  print(initial_vals)
  opvals = optim(unlist(initial_vals), loglikkk, control=list(maxit=1000), method='SANN')
  # We use the SANN method as it seems to perform best in optimisation
  return(opvals)
}
```

Table 6.5 Code

```
library(mvtnorm)
library(nlme)
library(lme)
library(mice)
library(ggplot2)


meanv4 = c(100, 200, 300, 400)
cov4 = matrix(c(20,15,10,5,15,20,15,10,10,15,20,15,5,10,15,20), nrow=4, byrow=TRUE)
##call this to get exactly the same dataset
set.seed(220202)
Y4 = rmvnorm(n=100, mean=meanv4, sigma=cov4)

# getting the design matrix to help us calculate the covariance matrix
mati = model.matrix(lmer(Y ~ Time + (1|ID), data=longdata(Y4)))[1:4,]
```

```
matt = matrix(NA, nrow=20, ncol=0)
matt1 = matrix(NA, nrow=20, ncol=0)
matt2 = matrix(NA, nrow=20, ncol=0)
matt3 = matrix(NA, nrow=20, ncol=0)
simulation = matrix(NA, nrow=25, ncol=0)
library(mice)
library(nlme)
library(lme4)
for (i in 1:85){
  misi = round(runif(30, min=0, max=100))
  Y4misi = Y4
  Y4misi[misi,4] = NA

  ## our selection model

  simulation = rbind(simulation, selectmultMNAR(Y4new)$par)

  ## no imputation performed - complete case analysis

  Y4misimod = lmer(Y ~ Time + (Time|ID), data=longdata(Y4misi))
  result = mati%*%summary(Y4misimod)$coef[,1]
  sigi = (summary(Y4misimod)$sigma)^2
  zqzsig = mati%*%matrix(VarCorr(Y4misimod)[[1]][1:4], nrow=2, ncol=2)%*%t(mati)+sigi*diag(4)
  covsi = rbind(t(t(zqzsig[1,])), t(t(zqzsig[2,])), t(t(zqzsig[3,])), t(t(zqzsig[4,])))
  print(covsi)
  result = rbind(result, covsi)
  matt = cbind(matt, result)


  ## last observation carried forward


  Y4locf = Y4misi
  for (j in misi){
    Y4locf[j,4] = Y4locf[j,3]
  }
  Y4locfmod = lmer(Y ~ Time + (Time|ID), data=longdata(Y4locf))
  resultlocf = mati%*%summary(Y4locfmod)$coef[,1]
  sigilocf = (summary(Y4locfmod)$sigma)^2
  zqzsiglocf = mati%*%matrix(VarCorr(Y4locfmod)[[1]][1:4], nrow=2,
  ncol=2)%*%t(mati)+sigilocf*diag(4)
  covsilocf = rbind(t(t(zqzsiglocf[1,])), t(t(zqzsiglocf[2,])), t(t(zqzsiglocf[3,])),
  t(t(zqzsiglocf[4,])))
  resultlocf = rbind(resultlocf, covsilocf)
  matt1 = cbind(matt1, resultlocf)
```

```
  # single mean imputation

  Y4meanimp = Y4misi
  Y4meanimp[misi,4] = mean((Y4misi[-misi,])[,4])
  Y4mimod = lmer(Y ~ Time + (Time|ID), data=longdata(Y4meanimp))
  resultmi = mati%*%summary(Y4mimod)$coef[,1]
  sigimi = (summary(Y4mimod)$sigma)^2
  zqzsigmi = mati%*%matrix(VarCorr(Y4mimod)[[1]][1:4], nrow=2, ncol=2)%*%t(mati)+sigimi*diag(4)
  covsimi = rbind(t(t(zqzsigmi[1,])), t(t(zqzsigmi[2,])), t(t(zqzsigmi[3,])), t(t(zqzsigmi[4,])))
  resultmi = rbind(resultmi, covsimi)
  matt2 = cbind(matt2, resultmi)

  # multiple imputation

  Y4mice = complete(mice(Y4misi, m=8))
  Y4micemod = lmer(Y ~ Time + (Time|ID), data=longdata(Y4mice))
  resultmice = mati%*%summary(Y4micemod)$coef[,1]
  sigimice = (summary(Y4micemod)$sigma)^2
  zqzsigmice = mati%*%matrix(VarCorr(Y4micemod)[[1]][1:4], nrow=2,
  ncol=2)%*%t(mati)+sigimice*diag(4)
  covsimice = rbind(t(t(zqzsigmice[1,])), t(t(zqzsigmice[2,])), t(t(zqzsigmice[3,])),
  t(t(zqzsigmice[4,])))
  resultmice = rbind(resultmice, covsimice)
  matt3 = cbind(matt3, resultmice)
}

#splitting up the data in our parameter estimate matrix

simulation = simulation[,1:20]
simulationmean = simulation[,1:4]
simulationcov = simulation[,5:20]

# splitting up the data for each of the methods. Need to take transpose

mattmean = matt[1:4,]
mattmean = t(mattmean)
matcov = matt[5:20,]
matcov = t(matcov)

matt1mean = matt1[1:4,]
matt1mean = t(matt1mean)
matt1cov = matt1[5:20,]
matt1cov = t(matt1cov)

matt2mean = matt2[1:4,]
```

```
matt2mean = t(matt2mean)
matt2cov = matt2[5:20,]
matt2cov = t(matt2cov)

matt3mean = matt3[1:4,]
matt3mean = t(matt3mean)
matt3cov = matt3[5:20,]
matt3cov = t(matt3cov)

# calculating average parameter estimates for each method

totalmatt = colMeans(t(matt))
totalmatt1 = colMeans(t(matt1))
totalmatt2 = colMeans(t(matt2))
totalmatt3 = colMeans(t(matt3))
totalsim = colMeans(simulation)

true = c((meanv4), as.vector(cov4)) # vector containing true values

# Calculating average absolute estimation error over the 4 mean parameters
# Corresponds to the 'Mean' column in Table 6.5
# Can also calculate individual parameters using this code.

mean(abs(totalmatt[1:4] - true[1:4]))
mean(abs(totalmatt1[1:4] - true[1:4]))
mean(abs(totalmatt2[1:4] - true[1:4]))
mean(abs(totalmatt3[1:4] - true[1:4]))
mean(abs(totalsim[1:4] - true[1:4]))
```

Figure 6.5 Code

```
# Calculating matrices of parameter estimation errors

# Complete Case Analysis
errormattmeanna = matrix(NA, nrow=85, ncol=4)
errormattcovna = matrix(NA, nrow=85, ncol=16)

for (i in 1:85){
  errormattmeanna[i,] = (mattmean[i,] - meanv4)
  errormattcovna[i,] = (matcov[i,] - as.vector(cov4))
}

# Last Observation Carried Forward
errormatt1meanna = matrix(NA, nrow=85, ncol=4)
```

```
errormatt1covna = matrix(NA, nrow=85, ncol=16)

for (i in 1:85){
  errormatt1meanna[i,] = (matt1mean[i,] - meanv4)
  errormatt1covna[i,] = (matt1cov[i,] - as.vector(cov4))
}

# Single Mean Imputation
errormatt2meanna = matrix(NA, nrow=85, ncol=4)
errormatt2covna = matrix(NA, nrow=85, ncol=16)

for (i in 1:85){
  errormatt2meanna[i,] = (matt2mean[i,] - meanv4)
  errormatt2covna[i,] = (matt2cov[i,] - as.vector(cov4))
}

# Multiple Imputation
errormatt3meanna = matrix(NA, nrow=85, ncol=4)
errormatt3covna = matrix(NA, nrow=85, ncol=16)

for (i in 1:85){
  errormatt3meanna[i,] = (matt3mean[i,] - meanv4)
  errormatt3covna[i,] = (matt3cov[i,] - as.vector(cov4))
}

# Using selectmultMNAR
errorsimmeanna = matrix(NA, nrow=85, ncol=4)
errorsimcovna = matrix(NA, nrow=85, ncol=16)

for (i in 1:85){
  errorsimmeanna[i,] = (simulationmean[i,] - meanv4)
  errorsimcovna[i,] = (simulationcov[i,] - as.vector(cov4))
}

# Putting each matrix into a dataframe format for the purpose of using ggplot
# Creating a list of the standard errors. We omit LOCF since it will skew the plot too much.

datalmercovna = data.frame(error = as.vector(errormattcovna))
datameanimpcovna = data.frame(error = as.vector(errormatt2covna))
datamultimpcovna = data.frame(error = as.vector(errormatt3covna))
datameansimcovna = data.frame(error = as.vector(errorsimcovna))

dataviocovna = data.frame(Method = rep(c('LMEM - Complete Case Analysis',
'LMEM - Single Mean Imputation', 'LMEM - Multiple Imputation',
'Selection Model Function'), each=1360),
```

```
error = c(as.matrix(datalmercovna),as.matrix(datameanimpcovna),
as.matrix(datamultimpcovna), as.matrix(datameansimcovna)))

ggplot(data=dataviocovna, aes(x=Method, y=error, fill=Method))+
  geom_violin()+
  geom_boxplot(width=0.2, color='black', outlier.shape = NA)+
  theme_minimal()+
  labs(y = 'Estimation Error',
  title = 'Violin Plot of Estimation Errors for Covariance Parameters')+
  theme(axis.text.y = element_text(size=15), axis.text.x = element_text(size = 15),
  legend.position = 'none', axis.title.x = element_text(size = 15),
  axis.title.y = element_text(size=20), title = element_text(size=15),
  plot.title = element_text(hjust=0.5))
```

Code 6.2 - `selectmultMNARevery`

```
selectmultMNARevery = function(data){
  # Calculating number of measurements per subject
  n_obs = as.numeric(dim(data)[2])
  # Calculating number of subjects
  nsub = as.numeric(dim(data)[1])
  # Calculating total number of observations in dataset
  N = n_obs*nsub
  # The function does not assume that the inputted data is ordered in terms of
  # missingness (ie all individuals with no missing observations above the
  # individuals who have the last observation missing). This part of the code
  # rearranges the dataset to give a new dataset which has the individuals
  # with fully observed measurements at the top of the dataset, then the
  # individuals only missing the last observation, then the individuals missing
  # the last two observations etc...
  data1=data
  empty = matrix(NA, nrow=0, ncol=n_obs)
  empty = as.data.frame(empty)
  for (i in 2:n_obs){
    if (identical(which(is.na(data1[,i])), integer(0))){
      empty = empty
    }else{
    mis = data1[which(is.na(data1[,i])),]
    data1 = data1[-which(is.na(data1[,i])),]
    empty = rbind(mis,empty)
    }
  }
  empty = rbind(data1, empty)
  data = empty
```

```
# This section calculates at which individual we start to observe missingness
# in a measurement. We know the individuals are now ordered in terms of
# missingness from the previous section

m = as.numeric(min(which(is.na(data[,n_obs])==TRUE)))
missingindicator = function(data){
  m = matrix(NA, nrow=1, ncol=(dim(data)[2]-1))
  for (i in 1:(dim(data)[2]-1)){
    m[i] = as.numeric(min(which(is.na(data[,(dim(data)[2]+1-i)])==TRUE)))
  }
  return(m)
}

ms = missingindicator(data)
ms = cbind(ms, (nsub+1))

data = as.matrix(data)

#This function now calculates the log-likelihood given a set of values
loglikkk = function(inits){
  mu = matrix(c(inits[1:n_obs]),nrow=1, ncol=n_obs)
  Sigma = matrix(c(inits[(n_obs+1):(n_obs+(n_obs*n_obs))]), nrow=n_obs, ncol=n_obs, byrow=TRUE)
  phiv = matrix(c(inits[(n_obs+(n_obs*n_obs)+1):length(inits)]), nrow=1, ncol=(n_obs+1))
  #print(phiv)
  detsig = det(Sigma)
  invsig = as.matrix(solve(Sigma))
  #print(invsig)
  ll = 0


# Calculating the log likelihood when we have missingness anywhere is much
# more complicated. We do it by using a for loop going over each set of
# individuals who have a certain amount of missingness.
for (i in 1:(ms[1]-1)){
  yone = as.matrix(cbind(1, t(data[i,])))
  logis = phiv%*%t(yone)
  P = exp(logis)/(1+exp(logis))
  difff = data[i,] - mu
  mat1 = difff%*%invsig
  mat1 = mat1%*%t(difff)
  mat1 = as.numeric(mat1)
  ll = ll + (-1/2)*log(detsig) - (1/2)*mat1 + log(P)
}
for(i in 1:(dim(ms)[2]-1)){
  for (j in (ms[i]:(ms[i+1]-1))){
```

```
    gfunc = function(ql){
      ql = t(as.matrix(ql))
      yone = cbind(as.matrix(1),t(as.matrix(data[j,1:(n_obs-i)])),ql)
      logis = phiv%*%t(yone)
      P = exp(logis)/(1+exp(logis))
      difff = cbind(t(as.matrix(data[j,1:(n_obs-i)])), ql) - mu
      mat1 = difff%*%invsig
      mat1 = mat1%*%t(difff)
      mat1 = as.numeric(mat1)
      return(exp(-(1/2)*mat1)*(1-P))
    }
    hfunc = function(x){
      # Performing multivariate sampling
      return(dmvnorm(x=x, as.vector(mu[,((n_obs+1-i):n_obs)]), diag(i)))
    }
    impsample = matrix(NA, nrow=1, ncol=100)
    for (q in 1:100){
      r = as.vector(mvrnorm(1, mu[,((n_obs-i+1):n_obs)], as.matrix(diag(i))))
      impsample[q] = (gfunc(r)/hfunc(r))
    }
    impsample = as.vector(impsample)
    intapprox = mean(impsample)
    covminn = as.matrix(Sigma[-(n_obs+1-i),-(n_obs+1-i)])
    detcovminn = det(covminn)
    d = (1/2)*log(detcovminn)
    p = log(intapprox)
    ll = ll - d +p
  }
  #print(ll)
}
return(ll)
}
#  getting some initial values (same as in Code 6.1)
Ycomp = data[-(ms[1]:nsub),]
meanvec = matrix(NA, nrow=1, ncol=n_obs)
for (l in 1:n_obs){
  meanvec[l] = mean(Ycomp[,l])
}
meanvec = as.vector(meanvec)

##now need to estimate a covariance matrix
mumat = matrix(meanvec, nrow=(m-1), ncol=n_obs, byrow=TRUE)
difmu = as.matrix(Ycomp - mumat)
sampcov = as.matrix((1/(m-1))*t(difmu)%*%difmu)
```

```
  phiv = cbind(matrix(0.01, nrow=1, ncol=n_obs),0.05)
  phiv = as.vector(phiv)
  initial_vals = list(meanvec, sampcov, phiv)
  opvals = optim(unlist(initial_vals), loglikkk, control=list(maxit=1000), method='SANN')
  return(opvals)
}
```

Code 6.3 - `selectmultMNARany`

```
install.packages('mvtnorm')
library(MASS)
library(mvtnorm)

selectmultMNARany = function(data){
  n_obs = as.numeric(dim(data)[2])
  nsub = as.numeric(dim(data)[1])
  N = n_obs*nsub



  data = as.matrix(data)


  loglikkk = function(inits){
  mu = matrix(c(inits[1:n_obs]),nrow=1, ncol=n_obs)
  Sigma = matrix(c(inits[(n_obs+1):(n_obs+(n_obs*n_obs))]), nrow=n_obs, ncol=n_obs, byrow=TRUE)
  phiv = matrix(c(inits[(n_obs+(n_obs*n_obs)+1):length(inits)]), nrow=1, ncol=(n_obs+1))
  #print(phiv)
  detsig = det(Sigma)
  invsig = as.matrix(solve(Sigma))
  #print(invsig)
  ll = 0
  for (i in 1:nsub){
    if (length(which(is.na(data[i,])))==0){
      yone = as.matrix(cbind(1, t(data[i,])))
      logis = phiv%*%t(yone)
      P = exp(logis)/(1+exp(logis))
      difff = data[i,] - mu
      mat1 = difff%*%invsig
      mat1 = mat1%*%t(difff)
      mat1 = as.numeric(mat1)
      ll = ll + (-1/2)*log(detsig) - (1/2)*mat1 + log(P)
    }else{
    miss = which(is.na(data[i,]))
```

```
  miss = as.vector(miss)
  signew = Sigma[-miss, -miss]
  gfunc = function(ql){
  yone = as.matrix(cbind(1, t(data[i,])))
  yone[miss] = ql
  logis = phiv%*%t(yone)
  P = exp(logis)/(1+exp(logis))
  difff = t(as.matrix(yone[-1])) - mu
  mat1 = difff%*%invsig
  mat1 = mat1%*%t(difff)
  mat1 = as.numeric(mat1)
  return(exp(-(1/2)*mat1)*(1-P))
  }
  hfunc = function(x){
    return(dmvnorm(x=x, as.vector(mu[,miss]), diag(length(miss))))
  }
  impsample = matrix(NA, nrow=1, ncol=100)
    for (q in 1:100){
      r = as.vector(mvrnorm(1, mu[,miss], as.matrix(diag(length(miss)))))
      impsample[q] = (gfunc(r)/hfunc(r))
    }
  impsample = as.vector(impsample)
  intapprox = mean(impsample)
  covminn = as.matrix(Sigma[-miss,-miss])
  detcovminn = det(covminn)
  d = (1/2)*log(detcovminn)
  p = log(intapprox)
  ll = ll - d +p
  }
}
return(ll)
}
###  grabbing some initial values for the y's
### fist grabbing a mu hat, just by doing a complete case analysis and taking the
### mean of each column

# can make a slightly naive assumption that there will be some individuals
# who do not miss measurements

m = which(is.na(data))%%nsub
for (i in m){
  if(i==0){
    i = nsub
  }else{i=i}
}
```

```
  Ycomp = data[-m,]
  meanvec = matrix(NA, nrow=1, ncol=n_obs)
  for (l in 1:n_obs){
    meanvec[l] = mean(Ycomp[,l])
  }
  meanvec = as.vector(meanvec)
  ##now need to estimate a covariance matrix
  mumat = matrix(meanvec, nrow=dim(Ycomp)[1], ncol=n_obs, byrow=TRUE)
  difmu = as.matrix(Ycomp - mumat)
  sampcov = as.matrix((1/(dim(Ycomp)[1]))*t(difmu)%*%difmu)
  phiv = cbind(matrix(0.01, nrow=1, ncol=n_obs),0.05)
  phiv = as.vector(phiv)
  initial_vals = list(meanvec, sampcov, phiv)
  print(initial_vals)
  opvals = optim(unlist(initial_vals), loglikkk, control=list(maxit=1000), method='SANN')
  return(opvals)
}
```