Mean-variance dependency in ANOVA models definition and estimation

Boom, E.F.J.

Award date:
2019

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science

Mean-variance dependency in ANOVA models: definition and estimation

Master Thesis

Boom, E.F.J

Supervisors:
prof. dr. E.R. van den Heuvel
M. Regis MSc

Eindhoven, April 2019

Abstract

Hierarchical models constitute a popular approach for the analysis of clustered data (e.g. repeated measurements data). Standard working assumptions include variance components being constant. However, this hypothesis is often violated in practical examples, and there is an increasing interest in modeling variances and understanding how they affect the response variable. Examples in the literature include time-, covariates- and factor- dependent variance components.

In this thesis we focus on hierarchical models in which the standard deviation at each level is proportional to the average realization of the hierarchical levels below. We tackle estimation by proposing the method of moments, maximum likelihood and an adaptation of the two-step approaches proposed in the literature by Yeap, Davidian (2001) and Davidian, Giltinan (1993). We run two intensive simulation studies and report the estimation biases of different methods. We show the results on two- and three-level hierarchical models, but the formulation and methods are general and can be applied to heteroskedastic hierarchical models with different number of levels.

Contents

Chapter 1

Introduction

Study designs often involve the collection of measurements from multiple subjects repeatedly over time, for example for clinical trials and the assessment of product quality. In these studies, the interest lies on estimating the effect of certain factors or covariates, for example the effect of a treatment in a clinical trial, while correcting for other confounding factors (e.g. age and gender). Mixed models are characterized by the inclusion of fixed (traditional coefficients linked to some covariates) and random effects. Fixed effects are deterministic coefficients associated with factors that one can control during the measurements (for instance the treatment in a clinical trial), while random effects are coefficients associated with factors from a random sample from some distribution (for instance the subjects in a similar trial). Fixed effects model the average trend, while random effects are used to capture subject-specific features under a common modeling structure.
The measurements are modeled through the following equation:

$$y_{ij} = f(\beta, u_i) + e_{ij},$$

where y_{ij} is the jth measurement taken on the ith subject, β represents the fixed effects, u_i the random effects, and e_{ij} the intra-subject residual. When the subject-specific trend is assumed to be constant and a random intercept only, more specifically $f(\beta, u_i) = \beta + u_i$, the model reduces to a simple ANOVA model with β the overall mean and u_i the subject-specific term for subject i. In this setting, combined with the residual having mean 0, the possibly non-linear function $f(\beta, u_i)$ represents the subject-specific mean.
Traditional assumptions for mixed effect models include independence across subjects i. Furthermore, for the ANOVA case, the random terms are assumed to be both normally distributed, with constant variances, and mutually independent:

$$u_i \sim \mathcal{N}(0, \sigma_G^2), e_{ij} \sim \mathcal{N}(0, \sigma_E^2).$$

To estimate the parameters in this model (β and the two variance components), general methods include (restricted) maximum likelihood and the Method of Moments.

This model is very popular and has been applied to a broad spectrum of applications. However, in practical scenarios the hypothesis of constant variance is often violated. Residual variance and variance of the random effects are often varying across units, with some level of covariates or with the average value of the measurement being taken. This thus raises the need for more flexible modeling frameworks, allowing modeling of the heterogeneities encountered in practical examples. In fact, a broad literature on heteroskedastic mixed models exist. These approaches can be generalized into two directions: modeling extra randomness or extra dependencies. The extra randomness is modeled, by taking the variance of the random model terms as a random variable itself.

A common choice is to introduce a latent random variable following a Gamma distribution in the definition of the variance components:

$$u_i|\tau_i \sim \mathcal{N}(0, \frac{1}{\tau_i}\sigma_G^2), e_{ij}|\tau_i \sim \mathcal{N}(0, \frac{1}{\tau_i}\sigma_E^2),$$
$$\tau_i \sim \Gamma(\nu_i/2, \nu_i/2),$$

where ν_i is an additional parameter that needs to be estimated. Consequently, it holds that the response variable, alongside u_i and e_{ij}, are marginally t distributed, with ν_i degrees of freedom:

$$u_i \sim t(0, \sigma_G^2, \nu_i), e_{ij} \sim t(0, \sigma_E^2, \nu_i).$$

Because of this property and the assumption of linear mean structure, this model is known as t linear mixed model. Pinheiro, Liu, Wu (2001) and Lin, Lee (2006) apply this model with some different characteristics. Pinheiro, Liu, Wu (2001) assumes that the degrees of freedom ν_i is allowed to be different per subject, but warns about identifiability in that case. Lin, Lee (2006) assumes that ν_i is independent and constant for all subjects: $\nu_i = \nu$. However, they added an AR(1) structure to the residual variance term and propose a likelihood-based estimation method.

The second direction is to model additional dependencies in the variance structure, using either some level of covariates or the entire mean structure. This approach has been used in both the non-linear and the linear setting. In the linear setting, Kizilkaya, Templeman (2005) applied a generalized version of the mixed model, where the covariate levels are used as scaling factors of the residual variance. Li, Bruyneel, Lessafre (2014) opted for a similar strategy, where they used the following:

$$e_{ij} = F_{ij}\Lambda_i + \epsilon_{ij},$$
$$F_{ij} \sim \mathcal{N}(0,1), \Lambda_i = p(\beta + u_i), \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

where p is an unknown function, and σ is to be determined. They applied Gibbs sampling to estimate the model parameters.

Davidian, Giltinan (2003) provide a complete overview of the possible structures in the non-linear setting. Both Davidian, Giltinan (1993) and Yeap, Davidian (2001) apply the model where the residual is dependent on the mean structure:

$$e_{ij} = \epsilon_{ij}\sigma g(f(\beta, u_i), \theta),$$

where ϵ_{ij} is standard normally distributed. They both provide two-step estimation methods that first estimate the subject-specific means and the population coefficient of variation σ, and then in a second step estimate the parameters of the distribution of the random effects. These methods are all based on M-estimators.

Regis, Postma, van den Heuvel (2017) opted to include dependencies on the mean in the subject-specific term as well, which we can see it as an extension of the linearized version of the model of Yeap Davidian (2001) and Davidian, Giltinan (1993), and applied this model when there are exactly 2 consecutive measurements. In this thesis we extend the model with mean-variance dependencies of Regis, Postma, van den Heuvel (2017), to allow for multiple measurements and investigate its identifiability. In fact, our hierarchical model with two levels is a special case of the model of Davidian, Giltinan (1993) and Yeap, Davidian (2001) in which the average trend is linear.

The outline of the thesis is as follows: in Section 2, we will describe the models we are going to use. Section 3 consists of the different estimation methods we will apply. We will end with a simulation study and a discussion.

Chapter 2

Model description

In this section, we define the hierarchical mixed model with mean-variance dependencies, and introduce the notation that will be used in the following. Furthermore, we make some qualitative considerations on the behavior of the marginal distribution.

2.1 Two-level hierarchical model

Let y_{ij} be a measurement for subject $i \in 1, ..., I$, at time point $j \in 1, ..., J$, where we work under the assumption of balanced design. Furthermore, we assume that all subjects are independent of each other, and that the measurements can be modeled by the following additive structure:

$$y_{ij} = \mu_i + \delta_{ij}, \tag{2.1}$$

where μ_i is the true mean of subject i and δ_{ij} represents the residual term, resulting from the combination of the random intra-subject variability and the random analytical variability. The individual mean μ_i is a random sample from a normal distribution:

$$\mu_i \sim \mathcal{N}(\mu, (c_0\mu)^2), \tag{2.2}$$

with μ the population mean, and c_0 the between-subject coefficient of variation.

Next we assume that δ_{ij} is normally distributed, with mean 0 and variance proportional to μ_i:

$$\delta_{ij} \sim \mathcal{N}(0, (c_S\mu_i)^2),$$

where c_S is the within-subject coefficient of variation, constant for all subjects. As a consequence, y_{ij} will be normally distributed when conditioned on μ_i:

$$y_{ij}|\mu_i \sim \mathcal{N}(\mu_i, (c_S\mu_i)^2). \tag{2.3}$$

This model is a special case of the non-linear mixed model described in Yeap, Davidian (2001) and Davidian, Giltinan (1993). As such, the model will serve as a reference and as validation of the non-linear model.

Moments and correlation

Given the conditional distribution of y_{ij} in equation (2.3), we can calculate the moments as follows: We first apply the law of total expectation, where we condition on μ_i. Thus, the first moment is given by:

$$\mathbb{E}[y_{ij}] = \mathbb{E}[\mathbb{E}[y_{ij}|\mu_i]].$$

Since the inner expectation is equal to μ_i, we get that

$$\mathbb{E}[y_{ij}] = \mathbb{E}[\mu_i] = \mu. \tag{2.4}$$

For the second moment, we use the same strategy and get:

$$\mathbb{E}[y_{ij}^2] = \mathbb{E}[\mathbb{E}[y_{ij}^2|\mu_i]] = \mathbb{E}[\mu_i^2(c_S^2 + 1)],$$

since the second moment of a normal distribution is given by the variance plus the mean squared. Using the same idea with the second moment of μ_i, we get:

$$\mathbb{E}[y_{ij}^2] = \mu^2(c_0^2 + 1)(c_S^2 + 1) = \mu^2(c_0^2c_S^2 + c_0^2 + c_S^2 + 1).$$

This implies that the variance of y_{ij} is given by:

$$\mathbb{V}[y_{ij}] = \mu^2(c_0^2c_S^2 + c_0^2 + c_S^2). \tag{2.5}$$

Higher moments are retrieved in the same manner. As for the correlation between two observations, note that we assumed independence between subjects. This means the following:

$$Cov(y_{i_1j_1}, y_{i_2j_2}) = 0, \text{ if } i_1 \neq i_2. \tag{2.6}$$

We now fix subject i and calculate the covariance between 2 different observations ($j_1 \neq j_2$), as follows:

$$Cov(y_{ij_1}, y_{ij_2}) = \mathbb{E}[(y_{ij_1} - \mu)(y_{ij_2} - \mu)].$$

Note that we have used equation (2.4) in this calculation. We now apply the law of total expectation as before, with the addition that observations of the same subject are independent and identically distributed when conditioned on μ_i:

$$\mathbb{E}[(y_{ij_1} - \mu)(y_{ij_2} - \mu)] = \mathbb{E}[\mathbb{E}[(y_{ij_1} - \mu)(y_{ij_2} - \mu)|\mu_i]] = \mathbb{E}[(\mu_i - \mu)^2].$$

Therefore, it holds that:

$$Cov(y_{ij_1}, y_{ij_2}) = \mathbb{E}[(\mu_i - \mu)^2] = c_0^2\mu^2, \text{ if } j_1 \neq j_2. \tag{2.7}$$

This implies the following for the correlation coefficient between 2 observations of the same subject:

$$\rho_{y_{ij_1}, y_{ij_2}} = \frac{c_0^2}{c_0^2c_S^2 + c_0^2 + c_S^2}. \tag{2.8}$$

2.2 Three-level hierarchical model

It is important to note that the two-level hierarchical model did not separate the intra-subject variability and the analytical variability. To be able to distinguish the two sources, we consider a three-level hierarchical model: we will now consider study designs involving measurements from multiple subjects, repeatedly over time, and replicated at the same time point. In fact, the three-level hierarchical model is not identifiable without replications.

Therefore, we introduce an extra dimension $k \in 1, ..., K(K > 1)$. Now, let y_{ijk} be the k'th measurement on subject i at time point j. We will again assume balance, independence among subjects, and a similar additive structure:

$$y_{ijk} = \mu_i + \delta_{ij} + e_{ijk}, \tag{2.9}$$

where μ_i is the true mean for subject i, δ_{ij} the within-subject effect, and e_{ijk} the analytical error.

We assume again that the subject-specific means to be random samples from a normal distribution:

$$\mu_i \sim N(\mu, (c_0\mu)^2).$$

Similarly, we assume that the within-subject effect is normally distributed and independent, conditionally on the subject-specific means:

$$\delta_{ij}|\mu_i \sim N(0, (c_S\mu_i)^2).$$

We further assume the variance of the analytical error to be proportional to the true value (in the hypothetical assumption of no measurement error):

$$e_{ijk}|\mu_i, \delta_{ij} \sim N(0, (c_R(\mu_i + \delta_{ij}))^2),$$

where c_R is the residual coefficient of variation, constant for all subjects and time points.

Consequently, the conditional distribution of the response variable will be:

$$y_{ijk}|\mu_i, \delta_{ij} \sim \mathcal{N}(\mu_i + \delta_{ij}, (c_R(\mu_i + \delta_{ij}))^2). \tag{2.10}$$

Moments and correlation

In the three-level hierarchical model we are required to use the law of total expectation twice, since the conditional distribution of y_{ijk} is dependent on δ_{ij}, which is by itself dependent on μ_i. This implies for the first moment:

$$\mathbb{E}[y_{ijk}] = \mathbb{E}[\mathbb{E}[\mathbb{E}[y_{ijk}|\mu_i, \delta_{ij}]|\mu_i]].$$

Since the inner expectation is equal to $\mu_i + \delta_{ij}$, and that $\mathbb{E}[\mathbb{E}[\delta_{ij}|\mu_i]] = 0$, we get that

$$\mathbb{E}[y_{ijk}] = \mathbb{E}[\mathbb{E}[\mu_i + \delta_{ij}|\mu_i]] = \mathbb{E}[\mu_i] = \mu. \tag{2.11}$$

For the second moment, we use the same rules and get:

$$\mathbb{E}[y_{ijk}^2] = \mathbb{E}[\mathbb{E}[\mathbb{E}[y_{ijk}^2|\mu_i, \delta_{ij}]|\mu_i]] = \mathbb{E}[\mathbb{E}[(\mu_i + \delta_{ij})^2(c_R^2 + 1)|\mu_i]].$$

Writing the product out further, we get:

$$\mathbb{E}[\mathbb{E}[(\mu_i + \delta_{ij})^2(c_R^2 + 1)|\mu_i]] = (c_R^2 + 1)\mathbb{E}[\mathbb{E}[\mu_i^2 + \delta_{ij}^2|\mu_i]],$$

since the product term $\mu_i\delta_{ij}$ has a conditional expectation of 0. Next, we calculate the inner expectation:

$$(c_R^2 + 1)\mathbb{E}[\mathbb{E}[\mu_i^2 + \delta_{ij}^2|\mu_i]] = (c_R^2 + 1)\mathbb{E}[\mu_i^2(c_S^2 + 1)] = (c_R^2 + 1)(c_S^2 + 1)(c_0^2 + 1)\mu^2,$$

where the last step has been calculated before in the base model. Writing out the latter brackets, we get:

$$\mathbb{E}[y_{ijk}^2] = \mu^2(c_R^2c_S^2c_0^2 + c_R^2c_S^2 + c_S^2c_0^2 + c_R^2c_0^2 + c_R^2 + c_S^2 + c_0^2 + 1)$$

This implies that the variance of y_{ijk} is given by:

$$\mathbb{V}[y_{ijk}] = \mu^2(c_R^2c_S^2c_0^2 + c_R^2c_S^2 + c_S^2c_0^2 + c_R^2c_0^2 + c_R^2 + c_S^2 + c_0^2). \tag{2.12}$$

Higher moments are again retrieved in the same manner. The correlation between observations will be calculated similarly to what has been done for the two-level hierarchical model. We first note that observations between different subjects are uncorrelated:

$$Cov(y_{i_1 j_1 k_1}, y_{i_2 j_2 k_2}) = 0, \text{ if } i_1 \neq i_2. \tag{2.13}$$

We again fix subject i and calculate the covariance between observations on different time points ($j_1 \neq j_2$), by applying equation (2.11), the law of total expectation (twice) and conditional independence on both the time point and repeated measurement level:

$$Cov(y_{ij_1k_1}, y_{ij_2k_2}) = \mathbb{E}[(y_{ij_1k_1} - \mu)(y_{ij_2k_2} - \mu)]$$

$$= \mathbb{E}[\mathbb{E}[(y_{ij_1k_1} - \mu)(y_{ij_2k_2} - \mu)|\mu_i, \delta_{ij_1}, \delta_{ij_2}]]$$

$$= \mathbb{E}[(\mu_i + \delta_{ij_1} - \mu)(\mu_i + \delta_{ij_2} - \mu)] = \mathbb{E}[\mathbb{E}[(\mu_i - \mu)(\mu_i - \mu)|\mu_i]]$$

$$= \mathbb{E}[(\mu_i - \mu)^2] = c_0^2\mu^2.$$

This implies for the correlation coefficient:

$$\rho_{y_{ij_1k_1}, y_{ij_2k_2}} = \frac{c_0^2}{c_R^2c_S^2c_0^2 + c_R^2c_S^2 + c_S^2c_0^2 + c_R^2c_0^2 + c_R^2 + c_S^2 + c_0^2} \tag{2.14}$$

We now fix the time point j as well, and calculate the covariance between 2 repeats ($k_1 \neq k_2$).

$$Cov(y_{ij1k_1}, y_{ijk_2}) = \mathbb{E}[(\mu_i + \delta_{ij} - \mu)^2]. \tag{2.15}$$

Note that we skipped a few steps, since the calculation to this part has been shown before in the derivation of $\rho_{y_{ij_1k_1}, y_{ij_2k_2}}$, where we fix j. Also note that equation (2.15) amounts to the variance of the two-level hierarchical model, given by equation (2.5). Therefore, we have that $Cov(y_{ij1k_1}, y_{ijk_2}) = \mu^2(c_0^2c_S^2 + c_0^2 + c_S^2)$. As such, we have that:

$$\rho_{y_{ijk_1}, y_{ijk_2}} = \frac{c_0^2c_S^2 + c_0^2 + c_S^2}{c_R^2c_S^2c_0^2 + c_R^2c_S^2 + c_S^2c_0^2 + c_R^2c_0^2 + c_R^2 + c_S^2 + c_0^2}. \tag{2.16}$$

2.3 Comparison to normal distribution

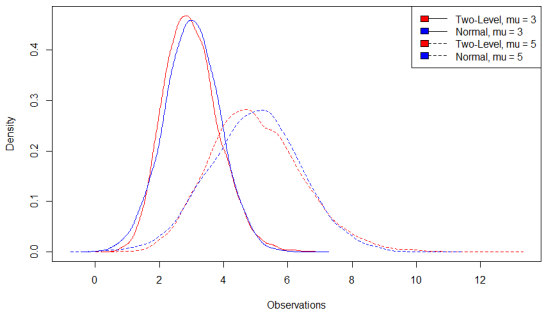
In this section we compare both the base and extended model to random samples from a corresponding normal distribution. To make sure the comparison is meaningful, the following 2 properties must hold for the simulated data:

1. We generate datasets having I subjects and only one measurement each. This is done to exclude the added correlation from the models, which is not accounted for by a random sample.

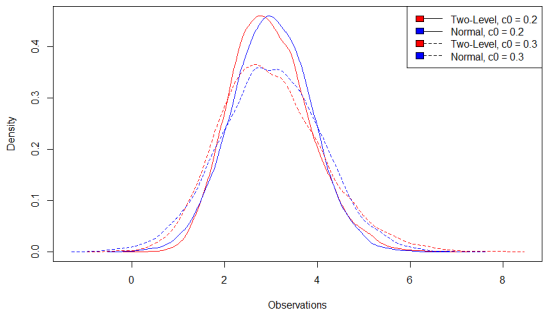
2. The mean and variance of the sampled normal distribution are set to the mean and variance of the (two- and three-level) hierarchical distributions, respectively. In practice, this means that the normal distribution to compare will have mean and variance given by equation (2.4) and (2.5) when compared to the two-level hierarchical model and adheres to equation (2.11) and (2.12) when compared to the three-level hierarchical model.

Furthermore, to get an effective comparison, we set the number of subjects I to 10000. In each plot we fix all the parameters but one, and show the density of the response variable corresponding to two values of the varying parameter. When the parameter value is fixed, its value is meant to be taken $\mu = 5$ and $c_0 = c_S = c_R = 0.2$. Figure 2.1 shows the comparison with the two-level hierarchical model, Figure 2.3 with the three-level hierarchical model. Figures 2.2 and 2.4 show QQ-plots of the different models, under the specified parameter settings.

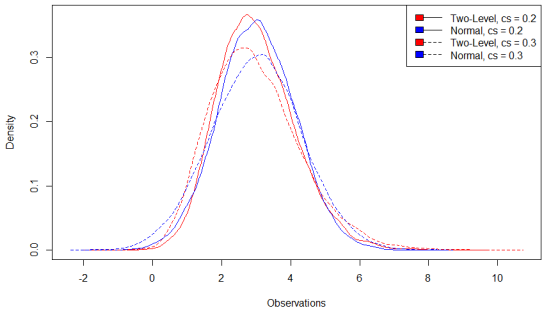
Both models show to have a slight positive skewness when compared to the corresponding normal distribution. Moreover, in most cases, the upper tails seem to be significantly heavier, particularly in the case of the extended model. However, it is difficult to conclude based on the density plots alone that the deviation is significant enough to exclude normality. The QQ-plots do show this deviation more clearly. These plots confirm the heavier upper tails shown in the empirical densities, and show a significant deviation in these tails from the normal distribution, especially in the three-level hierarchical model. Therefore, we conclude that the marginal distribution of both hierarchical models deviates significantly from the normal distribution.



(a) μ

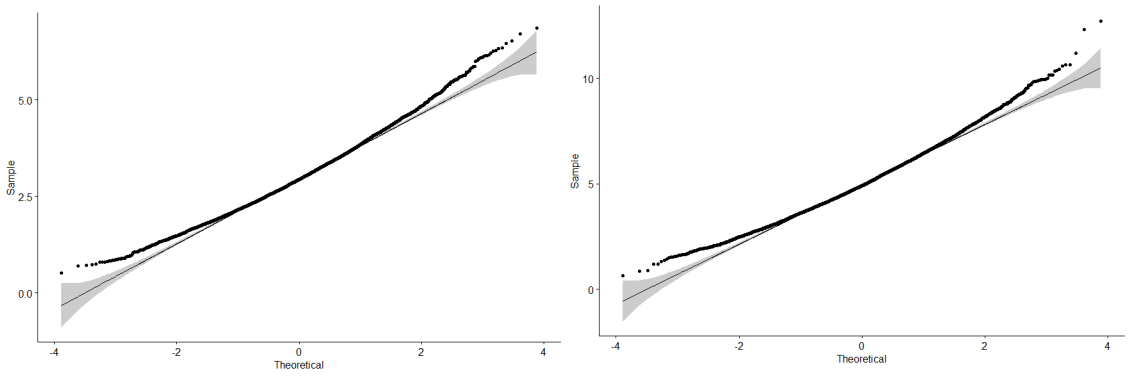


(b) c_0

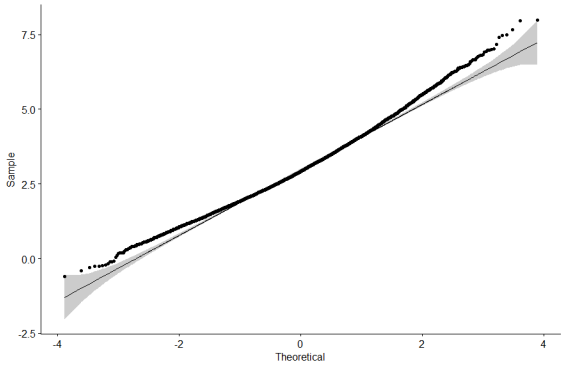


(c) c_S

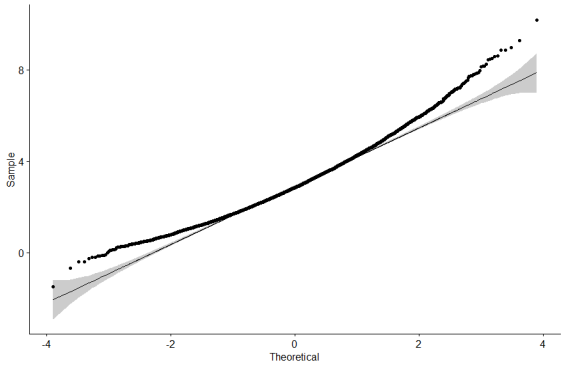
Figure 2.1: Density plots of observations y_{ij} along the two-level hierarchical model (in red), compared to samples from a normal distribution (in blue). In each plot we fix all but one parameter, which is specified in the sub caption. The different line styles indicate the value shifts.



(a) μ

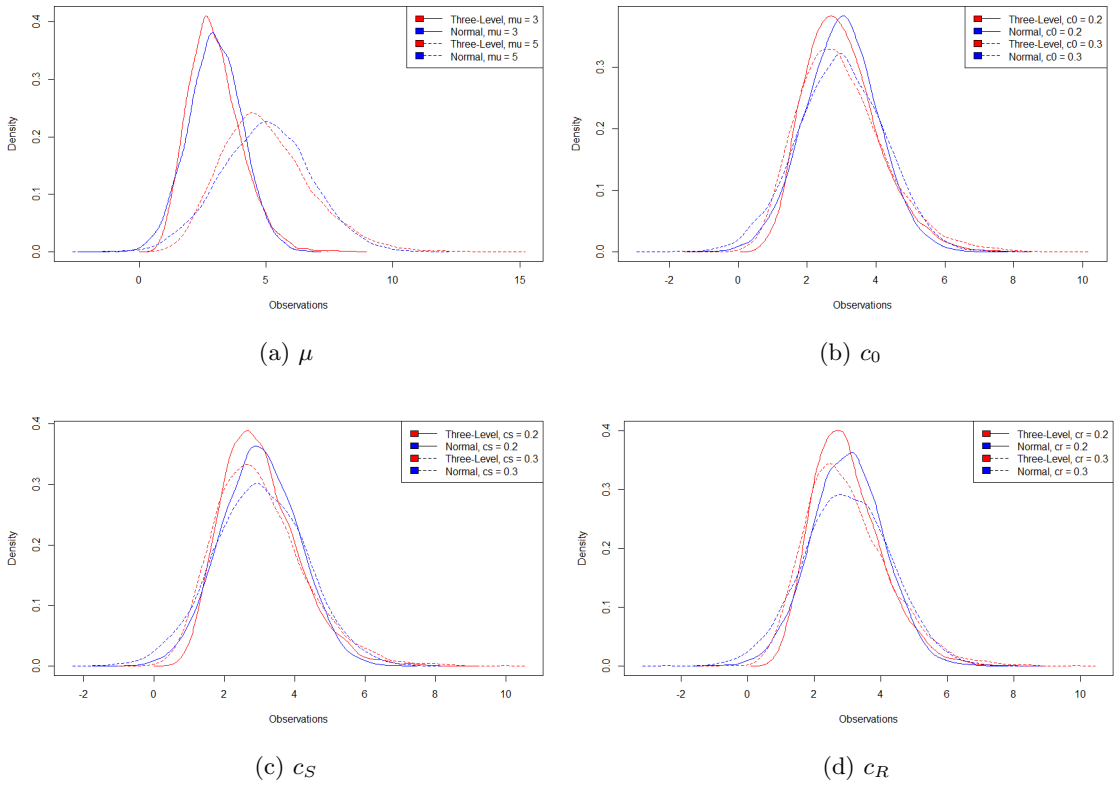


(b) c_0



(c) c_S

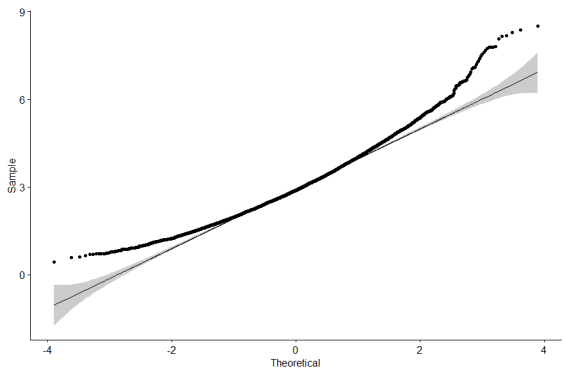
Figure 2.2: QQ plots of observations y_{ij} along the two-level hierarchical model, compared to a normal distribution. The grey area indicates the 95% confidence region of the compared quantiles. The first subplot uses the fixed configuration, the other plots have a shift in one parameter, which is specified in the subcaption.

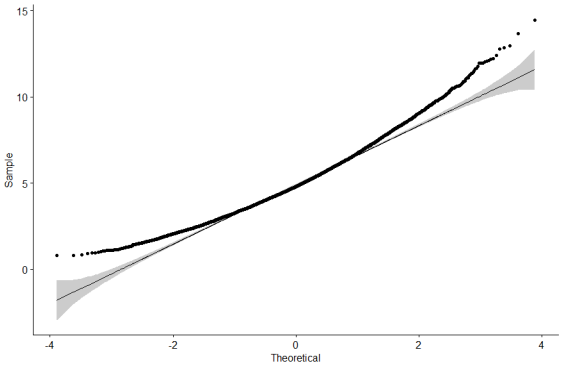


(a) μ (b) c_0

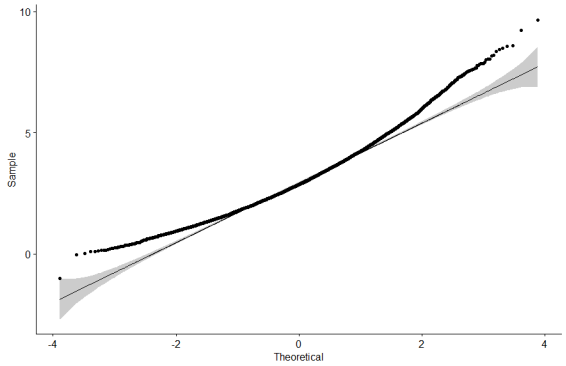
(c) c_S (d) c_R

Figure 2.3: Density plots of observations y_{ijk} along the three-level hierarchical model (in red), compared to samples from a normal distribution (in blue). In each plot we fix all but one parameter, which is specified in the sub caption. The different line styles indicate the value shifts.

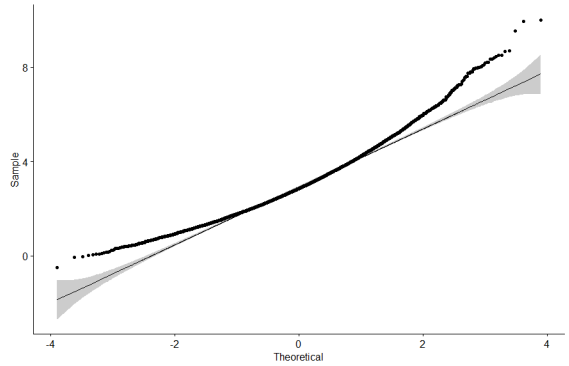




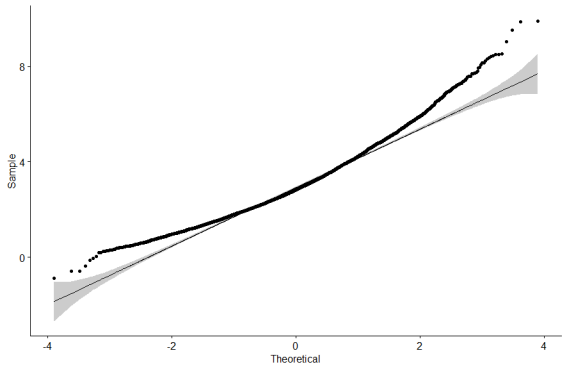
(a) μ



(b) c_0



(c) c_S



(d) c_R

Figure 2.4: QQ plots of observations y_{ijk} along the three-level hierarchical model, compared to a normal distribution. The grey area indicates the 95% confidence region of the compared quantiles. The first subplot uses the fixed configuration, the other plots have a shift in one parameter, which is specified in the subcaption.

Chapter 3

Estimation methods

3.1 Two-level hierarchical model

In this section, we describe some methods to estimate the parameters of the model described in Section 2.1. The methods can in general be separated into two different types; One-step and two-step methods.
We will start with the one-step methods, which does not make use of the apparent hierarchical structure in this model. These methods therefore make use of all the data at once. The two methods of this kind we will look into are the method of moments and maximum likelihood.
The two-step methods first estimate the subject-specific means and the residual coefficient of variation. Then in a second step derives the population parameters (mean and coefficient of variation) from the distributional assumptions on the values estimated in the first step. We adapt the methods described in Davidian, Giltinan (1993) and Yeap, Davidian (2001).

Method of Moments

To estimate the population mean μ, we have to estimate $\mathbb{E}[y_{ij}]$ (as given by equation (2.4)). We have that the sample mean $\frac{1}{IJ}\sum_{i,j} y_{ij}$ is an unbiased estimator of $\mathbb{E}[y_{ij}]$ and therefore of μ.

Next, to estimate the coefficients of variance, we need to calculate the expected mean sum of squares of the model, since the moment estimators consist of linear combinations of these expectations. To retrieve the mean sum of squares, we first define the total sums of squares:

$$SS_0 = J\sum_i(\bar{y}_{i.} - \bar{y}_{..})^2,$$

$$SS_S = \sum_{i,j}(y_{ij} - \bar{y}_{i.})^2,$$

where the dot subscript indicates the average taken over that dimension (for instance, a dot over the first index indicates the average taken over I). We now get the mean sum of squares by dividing the total sum of squares with their respective degrees of freedom:

$$MS_0 = \frac{1}{I-1}SS_0,$$

$$MS_S = \frac{1}{I(J-1)}SS_S.$$

We take the expectation (calculations are found in Appendix A)

$$\mathbb{E}[MS_0] = Jc_0^2\mu^2 + c_S^2(1+c_0^2)\mu^2,$$

$$\mathbb{E}[MS_S] = c_S^2(1 + c_0^2)\mu^2.$$

The moment estimators are now given by:

$$\hat{\mu} = \frac{1}{IJ}\sum_{i,j} y_{ij}, \tag{3.1}$$

$$\hat{c}_0^2 = \frac{MS_0 - MS_S}{J\hat{\mu}^2}, \tag{3.2}$$

$$\hat{c}_S^2 = \frac{MS_S}{\hat{\mu}^2(\hat{c}_0^2 + 1)}. \tag{3.3}$$

Maximum Likelihood

Let $\theta = \{\mu, c_0, c_S\}$ be the vector of parameters to be estimated. Denote the likelihood function as follows:

$$L(\theta|\mathbf{y}) = f(\mathbf{y}|\theta),$$

where $f(\mathbf{y}|\theta)$ is the joint distribution of all observations. Since we know that the subjects are independent, we get:

$$f(\mathbf{y}|\theta) = \prod_i f(\mathbf{y}_i|\theta),$$

with $\mathbf{y}_i$ the vector of measurements belonging to subject i. We now have to determine $f(\mathbf{y}_i|\theta)$ for all i, which is done by using conditional expectation. Since the observations are independent when conditioned on μ_i, we now get that:

$$f(\mathbf{y}_i|\theta) = \int \prod_j f(y_{ij}|\mu_i, \theta) f(\mu_i|\theta) d\mu_i.$$

Given the distribution of $f(\mathbf{y}_i)$, we can now write the likelihood as follows:

$$L(\theta|\mathbf{y}) = \prod_i \int \prod_j f(y_{ij}|\mu_i, \theta) f(\mu_i|\theta) d\mu_i. \tag{3.4}$$

Since the conditional distributions in the likelihood are known, we can rewrite equation (3.4) as follows:

$$L(\theta|\mathbf{y}) = \prod_i \int \prod_j \frac{1}{\sqrt{2\pi c_s^2 \mu_i^2}} e^{-\frac{1}{2}\left(\frac{y_{ij}-\mu_i}{c_s\mu_i}\right)^2} \frac{1}{\sqrt{2\pi c_0^2 \mu^2}} e^{-\frac{1}{2}\left(\frac{\mu_i-\mu}{c_0\mu}\right)^2} d\mu_i.$$

Maximizing the likelihood with respect to θ will give the maximum likelihood estimates of θ. Since the maximization procedure is iterative, we choose the moment estimators from Section 3.1 as starting values. This will increase the speed of convergence, and enhance the possibility of achieving the global maximum.

Two-step methods

In this section we shortly describe the two steps approaches proposed by Yeap, Davidian (2001) and Davidian, Giltinan (1993) by showing how they can be simplified in case of linear mean structure and simple proportional mean-variance relationship.

In the first step, we apply the same model equation as before:

$$y_{ij} = \mu_i + \delta_{ij},$$

but we now treat the means per subject μ_i as fixed values instead of a random variable and estimate $\mu_i, i = 1...I$ and c_S, the coefficient of variation of the residual term. The handling of μ_i as a random variable will occur in the second step, where we apply the following alternative representation of μ_i:

$$\mu_i = \mu + \zeta_i, \tag{3.5}$$

where $\zeta_i \sim \mathcal{N}(0, \zeta)$, with $\zeta = c_0^2\mu^2$. Note the similarities in model structure between the two steps; One fixed term and one variance term. However, different methods need to be applied to estimate the corresponding parameters: note that once we arrive in the second step, we only have estimates for μ_i, while we are dealing with observations in the first step. In the following section we will provide the details on the possible alternatives for each step and on how to adequately deal with an estimated μ_i in the second step.

Step 1: Approximating c_S and μ_i

The two-step methods approach the first step iteratively: They initialize the subject specific means μ_i and the residual coefficient of variance c_S, via an initial approximation. These values are then updated at the nth iteration to the new value $\hat{c}_{S,n}$, $\hat{\mu}_{i,n}$: They first update $\hat{c}_{S,n-1}$ to $c_{S,n}$ using the previous estimates of μ_i, $\hat{\mu}_{i,n-1}$ and consequently adjust the μ_i's to the new coefficient of variation, until convergence. In our case, we first get initial estimates of c_S, by using the moment estimator as described in the Method of Moments, and μ_i, by using the sample means. We call these initial estimates $\hat{c}_{S,0}$ and $\hat{\mu}_{i,0}$ respectively.

To retrieve a new estimate for c_S, $\hat{c}_{S,n}$, Yeap, Davidian (2001) propose robust M-estimators, with the use of estimating equations. In their estimation, they calculate the difference between the true squared deviation and the theoretical variance . These differences are then weighted to take the heterogeneity of variances into account. Applying this concept to our model, the following equation needs to be solved:

$$\sum_{i,j} \frac{1}{\hat{c}_{S,n-1}^4 \hat{\mu}_{i,n-1}^4} \left((y_{ij} - \hat{\mu}_{i,n-1})^2 - \hat{c}_{S,n-1}^2 \hat{\mu}_{i,n-1}^2 \right) = 0. \tag{3.6}$$

Equation (3.6) is solved in $\hat{c}_{S,n-1}$ by applying weighted non-linear least squares to regress the response $(y_{ij} - \hat{\mu}_{i,n-1})^2$ on the mean function $\hat{c}_{S,n-1}^2 \hat{\mu}_{i,n-1}^2$, using weights $\hat{c}_{S,n-1}^4 \hat{\mu}_{i,n-1}^4$, and will yield the new estimate $\hat{c}_{S,n}$.

Note that in the non-linear case, a robust weight to the deviation needs to be added and the gradient as the vector of the derivatives with respect to all the the different variance components needs to be measured. However, due to the linearity of our model and to the simple proportional mean-variance relationship, both of these components are not used here.

Davidian, Giltinan (1993) elect for a likelihood-based approach for updating the estimate of c_S to a new value $c_{S,n}$. Note that the likelihood in this case does not depend on an integral as in our likelihood method, since the subject means are not considered random variables at this stage. In their method, the intent is to minimize a variant of the true log-likelihood $P(\hat{\mu}_{i,n-1}, c_S)$:

$$\hat{c}_{S,n} = \arg\min_{c_S} P(\hat{\mu}_{i,n-1}, c_S). \tag{3.7}$$

Three different functions $P(\hat{\mu}_{i,n-1}, c_S)$ are proposed:

1. Pseudolikelihood (PL):

$$P(\hat{\mu}_{i,n-1}, c_S) = \sum_{i,j} \frac{(y_{ij} - \hat{\mu}_{i,n-1})^2}{c_S^2 \hat{\mu}_{i,n-1}^2} + \log(c_S^2 \hat{\mu}_{i,n-1}^2) \tag{3.8}$$

.

2. Restricted maximum likelihood (REML):

$$P(\hat{\mu}_{i,n-1}, c_S) = \sum_{i,j} \left(\frac{(y_{ij} - \hat{\mu}_{i,n-1})^2}{c_S^2 \hat{\mu}_{i,n-1}^2} + \log(c_S^2 \hat{\mu}_{i,n-1}^2) \right) - I \log c_S^2 + \sum_i \log(J \hat{\mu}_{i,n-1}^2). \tag{3.9}$$

3. Absolute Residuals (AR):

$$P(\hat{\mu}_{i,n-1}, c_S) = \sum_{i,j} \frac{|y_{ij} - \hat{\mu}_{i,n-1}|}{c_S \hat{\mu}_{i,n-1} \eta} + \log(c_S \hat{\mu}_{i,n-1} \eta), \tag{3.10}$$

where η is the expected value of the absolute value of a standard normal distribution, which is $\sqrt{\frac{2}{\pi}}$.

Therefore, there are a total of 4 different methods to acquire a new estimate for c_S.
After getting the new estimate $\hat{c}_{S,n}$, this value is used to update the estimates of μ_i to new values $\hat{\mu}_{i,n}$, by using a least squares approach. A standardized residual is composed for every subject i. This is achieved here by subtracting the observed value with the current estimate for the mean, which is again weighted to account for heterogeneous variances. The I equations are now as follows:

$$\sum_j \frac{1}{\hat{c}_{S,n}^2 \hat{\mu}_{i,n-1}^2} (y_{ij} - \hat{\mu}_{i,n-1}) = 0, i \in 1, ..., I. \tag{3.11}$$

The equations are solved in $\hat{\mu}_{i,n-1}$ by applying weighted non-linear least squares regression on the response y_{ij} on the mean function $\hat{\mu}_{i,n-1}$, using weights $\frac{1}{\hat{c}_{S,n}^2 \hat{\mu}_{i,n-1}^2}$, and yields the new estimate $\hat{\mu}_{i,n}$.
We apply the iteration scheme, until the absolute difference in the updated values compared to the last estimate is smaller than 10^{-5}. When this is achieved, we set the estimates to the final updated values:

$$\hat{c}_S = \hat{c}_{S,n}, \hat{\mu}_i = \hat{\mu}_{i,n} (i \in 1, ..., I).$$

Since the mean structure we use is linear and independent of j, the first step is only iterative in theory. In practice, equation (3.11) gives the subject means as solutions (which are the starting values). Therefore, the only variable that gets updated (and only once) is the coefficient of variation c_S.

Step 2: Approximating c_0 and μ

In this section we describe various approaches that can be used to obtain an estimate of the population mean and coefficient of variation from the subject-specific estimate. In the literature, authors rely on the distributional assumptions in (3.5) and on the assumptions for which each estimator $\hat{\mu}_i$ of μ_i is assumed to be normally distributed, conditionally on the true value:

$$\hat{\mu}_i | \mu_i \sim \mathcal{N}(\mu_i, \hat{C}_i), \tag{3.12}$$

where $\hat{C}_i$ is an estimate of the asymptotic covariance matrix. This estimate is obtained as follows:

$$\hat{C}_i = \frac{1}{I(\hat{\mu}_i)}, \tag{3.13}$$

where $I(\mu_i)$ is the Fisher information matrix. The derivation of this matrix under our modeling assumptions can be found in Appendix B.

Based on equation (3.12) and distributional assumption on μ_i, we have that

$$\hat{\mu}_i \sim \mathcal{N}(\mu, \hat{C}_i + \zeta).$$

Note that from the subject-specific mean estimates, one can derive the population parameters by computing the sample mean and variance:

$$\hat{\mu} = \frac{1}{I}\sum_i \hat{\mu}_i, \tag{3.14}$$

and

$$\hat{\zeta} = \frac{1}{I}\sum_i (\hat{\mu}_i - \hat{\mu})^2. \tag{3.15}$$

Note that from $\hat{\zeta}$ we get the estimate of c_0 as follows:

$$\hat{c}_0 = \sqrt{\frac{\hat{\zeta}}{\hat{\mu}^2}}. \tag{3.16}$$

However, these estimates will be upwardly biased, since they are not corrected by the asymptotic covariance matrix. The methods proposed by Yeap, Davidian (2001) and Davidian, Giltinan (1993) aim to overcome this problem.
Yeap, Davidian (2001) opt for a roughly similar strategy as in the first step: They look at the standardized difference in both the mean and variance component and try to get these values as close as possible to 0. When applying this structure in the second step, the aim is to simultaneously solve these equations:

$$\sum_i (C_i + \zeta)^{-1} \cdot (\hat{\mu}_i - \mu) = 0, \tag{3.17}$$

and

$$\sum_i (\hat{C}_i + \zeta)^{-2} \cdot (\hat{\mu}_i - \mu)^2 - (\hat{C}_i + \zeta)^{-1} = 0. \tag{3.18}$$

The solution to these equations are $\{\hat{\mu}, \hat{\zeta}\}$. We next use equation 3.16 to retrieve the estimate $\hat{c}_0$. Like in the first step, the method they propose uses robust weights in their equations, and solves them iteratively. However, we are able to use simple equation solvers, since we do not deal with vectorized observations. Thus we are able to get solutions directly.

To continue with the method by Davidian, Giltinan (1993), their approach is to estimate the population parameters iteratively.

μ and ζ are first initialized to the values $\hat{\mu}_0$ and $\hat{\zeta}_0$ respectively. To take the asymptotic covariance matrix into account for the estimation of μ, we also initialize μ_i by letting $\mu_{i,0} = \hat{\mu}_i$. We now apply the following iteration:

- We provide new estimates for μ_i, $\hat{\mu}_{i,n}$, by reweighting $\mu_{i,n-1}$ as follows:

$$\mu_{i,n} = (1/\hat{C}_i + 1/\hat{\zeta}_{n-1})^{-1} \cdot (\hat{\mu}_i/\hat{C}_i + \hat{\mu}_{i,n-1}/\hat{\zeta}_{n-1}),$$

  where $\hat{\mu}_i$ is the initial estimate of μ_i, coming from Step 1 of the two-step methods.

- The new estimate for μ, $\hat{\mu}_n$, is retrieved by taking the mean of $\mu_{i,n}$:

$$\hat{\mu}_n = \frac{1}{I}\sum_i \hat{\mu}_{i,n}.$$

- The new estimate for ζ, $\hat{\zeta}_n$, is retrieved by the following equation:

$$\hat{\zeta}_n = \frac{1}{I}\sum_i (\hat{\mu}_{i,n} - \hat{\mu}_n)^2 + \frac{1}{I}\sum_i (1/\hat{C}_i + 1/\hat{\zeta}_{n-1})^{-1} \tag{3.19}$$

- This iteration scheme is applied, until $|\hat{\mu}_n - \hat{\mu}_{n-1}| < 10^{-5}$.

Once convergence is achieved, we get the final estimates $\{\hat{\mu}, \hat{\zeta}\}$. The estimate $\hat{c}_0$ is retrieved by again using equation (3.16).

3.2 Three-level hierarchical model

In this section, we show how the methods described in Section 3.1 can be extended to estimate the parameters of the three-level hierarchical model, described in Section 2.2.

Method of Moments

Like in the two-level hierarchical model, the expected value of each observation y_{ijk} will be equal to μ (as is shown by equation (2.11)). Therefore, we can use the sample mean $\frac{1}{IJK}\sum_{i,j,k} y_{ijk}$ to estimate μ. Next, we employ the same principle and define the total sums of squares:

$$SS_0 = JK\sum_i (\bar{y}_{i..} - \bar{y}_{...})^2,$$

$$SS_S = K\sum_{i,j} (\bar{y}_{ij.} - \bar{y}_{i..})^2,$$

$$SS_E = \sum_{i,j,k} (y_{ijk} - \bar{y}_{ij.})^2.$$

From these the mean squares follow by dividing the sums of squares with their degrees of freedom:

$$MS_0 = \frac{1}{I-1}SS_0,$$

$$MS_S = \frac{1}{I(J-1)}SS_S,$$

$$MS_E = \frac{1}{IJ(K-1)}SS_E.$$

The expected mean squares are (see Appendix A for the calculations):

$$\mathbb{E}[MS_0] = JKc_0^2\mu^2 + Kc_S^2(1+c_0^2)\mu^2 + c_R^2(1+c_s^2)(1+c_0^2)\mu^2,$$
$$\mathbb{E}[MS_S] = Kc_S^2(1+c_0^2)\mu^2 + c_R^2(1+c_s^2)(1+c_0^2)\mu^2,$$
$$\mathbb{E}[MS_E] = c_R^2(1+c_s^2)(1+c_0^2)\mu^2.$$

The moment estimators for μ, c_0, c_S, and c_R are as follows:

$$\hat{\mu} = \frac{1}{IJK}\sum_{i,j,k} y_{ijk}, \tag{3.20}$$

$$\hat{c}_0^2 = \frac{MS_0 - MS_S}{JK\hat{\mu}^2}, \tag{3.21}$$

$$\hat{c}_S^2 = \frac{MS_S - MS_E}{K\hat{\mu}^2(\hat{c}_0^2 + 1)}, \tag{3.22}$$

$$\hat{c}_R^2 = \frac{MS_E}{\hat{\mu}^2(\hat{c}_0^2 + 1)(\hat{c}_S^2 + 1)}. \tag{3.23}$$

Maximum Likelihood

Let $\theta = \{\mu, c_0, c_S, c_R\}$ be the vector of coefficients to be estimated. Using the independence across subjects, we can write:

$$f(\mathbf{y}|\theta) = \prod_i f(\mathbf{y}_i|\theta),$$

where $\mathbf{y}_i$ now represents the vector of all measurements y_{ijk} for subject i. We get the distribution of this quantity by applying conditional expectation on both μ_i and δ_{ij}. We can now apply conditional independence of the repeated measurements and time points to get:

$$f(\mathbf{y}_i|\theta) = \int\int \prod_j \prod_k f(y_{ijk}|\mu_i, \delta_{ij}, \theta) f(\delta_{ij}, \mu_i|\theta) d\delta_{ij} d\mu_i.$$

This gives us the following representation of the likelihood:

$$L(\theta|\mathbf{y}) = \prod_i \int\int \prod_j \prod_k f(y_{ijk}|\mu_i, \delta_{ij}, \theta) f(\delta_{ij}|\mu_i, \theta) f(\mu_i|\theta) d\delta_{ij} d\mu_i. \tag{3.24}$$

Since all distributions in equation (3.24) are known, we can rewrite the equation to:

$$L(\theta|\mathbf{y}) = \prod_i \int\int \prod_j \prod_k \frac{1}{\sqrt{2\pi c_R^2(\mu_i + \delta_{ij})^2}} e^{-\frac{1}{2}\left(\frac{y_{ijk} - (\mu_i + \delta_{ij})}{c_R(\mu_i + \delta_{ij})}\right)^2} \frac{1}{\sqrt{2\pi c_s^2 \mu_i^2}} e^{-\frac{1}{2}\left(\frac{\delta_{ij}}{c_S \mu_i}\right)^2} \frac{1}{\sqrt{2\pi c_0^2 \mu^2}} e^{-\frac{1}{2}\left(\frac{\mu_i - \mu}{c_0 \mu}\right)^2} d\mu_i d\delta_{ij}. \tag{3.25}$$

Maximizing the likelihood in equation (3.25) with respect to θ will give the maximum likelihood estimates of θ. To increase the optimization speed and likelihood of finding a global maximum, we again use the moment estimators as starting values.

Three-step methods

In this section we shortly describe how the two-step methods described in Section 3.1 can be adjusted to fit the three-level hierarchical model.
The idea is to reduce the three-level hierarchical model to the structure of the two-level hierarchical model. That is, we rewrite model (2.9) as

$$y_{ijk} = \gamma_{ij} + e_{ijk}, \tag{3.26}$$

where $\gamma_{ij} = \mu_i + \delta_{ij}$ stands for the true value, without measurement error. Through the steps described in Section 3.1, we will thus be able to estimate c_R and the γ_{ij}'s in the exact same manner as in the 2-step methods.
Step 2 works with the estimates of γ_{ij}:

$$\gamma_{ij} = \mu_i + \delta_{ij}, \tag{3.27}$$

and the goal here is to estimate the μ_i's and the within-subject coefficient of variation c_S. Although this extension is very natural, one needs to take into account the fact that γ_{ij} are estimated: In the two-level model, the response variable is an observation, and thus given. Now we are dealing with an estimated response variable. Like in Step 2 of the 2-step methods in the two-level model, we assume the following: Given the estimator $\hat{\gamma}_{ij}$ of γ_{ij}, we have:

$$\hat{\gamma}_{ij}|\gamma_{ij} \sim \mathcal{N}(\gamma_{ij}, \hat{C}_{ij}),$$

where $\hat{C}_{ij}$ is an estimate of the corresponding asymptotic covariance matrix. We obtain this estimate by calculating the Fisher Information matrix of γ_{ij} and taking the reciprocal. The calculation can be found in Appendix B. Given this matrix, we now have the following:

$$\hat{\gamma}_{ij} \sim \mathcal{N}(\mu_i, c_S^2\mu_i^2 + \hat{C}_{ij}). \tag{3.28}$$

Using equation (3.28), we can now use the same iteration scheme as we used in Step 1 of the 2-step methods to estimate c_S and μ_i. These procedures can be found in Section 3.1. However, the different methods used to update c_S (given by either equation (3.6) or equation (3.7)) differ slightly due to the addition of the asymptotic covariance matrix:

1. Non linear least squares: Now, the response $(\hat{\gamma}_{ij} - \hat{\mu}_{i,n-1})^2$ is regressed on the mean function $\hat{c}_{S,n-1}^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}$ using weights $(\hat{c}_{S,n-1}^2\mu_{i,n-1}^2 + \hat{C}_{ij})^2$,

2. Pseudolikelihood:

$$\hat{c}_{S,n} = \arg\min_{c_S} \sum_{i,j} \frac{(\hat{\gamma}_{ij} - \hat{\mu}_{i,n-1})^2}{c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}} + \log(c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}),$$

3. Restricted maximum likelihood:

$$\hat{c}_{S,n} = \arg\min_{c_S} \sum_{i,j} \frac{(\hat{\gamma}_{ij} - \hat{\mu}_{i,n-1})^2}{c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}} + \log(c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}) - \sum_i \left(\log c_{SI}^2 + \log(J\hat{\mu}_{i,n-1}^2)\right),$$

   where c_{SI}^2 is the mean of $c_{S,n-1}^2 + \hat{C}_{ij}/\hat{\mu}_{i,n-1}$, with the mean taken over the index j. This is required, since the variance of $\hat{\gamma}_{ij}$ cannot be written as a constant times the mean, without being dependent on j. For Restricted maximum likelihood, this independence is required.

4. Absolute Residuals:

$$c_{S,n} = \arg\min_{c_S} \sum_{i,j} \frac{|\hat{\gamma}_{ij} - \hat{\mu}_{i,n-1}|}{\eta\sqrt{c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}}} + \log(\eta\sqrt{c_S^2\hat{\mu}_{i,n-1}^2 + \hat{C}_{ij}}),$$

   with η as before.

From this then, the population parameters are obtained by applying one of the three techniques illustrated in Section 3.1.

Chapter 4

Simulation Study

In this section, we are going to investigate the performance of the methods described in Section 3. We evaluate these methods based on the relative biases of all the estimated parameters.
To do so, we set up a simulation for the two- and three- level hierarchical models. We consider a set of values for each parameter in order to assess the performance of the estimation method in different scenarios.

4.1 Set-up

For the base model, we first choose sets of parameter values for $\{\mu, c_0, c_S\}$ as all the combinations of the following values:

$$\mu \in \{1, 5, 10\}$$

$$c_0 \in \{0.1, 0.2, 0.3\}$$

$$c_S \in \{0.1, 0.2, 0.3\}$$

We thus have 27 different settings. We will generate 1000 datasets per setting, according to our two-level hierarchical model, with $I = 50$ subjects and either $J = 5$ or $J = 10$ time points. We include the latter choice for J to investigate the effect of an increased number of observations per subject on estimation accuracy of the residual coefficient of variation.

On each dataset, we apply all methods that have been described in Section 3.1; the method of moments ((3.1) to (3.3) , maximum likelihood ((3.4), we further call this ML), and all possible combinations of the 2-step methods, where step 1 consists of Non-linear least squares (3.6), PL (3.8), REML (3.9), and AR (3.10). Step 2 consists of applying the sample mean/variance (Sample), the method by Yeap,Davidian (2001) (using (3.17) and (3.18), we further call this YD), and the method by Davidian,Giltinan) (1993) ((3.19), we further call this DG). This leads to 14 different parameter estimates per dataset. From these parameter estimates, we compare the relative bias for each setting, which is calculated as follows:

Let $\hat{a}_i$ be an estimator of variable a on dataset i. The relative bias b_a of the estimator is:

$$b_a = \frac{(\frac{1}{N}\sum_i a_i) - a}{a}. \tag{4.1}$$

For the three-level hierarchical model, we choose the sets of parameters as the combinations of the following values:

$$\mu \in \{1, 5\}$$

$$c_0 \in \{0.1, 0.2\}$$

$$c_S \in \{0.1, 0.2\}$$

$$c_R \in \{0.1, 0.2, 0.3\}$$

We consider small sets of values for μ, c_0 and c_S, to reduce memory issues. We again generate 1000 datasets per setting, this time with $I = 50$ subjects, $J = 5$ time points, and either $K = 3$ or $K = 10$ repeats, to determine the effect of the number of repeats per time point on the same subject.

Also in this case we estimate the parameters with all the estimation methods introduced in Section 3.1 (with all the possible combinations of methods for the two-step procedure) and calculate the relative bias using equation (4.1). However, we do not fit the model with ML, since the computational load is very high compared to the other methods. The R simulation code can be found in appendix C.

4.2 Results

In this section, we present the results of the simulation. First, the relative bias of the estimators has been captured for each generated dataset. Second, the biases have been aggregated by parameter setting, on which the results are plotted in a box plot for each parameter and method separately.

Two-level hierarchical Model

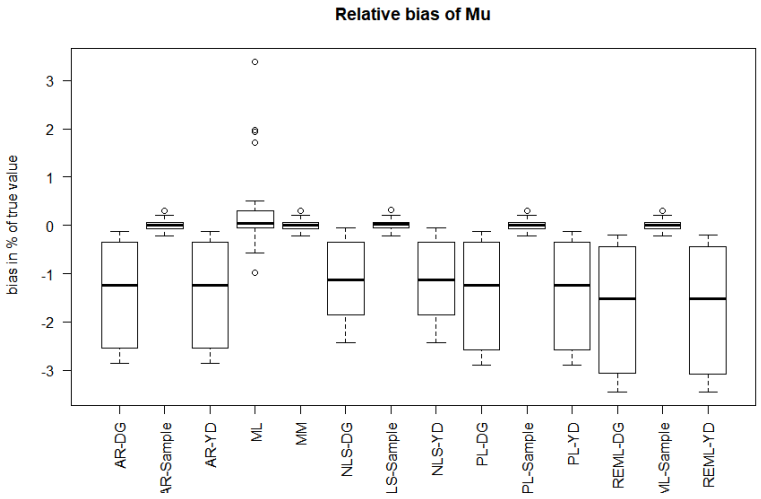


Figure 4.1: Relative bias in the estimation of μ in the two-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, and the coupled abbreviations refer to the choice of the two methods in the two-step procedures.

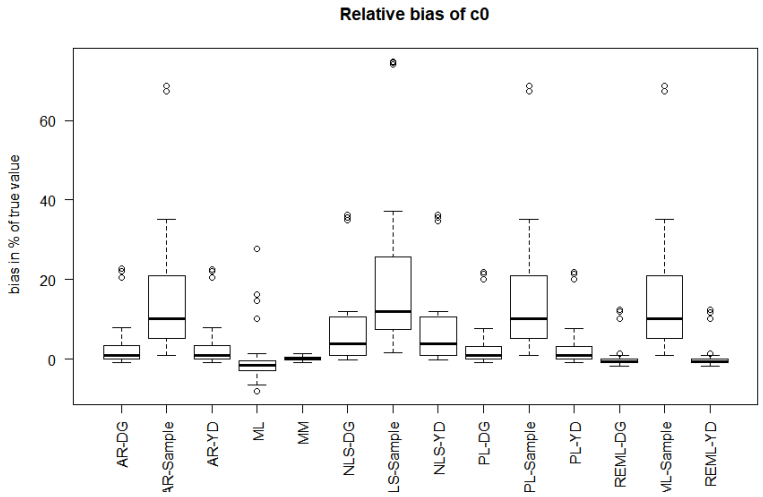


Figure 4.2: Relative bias in the estimation of c_0 in the two-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, and the coupled abbreviations refer to the choice of the two methods in the two-step procedures.

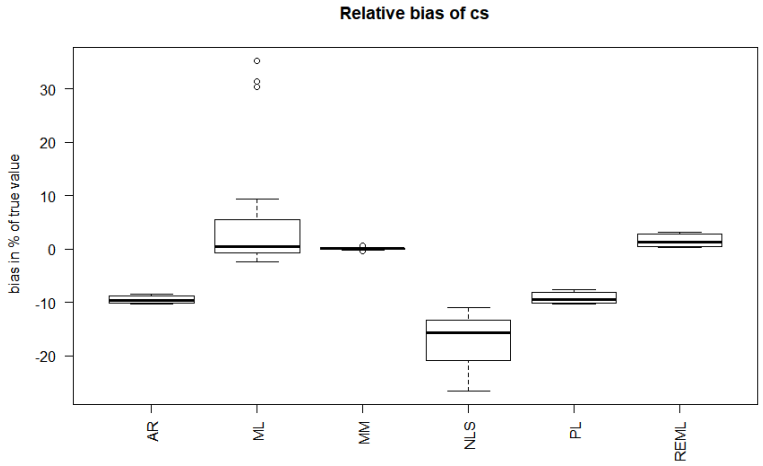


Figure 4.3: Relative bias in the estimation of c_S in the two-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1.

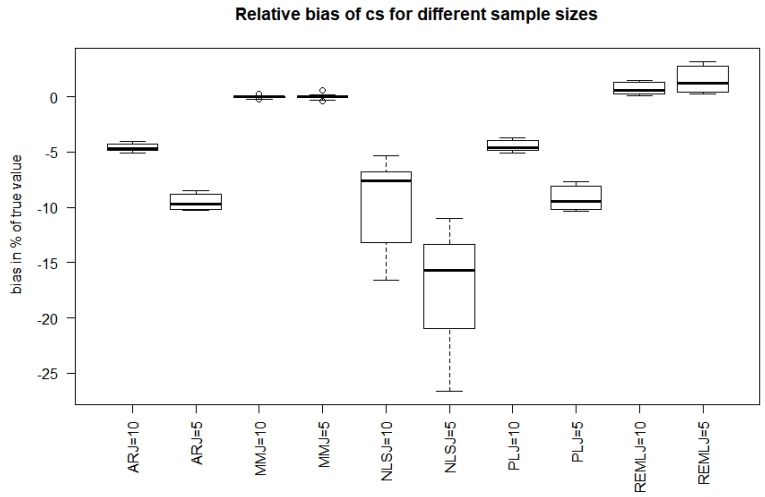


Figure 4.4: Relative bias in the estimation of c_S in the two-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, where the latter part of the abbreviation stand for the number of time points used in the settings.

Three-level hierarchical model

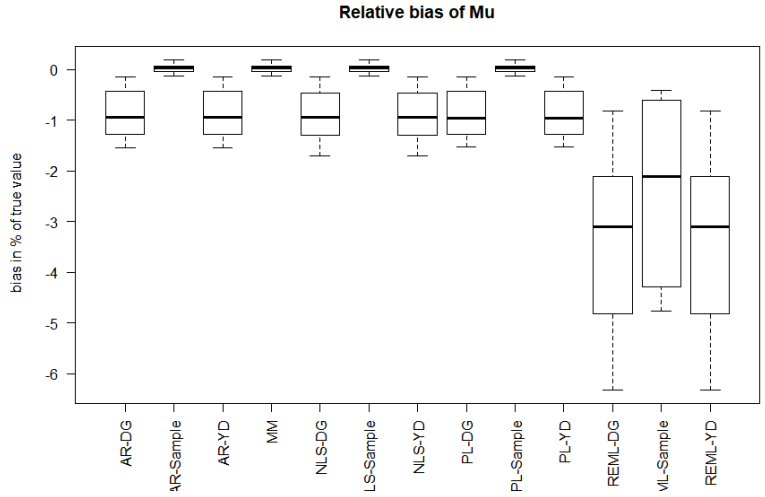


Figure 4.5: Relative bias in the estimation of μ in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, and the coupled abbreviations refer to the choice of the two methods in the two-step procedures.

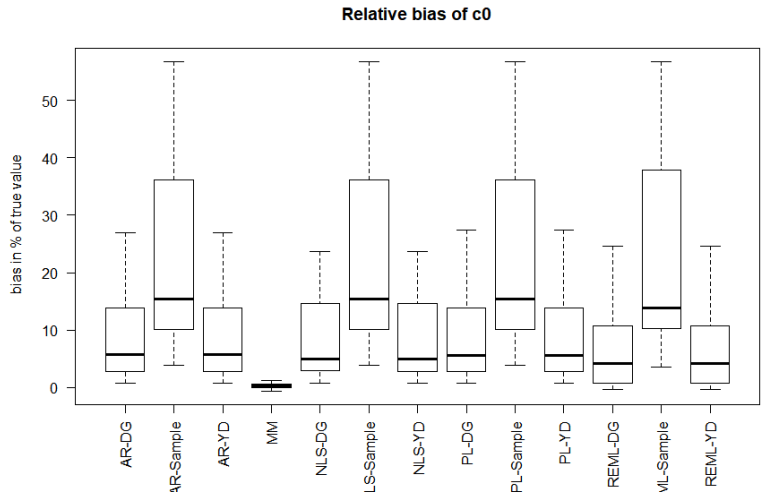


Figure 4.6: Relative bias in the estimation of c_0 in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, and the coupled abbreviations refer to the choice of the two methods in the two-step procedures.

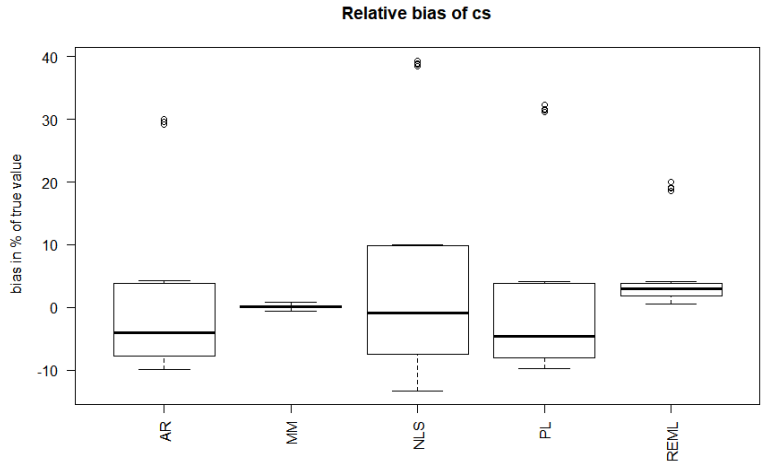


Figure 4.7: Relative bias in the estimation of c_S in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1.

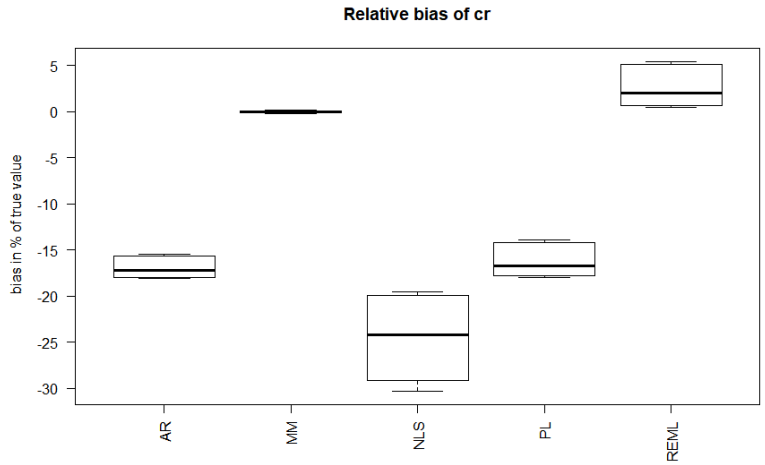


Figure 4.8: Relative bias in the estimation of c_R in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1.

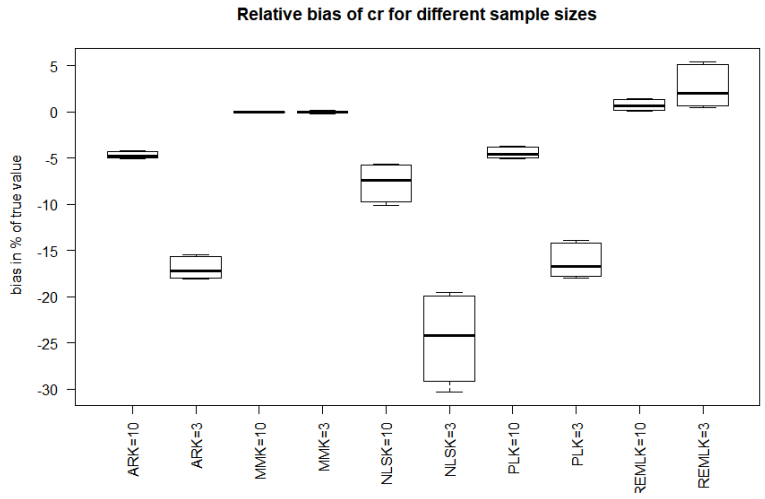


Figure 4.9: Relative bias in the estimation of c_R in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, where the latter part of the abbreviation stand for the number of repeats used in the settings.

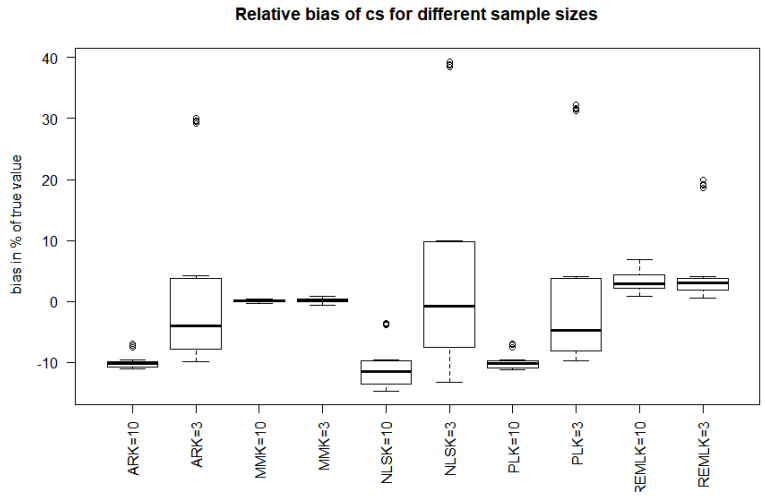


Figure 4.10: Relative bias in the estimation of c_S in the three-level hierarchical model. Each boxplot refers to the results on the estimation with one of the listed estimation methods. The abbreviations are described in Section 4.1, where the latter part of the abbreviation stand for the number of repeats used in the settings.

4.3 Discussion

For the 2-level hierarchical model, the method of moments seems to perform the best and most consistently (see Section 4.2). The relative bias in estimating the parameters is in fact smallest on average and in its variability when compared to all the other methods.

It seems that maximum likelihood performs well on average. This is seen in Figures 4.2 and 4.3, where the median of the relative bias of c_0 and c_S (indicated by the black line in the figures) is roughly near 0, but the spread is significantly larger than with the method of moments.

As for the 2-step methods, there seems to be a significant negative relative bias in c_S (except when choosing REML as first method in the two-step procedures) and a significant positive relative bias in c_0. However, Figure 4.4 shows that as the number of time points increases, the relative bias decreases significantly. It seems that all methods used in Step 1 are asymptotically unbiased. It is advised to use REML however, since it seems to be the most robust against a low amount of data.
For the second step, Figure 4.2 shows that the method of using the sample variance does not work at all in the estimation of c_0, showing the need for correcting for additional variability induced by estimation in the first step. What is more interesting, is that both other options give approximately the same relative bias when compared to each other.
Figure 4.2 also shows some extreme positive relative biases, for all methods. These values occur when c_0 is much smaller than c_S (the setting where $c_0 = 0.1, c_S = 0.3$). One possible cause could be that c_S is generally underestimated, which leads to the issue that the asymptotic covariance matrix is also underestimated, leading to an overestimation of c_0. This compensation is very large relative to the true value of c_0.

The bias in μ is generally very low. It is however noteworthy that the methods which use the sample mean as their estimates perform the best. The main reason for this is that the sample mean is an unbiased estimator of μ alongside our simple linear average structure, which has been shown in the model description.

Also for the three level hierarchical model the method of moments delivers the best estimates. This method is consistently close to being unbiased, even for the relatively low amount of time points and repeats (see Figures 4.5 to 4.8), for the same reason as in the two-level model.

On the other hand, when willing to include covariates or perform tests, the method of moments might incur in limitations. It is thus important to analyze the performance of two-step estimation methods to be considered as an alternative. The bias in c_R (Figure 4.8) shows the effect of having very little data to estimate parameters, when compared to 4.3. This effect is also shown by Figure 4.9, where all methods perform better (in terms of reduced relative bias) with an increased number of repeats. Still, among the two-step procedures, the ones using REML in the first step are shown to perform best.
The relative bias in estimating c_S has more spread than in the two-level hierarchical model (see Figure 4.7). This is sensible, since the true value γ_{ij} is unknown in the extended model and we should take this into account the induced variability in estimating these values (as we have done from the first to the second step in the two-level hierarchical model). More interestingly, if one adds more repeats per time point, the behavior becomes very similar to how it looks like in the two-level model. This shows by comparing Figure 4.3 to 4.10.
The relative biases in μ and c_0 are very similar to that of the two-level model, but some cases perform a bit worse, the most noteworthy being the estimates of μ that used REML in the first step. A probable reason for this is because the estimates in c_S and c_R have shown to be more negatively biased, which are probably compensated in the estimates of the population parameters. Note that the method by Yeap, Davidian (2001) showed some very extreme estimates on a few datasets, leading to very skewed results (for instance a relative bias of 1000%). This mainly occurred when the parameter values for c_0 are low compared to the other variance coefficients, and only occurred for an average of 1 out of 500 datasets. These values have been treated as failed estimations, and are left out of the analysis.

Chapter 5

Conclusions

In this thesis, we have defined a way to extend the model of Regis, Postma, van den Heuvel (2017) to allow for multiple repeated measurements. We included a linear adaptation of Davidian, Giltinan (1993) to use as a reference. We reported several estimation methods, and analyzed the performance of these methods based on the relative bias. Based on these results, the method of moments is the best and most consistent choice. However moment estimators rely on the linearity explicitly, alongside the exclusion of covariates. When one would like to extend these ideas into the mixed model structure, for example by adding covariates or any structure within the residual, the moment estimators need to be adjusted.
However, the results also showed that restricted maximum likelihood is not far off, especially considering the small size of the simulated datasets. Therefore, it is the recommended method when going forward and further extending the ideas of this model.
Note that when one wants to relax the independence assumption to add a relationship between time points or add a structure within the residual, a vectorized approach is required, since the distribution of a single observation becomes more complex and difficult to find.
Also note that the model is not restricted to having 2 or 3 levels. The model can be extended to more hierarchical levels and the estimation methods can consequently be adjusted.

References

1. Davidian, Giltinan; Nonlinear Models for Repeated Measurement Data: An Overview and Update (2003)
2. Davidian, Giltinan; Some Simple Methods for Estimating IntraIndividual Variability in Nonlinear Mixed Effects Models (1993)
3. Yeap, Davidian; Robust Two-Stage Estimation in Hierarchical Nonlinear Models (2001)
4. Kizilkaya, Tempelman; A general approach to mixed effects modeling of residual variances in generalized linear mixed models (2005)
5. Li, Bruyneel, Lesaffre; A multivariate multilevel Gaussian model with a mixed effects structure in the mean and covariance part (2014)
6. Pinheiro, Liu, Wu; Efficient Algorithms for Robust Estimation in Linear Mixed-Effects Models Using the Multivariate t Distribution (2001)
7. Lin, Lee; A robust approach to t linear mixed models applied to multiple sclerosis data (2006)
8. Regis, Postma, van den Heuvel; A note on the calculation of reference change values for two consecutive normally distributed laboratory results (2017)

Appendix A

Calculation of Moments

Two-level hierarchical Model

We are given the sums of squares:

$$SS_S = \sum_{i,j}(y_{ij} - \bar{y}_{i.})^2 = \sum_{i,j}(\delta_{ij} - \bar{\delta}_{i.})^2,$$

$$SS_0 = J \cdot \sum_{i}(\bar{y}_{i.} - \bar{y}_{..})^2 = J \cdot \sum_{i}(\mu_i - \mu_{.} + \delta_{i.} - \bar{\delta}_{..})^2,$$

where the dots indicate the average value taken over the index covered by the dot. We now take expectations:

$$\mathbb{E}[SS_S] = \mathbb{E}[\sum_{i,j}(\delta_{ij} - \bar{\delta}_{i.})^2]$$

$$= \mathbb{E}[\sum_{i,j}(\delta_{ij}^2 - 2\delta_{ij}\bar{\delta}_{i.} + \bar{\delta}_{i.}^2)].$$

Since $\bar{\delta}_{i.} = \frac{1}{J}\sum_j \delta_{ij}$, we get that:

$$\mathbb{E}[\sum_{i,j}(\delta_{ij}^2 - 2\delta_{ij}\bar{\delta}_{i.} + \bar{\delta}_{i.}^2)] = \mathbb{E}[\sum_{i,j}\delta_{ij}^2 - \sum_{i} J\delta_{i.}^2].$$

By the tower rule, we have that $\mathbb{E}[\delta_{ij}^2] = \mathbb{E}[\mathbb{E}[\delta_{ij}^2|\mu_i]] = \mathbb{E}[c_S^2\mu_i^2]$. This simplifies to: $\mathbb{E}[c_S^2\mu_i^2] = c_S^2(c_0^2+1)\mu^2$. Applying the same principle to $\mathbb{E}[\delta_{i.}^2]$, we get that $\mathbb{E}[\delta_{i.}^2] = \frac{1}{J}c_S^2(c_0^2+1)\mu^2$ This implies that:

$$\mathbb{E}[\sum_{i,j}\delta_{ij}^2 - \sum_{i} J\delta_{i.}^2] = IJc_S^2(c_0^2+1)\mu^2 - Ic_S^2(c_0^2+1)\mu^2 = I(J-1)c_S^2(c_0^2+1)\mu^2.$$

We apply the same principles for the other sum of squares:

$$\mathbb{E}[SS_0] = \mathbb{E}[\sum_{i} J(\mu_i - \mu_{.} + \delta_{i.} - \bar{\delta}_{..})^2]$$

$$= \sum_{i} J\mathbb{E}[\mu_i^2] - IJ\mathbb{E}[\mu_{.}^2] + \sum_{i} J\mathbb{E}[\delta_{i.}^2] - IJ\mathbb{E}[\delta_{..}^2]$$

In the same manner, we have that $\mathbb{E}[\mu_i^2] = (c_0^2+1)\mu^2, \mathbb{E}[\mu_{.}^2] = \frac{1}{I}(c_0^2+1)\mu^2$, and $\mathbb{E}[\delta_{..}^2] = \frac{1}{IJ}c_S^2(c_0^2+1)\mu^2$. Combining all, we get:

$$\mathbb{E}[SS_0] = IJ(c_0^2+1)\mu^2 - J(c_0^2+1)\mu^2 + Ic_S^2(c_0^2+1)\mu^2 - c_S^2(c_0^2+1)\mu^2 = (I-1)J(c_0^2+1)\mu^2 + (I-1)c_S^2(c_0^2+1)\mu^2.$$

To conclude, we give the expected mean squares:

$$\mathbb{E}[MS_S] = c_S^2(c_0^2+1)\mu^2,$$

with $I(J-1)$ degrees of freedom, and:

$$\mathbb{E}[MS_0] = J(c_0^2+1)\mu^2 + c_S^2(c_0^2+1)\mu^2,$$

with $I-1$ degrees of freedom.

Three-level hierarchical Model

We now have the following sums of squares:

$$SS_E = \sum_{i,j,k}(y_{ijk} - \bar{y}_{ij.})^2,$$

$$SS_S = K\sum_{i,j}(\bar{y}_{ij.} - \bar{y}_{i..})^2,$$

$$SS_0 = JK\sum_{i}(\bar{y}_{i..} - \bar{y}_{...})^2.$$

The same principals apply as in the base model case, which implies for the expectations:

$$\mathbb{E}[SS_E] = \mathbb{E}[\sum_{i,j,k}(y_{ijk} - \bar{y}_{ij.})^2] = \mathbb{E}[\sum_{i,j,k} e_{ijk}^2 - \sum_{i,j} K\bar{e}_{ij.}^2],$$

$$\mathbb{E}[SS_S] = K\mathbb{E}[\sum_{i,j}(\bar{y}_{ij.} - \bar{y}_{i..})^2] = K\mathbb{E}[\sum_{i,j}\delta_{ij}^2 - \sum_{i} J\bar{\delta}_{i.}^2 + \sum_{i,j}\bar{e}_{ij.}^2 - \sum_{i} J\bar{e}_{i..}^2],$$

$$\mathbb{E}[SS_0] = JK\mathbb{E}[\sum_{i}(\bar{y}_{i..} - \bar{y}_{...})^2] = JK\mathbb{E}[\sum_{i}\mu_i^2 - I\mu_\cdot^2 + \sum_{i}\bar{\delta}_{i.}^2 - I\bar{\delta}_{..}^2 + \sum_{i}\bar{e}_{i..}^2 - I\bar{e}_{...}^2].$$

The calculation of $\mathbb{E}[\delta_{ij}^2], \mathbb{E}[\bar{\delta}_{i.}^2], \mathbb{E}[\bar{\delta}_{..}^2], \mathbb{E}[\mu_i^2], \mathbb{E}[\mu_\cdot^2]$ are all the same as in the base model. As for $\mathbb{E}[e_{ijk}^2]$, we apply the tower rule twice:

$$\begin{aligned}\mathbb{E}[e_{ijk}^2] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[e_{ijk}^2|\mu_i, \delta_{ij}]]] \\ &= \mathbb{E}[\mathbb{E}[c_R^2(\mu_i + \delta_{ij})^2|\mu_i]].\end{aligned}$$

Given that $(\mu_i + \delta_{ij})^2 = \mu_i^2 + 2\mu_i\delta_{ij} + \delta_{ij}^2$, taking the conditional expectation leads to:

$$\mathbb{E}[\mathbb{E}[c_R^2(\mu_i + \delta_{ij})^2|\mu_i]] = c_R^2\mathbb{E}[(c_S^2+1)\mu_i^2],$$

since the product term has 0 expectation. Taking the final part of the tower, we get:

$$\mathbb{E}[e_{ijk}^2] = c_R^2(c_S^2+1)(c_0^2+1)\mu^2.$$

The same calculations apply for $\mathbb{E}[\bar{e}_{ij.}^2], \mathbb{E}[\bar{e}_{i..}^2]$, and $\mathbb{E}[\bar{e}_{...}^2]$.
This implies that:

$$\mathbb{E}[SS_E] = IJ(K-1)c_R^2(c_S^2+1)(c_0^2+1)\mu^2,$$

$$\mathbb{E}[SS_S] = IK(J-1)c_S^2(c_0^2+1)\mu^2 + I(J-1)c_R^2(c_S^2+1)(c_0^2+1)\mu^2,$$

$$\mathbb{E}[SS_0] = JK(I-1)(c_0^2+1)\mu^2 + (I-1)Kc_S^2(c_0^2+1)\mu^2 + (I-1)c_R^2(c_S^2+1)(c_0^2+1)\mu^2.$$

To conclude, we give the expected mean squares:

$$\mathbb{E}[MS_E] = c_R^2(c_S^2+1)(c_0^2+1)\mu^2,$$

with $IJ(K-1)$ degrees of freedom,

$$\mathbb{E}[MS_S] = Kc_S^2(c_0^2+1)\mu^2 + c_R^2(c_S^2+1)(c_0^2+1)\mu^2,$$

with $I(J-1)$ degrees of freedom, and:

$$\mathbb{E}[MS_0] = JK(c_0^2+1)\mu^2 + Kc_S^2(c_0^2+1)\mu^2 + c_R^2(c_S^2+1)(c_0^2+1)\mu^2,$$

with $I-1$ degrees of freedom.

Appendix B

Calculation of Fisher information matrices

Fisher information matrix of μ_i

The Information matrix of μ_i is described as follows:

$$I(\mu_i) = -\mathbb{E}[\frac{\partial^2}{\partial \mu_i^2} \log f(Y_i|\mu_i)],$$

where Y_i is the vector of length J, consisting of y_{ij}. This implies that:

$$\log f(Y_i|\mu_i) = -J/2 \log(2\pi) - 1/2 \log(|\Omega_i|) - 1/2(Y_i - \mu_i)^T \Omega_i^{-1}(Y_i - \mu_i),$$

with Ω_i the $J \times J$ diagonal matrix with $c_S^2 \mu_i^2$ as its entries.
Taking the first derivative:

$$\frac{\partial}{\partial \mu_i} \log f(Y_i|\mu_i) = -1/2 \cdot \frac{1}{|\Omega_i|} \frac{\partial}{\partial \mu_i} |\Omega_i| + 1/2(Y_i - \mu_i)^T \Omega_i^{-1} - 1/2(Y_i - \mu_i)^T \frac{\partial}{\partial \mu_i} \Omega_i^{-1}(Y_i - \mu_i).$$

Given that $\frac{\partial}{\partial \mu_i}|\Omega_i| = |\Omega_i| \cdot tr(\Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i})$ and $\frac{\partial}{\partial \mu_i}\Omega_i^{-1} = -\Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i} \Omega_i^{-1}$, we get that:

$$\frac{\partial}{\partial \mu_i} \log f(Y_i|\mu_i) = -1/2 tr(\Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i}) + (Y_i - \mu_i)^T \Omega_i^{-1} + 1/2(Y_i - \mu_i)^T \Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i} \Omega_i^{-1}(Y_i - \mu_i).$$

We now fill in what we know about Ω_i. We have that $\Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i}$ is a diagonal matrix with entries $\frac{2}{\mu_i}$, such that $tr(\Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i}) = 2\frac{J}{\mu_i}$. It also implies that:

$$1/2(Y_i - \mu_i)^T \Omega_i^{-1} \frac{\partial \Omega_i}{\partial \mu_i} \Omega_i^{-1}(Y_i - \mu_i) = \sum_j \frac{(Y_{ij} - \mu_i)^2}{c_S^2 \mu_i^3},$$

and since:

$$\mathbf{1}(Y_i - \mu_i)^T \Omega_i^{-1} = \sum_j \frac{Y_{ij} - \mu_i}{c_S^2 \mu_i^2},$$

where we multiply with a vector of 1's in order to get the correct dimension, we have that:

$$\frac{\partial}{\partial \mu_i} \log f(Y_i|\mu_i) = -\frac{J}{\mu_i} + \frac{1}{c_S^2} \sum_j \frac{Y_{ij} - \mu_i}{\mu_i^2} + \frac{(Y_{ij} - \mu_i)^2}{\mu_i^3}.$$

Taking the second derivative, we get:

$$\frac{\partial^2}{\partial \mu_i^2} \log f(Y_i|\mu_i) = \frac{J}{\mu_i^2} + \frac{1}{c_S^2} \sum_j \frac{-\mu_i^2 + 2\mu_i(y_{ij} - \mu_i)}{\mu_i^4} + \frac{-2(y_{ij} - \mu_i)\mu_i^3 + 3\mu_i^2(y_{ij} - \mu_i)^2}{\mu_i^6}.$$

Rearranging terms and bringing in one fraction:

$$\frac{\partial^2}{\partial \mu_i^2} \log f(Y_i|\mu_i) = \frac{J}{\mu_i^2} + \frac{1}{c_S^2} \sum_j - \frac{\mu_i^2 + 4\mu_i(y_{ij} - \mu_i) + 3(y_{ij} - \mu_i)^2}{\mu_i^4}.$$

In order to get the Fisher information matrix, we now take the expectation (and flip the sign):

$$I(\mu_i) = -\frac{J}{\mu_i^2} + \frac{1}{c_S^2} \sum_j \frac{\mu_i^2 + 4\mathbb{E}[\mu_i(y_{ij} - \mu_i)] + 3\mathbb{E}[(y_{ij} - \mu_i)^2]}{\mu_i^4}$$

Applying conditional expectations, we get that $\mathbb{E}[\mu_i(y_{ij} - \mu_i)] = 0$, and $\mathbb{E}[(y_{ij} - \mu_i)^2] = c_S^2 \mu_i^2$. This implies that:

$$I(\mu_i) = -\frac{J}{\mu_i^2} + \frac{1}{c_S^2} \sum_j \frac{1 + 3c_S^2}{\mu_i^2}.$$

Fisher information matrix of γ_{ij}

Similar to the Fisher information matrix of μ_i, we have that

$$I(\gamma_{ij}) = -\mathbb{E}[\frac{\partial^2}{\partial \gamma_{ij}^2} \log f(Y_{ij}|\gamma_{ij})],$$

where Y_{ij} is the vector of length K, consisting of y_{ijk}. Similarly, let Ω_{ij} be the $K \times K$ diagonal matrix with $c_R^2 \gamma_{ij}^2$ as its entries.

We have that

$$\log f(Y_{ij}|\gamma_{ij}) = -K/2 \log(2\pi) - 1/2 \log(|\Omega_{ij}|) - 1/2 (Y_{ij} - \gamma_{ij})^T \Omega_{ij}^{-1} (Y_{ij} - \gamma_{ij}).$$

Calculating the first and second derivative consists of the same steps as the Fisher information matrix of μ_i, so:

$$\frac{\partial^2}{\partial \gamma_{ij}^2} \log f(Y_{ij}|\gamma_{ij}) = \frac{K}{\gamma_{ij}^2} + \frac{1}{c_R^2} \sum_k \frac{-\gamma_{ij}^2 + 2\gamma_{ij}(y_{ijk} - \gamma_{ij})}{\gamma_{ij}^4} + \frac{-2(y_{ijk} - \mu_i)\gamma_{ij}^3 + 3\gamma_{ij}^2(y_{ijk} - \gamma_{ij})^2}{\gamma_{ij}^6}.$$

Applying the same steps again, it holds that:

$$I(\gamma_{ij}) = -\frac{K}{\gamma_{ij}^2} + \frac{1}{c_R^2} \sum_j \frac{1 + 3c_R^2}{\gamma_{ij}^2}.$$

Appendix C

Simulation code

For all code, note that since we use the moment estimates of the variation constants, we have to estimate the squares of the constant in the simulation.

Two-level hierarchical model

Data Generation

Note that we used seed 12345 for data generation (in general)

```
#Data generation
DataGeneration <- function(I,J,mu,c0,cs){

Y <- array(rep(0, I*J), c(I,J)) #Observations

for(i in 1:I){
  m <- rnorm(1,mean = mu, sd = sqrt(c0^2*mu^2)) #Mean
  for (j in 1:J){
    v <- rnorm(1, mean = 0, sd = sqrt(cs^2*m^2)) # Variance
    Y[i,j] = m + v
  }

}

return(Y)
}
```

Method of Moments

```
MomentEstimators_TwoLevelModel <- function(Y){

# Dimension retrieval
a <- dim(Y)
I <- a[1]
J <- a[2]

#Setting up means for mean squares
M <- 1/(I*J) * sum(Y)

MV <- apply(Y, MARGIN = 1,FUN=sum) / J

#Set up squares
SV <- array(rep(0, I*J), c(I,J))
SM <- array(rep(0, I), c(I))
for (i in 1:I){
  SM[i] <- (MV[i] - M)^2
```

```
  for (j in 1:J){
    SV[i,j] <- (Y[i,j] - MV[i])^2
  }
}

#Mean Squares
MSV = sum(SV) / (I * (J-1))
MSM = J* sum(SM) / (I-1)

#Parameter estimates
c0 <- max((MSM - MSV)/(J* M^2),0)
cs <- max(MSV/((c0 + 1) * M^2), 0)

return(c(M,c0,cs))
}
```

Maximum Likelihood

```
Integrand <- function(y,mu,c0,cs,m){

  ProductNormal <- function(m){
  N <- array(rep(0, length(y)), dim = length(y))
  for(j in 1:length(y)){
  #when that part of the integrand does not exist
    if(cs*m^2 <= 0 || c0*mu^2 <= 0){
    N[j] <- 1

    }
    else{
    N[j] <- dnorm(y[j], mean = m, sd = sqrt(cs*m^2)) * dnorm(m, mean = mu, sd =
        sqrt(c0*mu^2))

    }
  }



  return(prod(N))
}
return(ProductNormal)



}

ObjFun <- function(params, design){

a <- dim(design)
I <- a[1]
J <- a[2]

mu = params[1]
c0 = params[2]
cs = params[3]

LLs <- array(rep(0, I), I) #All different log likelihood values
for(i in 1:I){
  K <- integrate(Vectorize(Integrand(design[i,], mu,c0,cs, m)), lower = -Inf, upper
      = Inf, stop.on.error=FALSE)$value
  LLs[i] <- -log(K)

}
```

```
return(min(sum(LLs), 1e5))

}

Gradient <- function(params, design){
return(grad(func = ObjFun, x = params, method = "simple", design = design) )
}


MLEst <- function(Y){
return(optim(MomentEstimators_TwoLevelModel(Y), ObjFun, design = Y,  method = "L-
    BFGS-B", gradient = Gradient, lower = c(0,0,0), upper = c(20,1,1), control =
    list(trace = 1,maxit = 15))$par)
}
```

Two-step methods

Step 1: General Iteration

```
Est<- function(Y){

# Dimension retrieval
a <- dim(Y)
I <- a[1]
J <- a[2]


Y2          <- matrix(t(Y),ncol=1)    # from wide to long format
id          <- rep(1:I, each=J)
time        <- rep(1:J, I)
Y3          <- as.data.frame(cbind(id,time,Y2))
colnames(Y3) <- c("id","t","val")

Est_MM <- MomentEstimators_TwoLevelModel(Y)

# step 1: initialize (sample means + moment estimator for the residual coefficient
    of variation)
# muis_0       <- apply(Y, FUN = mean, MARGIN = 1)
muis_0      <- rep(1,I)
cs_0        <- Est_MM[3]

# Iterative procedure for the estimation
c           <- 1
cs_it = Muis_it = list()              # list - estimation of muis and css at
    iteration it

Muis_it[[1]] <- muis_0                # initialize mean values to sample means
cs_it[[1]]   <- cs_0
CONT         <- TRUE

while(CONT){
# step 2: update residual coefficient of variation (NLS)

#Update cs with either Non-linear least squares or maximizing a variant of the
    likelihood

#update mui
ms          <- Muis_it[[c-1]]
r           <- (Y3$val - ms)/cs_it[[c]]
wght        <- v1(r)/(cs_it[[c]] * ms^2)
Y3          <- as.data.frame(cbind(id,time,Y2,wght))
colnames(Y3) <- c("id","t","val","wght")

M = nls(val ~ rep(mui, each=J), data=Y3, start=list(mui=ms), weights=wght, control
    = list(maxiter = 5000))
```

```
Muis_it[[c]] = M$m$getPars()
if(c>1){
  #Only check cs, because the mus do not change
   if(abs(cs_it[[c]]-cs_it[[(c-1)]])<1e-5){
  CONT=FALSE

    }
  }
}
cbind(Muis_it[[c]], apply(Y, FUN=mean, MARGIN=1))


# change name to the variable  to avoid confusions/notation replications
cs <- cs_it[[c]]
mus <- Muis_it[[c]]


# Estimate c0 and mu by either of the 2nd step of the 2-step methods

mu <- Est[1]
c0 <- Est[2]


return(c(mu,c0,cs))
}
```

NLS for estimation of c_S

```
#Fill this in the gap in the general iteration
Ms          <- rep(Muis_it[[c]], each=J)

wght        <-  cs_it[[c]]^2*Ms^4
r           <- (Y3$val-Ms)/cs_it[[c]]
Ymui        <- cbind(Y3, Ms, wght,r)

M           <- nls((v2(r)*(val-Ms)^2)~cs*(Ms)^2, data=Ymui, start=list(cs=cs_it[[c
    ]]), weights = wght,
control = list(maxiter = 5000, tol = 1e-4))

c           <- c+1
cs_it[[c]]  <- M$m$getPars()    # cs value at iteration it
```

Step 1: PL

```
PL = function(cs, params){

Y  = params$Y
MU = params$MU
Vi = matrix(NA, ncol=ncol(Y), nrow=nrow(Y))
for(i in 1:nrow(Y)){
  Vi[i,] = (Y[i,] - MU[i])^2/(cs*MU[i]^2) + log(abs(cs*MU[i]^2))
}
V = apply(Vi, FUN=sum, MARGIN=1)
return(sum(V))

}

#Fill this in the gap in the general iteration
cs_it[[c]] = optim(par=cs_it[[c]], fn=PL, params=list(Y=Y, MU=Muis_it[[c]]), method
    = "Brent",
lower = 0, upper = 10,
control = list(maxit=100))$par
```

Step 1: REML

```
REML = function(cs, params){

Y  = params$Y
MU = params$MU
Vi = matrix(NA, ncol=ncol(Y), nrow=1)
V  = matrix(NA, ncol=1, nrow=nrow(Y))
for(i in 1:nrow(Y)){
  Vi    = (Y[i,] - MU[i])^2/(cs*MU[i]^2) + log(abs(cs*MU[i]^2))
  V[i]  = sum(Vi) - log(cs) + log(det(matrix(1,ncol=ncol(Y),nrow=1)%*%solve(diag(
      rep(MU[i]^2,ncol(Y))))%*%matrix(1,ncol=1,nrow=ncol(Y))))
}
return(sum(V))

}
#Fill this in the gap in the general iteration
cs_it[[c]] = optim(par=cs_it[[c]], fn=REML, params=list(Y=Y, MU=Muis_it[[c]]),
    method = "Brent",
lower = 0, upper = 10,
control = list(maxit=100))$par
```

Step 1: AR

```
AR = function(cs, params){

Y  = params$Y
MU = params$MU
Vi = matrix(NA, ncol=ncol(Y), nrow=nrow(Y))
for(i in 1:nrow(Y)){
  Vi[i,] = abs(Y[i,] - MU[i])/abs(sqrt(2/pi)* sqrt(cs)*MU[i]) + log( abs(sqrt(2/pi)
      * sqrt(cs)*MU[i]))
}
V = apply(Vi, FUN=sum, MARGIN=1)
return(sum(V))

}
#Fill this in the gap in the general iteration
cs_it[[c]] = optim(par=cs_it[[c]], fn=AR, params=list(Y=Y, MU=Muis_it[[c]]), method
    = "Brent",
lower = 0, upper = 10,
control = list(maxit=100))$par
```

Step 2: Sample Means

```
CalculateC0Sample <- function(betas,cs,Est_MM, I){

mu <- mean(betas)
c0 <- var(betas)/mu^2
return(c(mu,c0))
}
```

Step 2: Estimating equation

```
#Asymptotic covariance matric
V = function(mui, J, cs){
Vis = matrix(NA, ncol=1, nrow=J)
for(j in 1:J){
  Vis[j] = (1 + 3*cs)/(mui^2)
```

```
# print(Vis[j])
}
Ii = -J/mui^2 + sum(Vis)/cs            # Fisher information matrix
return(1/Ii)
}

CalculateC0YeapEq <- function(betas, cs, Est_MM, I, J){
library(nleqslv)

C = matrix(NA,nrow = I)
for(i in 1:I){
  C[i] = V(betas[i], J, cs)
}


#Define equation
Eqs <- function(x){
  y <- numeric(2)
  D = x[1]^2 * x[2]
  E1 = E2 = matrix(NA,nrow = I)
  for(i in 1:I){
    E1[i] <- 1/(C[i] + D) * (betas[i] - x[1]) #Equation 3.6
    E2[i] <- (betas[i] - x[1])^2/(C[i] + D)^2 - 1/(C[i] + D) #Equation 3.7
  }

  y[1] <- sum(E1)
  y[2] <- sum(E2)

  y
}
xstart <- c(Est_MM[1], Est_MM[2])
x <- nleqslv(xstart, Eqs,control=list(ftol=1e-9, allowSingular=TRUE, trace = 0),
    method="Broyden")$x



mu <- x[1]
c0 <- max(x[2],0)

return(c(mu,c0))
```

Step 2: Iterative procedure

```
#Asymptotic covariance matrix
V = function(mui, J, cs){
Vis = matrix(NA, ncol=1, nrow=J)
for(j in 1:J){
  Vis[j] = (1 + 3*cs)/(mui^2)
  # print(Vis[j])
}
Ii = -J/mui^2 + sum(Vis)/cs            # Fisher information matrix
return(1/Ii)
}

CalculateC0David <- function(betas,cs,Est_MM, I, J){
Di_it = beta_it = betas_it = list()
beta_it[[1]] <- Est_MM[1]# mu - initially estimated as sample mean of mus, mean(
    betas) works as well
Di_it[[1]] <- var(betas)  different
betas_it[[1]] <- betas
c <- 1
CONT <- TRUE


#Calculate aysmptotic covariance of beta_i
```

```
C = matrix(NA,nrow = I)
for(i in 1:I){
  C[i] = V(betas[i], J, cs)
}
while(CONT){


Ds = matrix(NA, ncol=1, nrow=I)
Dt = matrix(NA, ncol=1, nrow=I)
betas_i = matrix(NA, ncol=1, nrow=I)
for(i in 1:I){

  betas_i[i] <- 1/(1/C[i] + 1/Di_it[[c]]) * (1/C[i] * betas[i] + 1/Di_it[[c]] *
      beta_it[[c]])

}
betas_it[[c+1]] = betas_i
beta_it[[c+1]] = mean(betas_it[[c+1]])


for(i in 1:I){
  Ds[i] = (betas_it[[c+1]][i]-beta_it[[c+1]])^2
  Dt[i] = 1/(1/C[i] + 1/Di_it[[c]])


}
Di_it[[c+1]] = mean(Ds) + mean(Dt)
if(c>2){
  if(abs(beta_it[[c+1]]-beta_it[[c]])<1e-5){
    CONT=FALSE
  }
}
c = c+1
}

mu <- beta_it[[c]]
c0 <- Di_it[[c]]/mu^2

return(c(mu,c0))
}
```

Relative Bias

Note that we also calculate the MSE (hence the name), but do not use it.

```
CalculateMSE <- function(Estimates, values){
MeanMSE <- cbind(mean(Estimates[,1],na.rm = TRUE) - values[1], var(Estimates[,1],na
    .rm = TRUE) + (mean(Estimates[,1],na.rm = TRUE)- values[1])^2)
c0MSE <- cbind(sqrt(mean(Estimates[,2],na.rm = TRUE)) - values[2], var(sqrt(
    Estimates[,2]),na.rm = TRUE) + (sqrt(mean(Estimates[,2],na.rm = TRUE)) - values
    [2])^2)
csMSE <- cbind(sqrt(mean(Estimates[,3],na.rm = TRUE)) - values[3], var(sqrt(
    Estimates[,3]),na.rm = TRUE) + (sqrt(mean(Estimates[,3],na.rm = TRUE)) - values
    [3])^2)

return(cbind(MeanMSE,c0MSE,csMSE))
}
```

Three-level hierarchical model

Data Generation

```
DataGeneration <- function(I,J,K,mu,c0,cs,cr){



Y <- array(rep(0, I*J*K), c(I,J,K)) #Observations

for(i in 1:I){
  m <- rnorm(1,mean = mu, sd = sqrt(c0^2*mu^2)) #Mean
  for (j in 1:J){
    v <- rnorm(1, mean = 0, sd = sqrt(cs^2*m^2)) # Variance
    for(k in 1:K){
      e <- rnorm(1, mean = 0, sd = sqrt(cr^2*(m+v)^2)) #Error
      Y[i,j,k] <- m + v + e
      }
    }
  }

return(Y)
}
```

Method of Moments

```
MomentEstimatorsThreeLevelModel <- function(Y){
#Dimension retrieval
a <- dim(Y)
I <- a[1]
J <- a[2]
K <- a[3]

#Setting up means for mean squares
ME <- array(rep(0, I*J), c(I,J))
MV <- array(rep(0, I), c(I))
M <- 1/(I*J*K) * sum(Y)

for (i in 1:I){
  MV[i] <- rowSums(rowSums(Y, dims = 2), dims = 1)[i] / (J*K)
  for (j in 1:J){
    ME[i,j] <- rowSums(Y, dims = 2)[i,j] /(K)
  }
}

#Set up squares
SE <- array(rep(0, I*J*K), c(I,J,K))
SV <- array(rep(0, I*J), c(I,J))
SM <- array(rep(0, I), c(I))
for (i in 1:I){
  SM[i] <- (MV[i] - M)^2
  for (j in 1:J){
    SV[i,j] <- (ME[i,j] - MV[i])^2
    for(k in 1:K){
      SE[i,j,k] <- (Y[i,j,k] - ME[i,j])^2
    }
  }
}


#Mean Squares
MSE <- sum(rowSums(rowSums(SE, dims = 2), dims = 1))/ (I * J * (K-1));
MSV <- K * sum(rowSums(SV, dims = 1)) / (I * (J-1))
MSM <- J*K * sum(SM) / (I-1)

#Parameter estimates
c0 <- max((MSM - MSV)/(J*K * M^2),0)
cs <- max((MSV- MSE)/ (K * (c0 + 1) * M^2),0)
cr <- max(MSE/(M^2 * (c0 + 1) * (cs + 1) ),0)
```

```
return(c(M,c0,cs,cr))
}
```

Three-step methods

Step 1: General Iteration

```
EstExtended <- function(Y){
#Step 1: Estimate cr and gammas
a<- dim(Y)
I <- a[1]
J <- a[2]
K <- a[3]
Est_MM        <- MomentEstimatorsThreeLevelModel(Y)
Y2            <- array(aperm(Y, c(3,2,1)),dim = I*J*K)  # from wide to long format
id            <- rep(1:I, each=J*K)
time          <- rep(1:J, I, each = K)
reps          <- rep(1:K, I*J)
Y0            <- as.data.frame(cbind(id,time,reps,Y2))
colnames(Y0) <- c("id","t", "reps", "val")

# Initial estimates
Gamma0    <- lm(val~1,data=Y0)                  # ordinary least squares
Gammas0  <- rep(coefficients(Gamma0),I*J)
cr0     <- Est_MM[4]

# Iterative procedure for the estimation
c             <- 1
cr_it = Gammas_it = list()                # list - estimation of muis and css at
    iteration it

Gammas_it[[1]] <- Gammas0                   # initialize mean values to sample means
cr_it[[1]]    <- cr0
CONT          <- TRUE

while(CONT){
#Retrieve an estimate for cr by one of the methods

gs            <- Gammas_it[[c-1]]
wght          <- 1/rep((cr_it[[c]]*gs^2), each=K)
Y3            <- as.data.frame(cbind(id,time,reps,Y2,wght))
colnames(Y3) <- c("id","t","reps","val","wght")

M = nls(val ~ rep(gammaij, each=K), data=Y3, start=list(gammaij=gs), weights=wght)
Gammas_it[[c]] = M$m$getPars()
if(c>1){
if(abs(cr_it[[c]]-cr_it[[(c-1)]])<1e-5) {
CONT <- FALSE
}
}
}

cbind(Gammas_it[[c]], apply(Y, FUN=mean, MARGIN=1))

# change name to the variable  to avoid confusions/notation replications
cr <- cr_it[[c]]
gammas = matrix(Gammas_it[[c]], nrow = I, byrow= TRUE)



#Step 2: Estimate cs and mui

#Calculate asymptotic covariance matrix for each gamma
```

```
C = matrix(NA,nrow = I, ncol = J)
for(i in 1:I){
for (j in 1:J){
C[i,j] = D(gammas[i,j], K, cr)
}
}

#Set everything up again

Y2           <- matrix(t(gammas),ncol=1)    # from wide to long format
id           <- rep(1:I, each=J)
time         <- rep(1:J, I)
Y0           <- as.data.frame(cbind(id,time,Y2))
colnames(Y0) <- c("id","t","val")

# Initial estimates
# Muis0  <- apply(Y, FUN=mean, MARGIN=1)    # sample means per subject
# Muis0  <- rep(1,I)                        # random initial value
Mu0    <- lm(val~1,data=Y0)                # ordinary least squares
Muis0  <- rep(coefficients(Mu0),I)
cs0    <- Est_MM[3]

# Iterative procedure for the estimation
c            <- 1
cs_it = Muis_it  = list()               # list - estimation of muis and css at
    iteration it

Muis_it[[1]] <- Muis0                  # initialize mean values to sample means
cs_it[[1]]   <- cs0
CONT         <- TRUE

while(CONT){


#Update cs with one of the methods


#Update mui
ms           <- Muis_it[[c-1]]
wght         <- 1/rep(cs_it[[c]]*ms^2, each=J)
Y3           <- as.data.frame(cbind(id,time,Y2,wght))
colnames(Y3) <- c("id","t","val","wght")

M = nls(val ~ rep(mui, each=J), data=Y3, start=list(mui=ms), weights=wght)
Muis_it[[c]] = M$m$getPars()
if(c>1){
if(abs(cs_it[[c]]-cs_it[[(c-1)]])<1e-5) {
CONT <- FALSE
}
}
}

cbind(Muis_it[[c]], apply(Y, FUN=mean, MARGIN=1))

# change name to the variable  to avoid confusions/notation replications
cs <- cs_it[[c]]
betas = Muis_it[[c]]
#Step 3: Estimate c0 and mu
#Add one of the steps
mu <- Est[1]
c0 <- Est[2]


return(c(mu,c0,cs, cr ))
}
```

Step 1: NLS

```
#Fill this in the gap in the general iteration
gs          <- rep(Gammas_it[[c]], each = K)
wght        <- cr_it[[c]]^2 * gs^4

Ygij            <- cbind(Y0, gs, wght)
M               <- nls(((val-gs)^2)~(cr*gs^2), data=Ygij, start=list(cr=cr_it[[c]]),
    weights = wght,
control = list(maxiter = 50, tol = 1e-4))
c               <- c+1
cr_it[[c]]    <- M$m$getPars()     # cr value at iteration it
```

Step 2: NLS

```
#Fill this in the gap in the general iteration
Ms              <- rep(Muis_it[[c]], each=J)
G               <- cs_it[[c]]*(Ms)^2 + Cs
wght            <-  G^2

Ymui            <- cbind(Y0, Ms, wght)

M               <- nls(((val-Ms)^2)~cs*(Ms)^2 + Cs , data=Ymui, start=list(cs=cs_it[[c
    ]]), weights = wght,
control = list(maxiter = 50, tol = 1e-4, printEval = FALSE))

c               <- c+1
cs_it[[c]]    <- abs(M$m$getPars())     # cs value at iteration it
```

Step 1: PL

```
PLStep1 = function(cr, params){

Y  = params$Y
a <- dim(Y)
I <- a[1]
J <- a[2]
K <- a[3]
G = params$G
G = matrix(G, nrow = I, byrow= TRUE)
Vij = matrix(NA, ncol= K , nrow=1)
V  = matrix(NA, ncol= J, nrow= I )
for(i in 1:I){
for(j in 1:J){
Vij     = (Y[i,j,] - G[i,j])^2/(cr*G[i,j]^2) + log(cr*G[i,j]^2)
V[i,j]  = sum(Vij)
}

}
return(sum(rowSums(V, dims = 1)))
}

#Fill this in the gap
cr_it[[c]] = optim(par=cr_it[[c]], fn=PLStep1,  params=list(Y=Y, G=Gammas_it[[c]]),
     method = "Brent",
lower = 0, upper = 10,
control = list(maxit=10000))$par
```

Step 2: PL

```
PLStep2 = function(cs, params){

Y  = params$Y
MU = params$MU
C =  params$C
Vi = matrix(NA, ncol = ncol(Y), nrow= nrow(Y))
Csa = matrix(NA, ncol=ncol(Y), nrow=nrow(Y))

for(i in 1:nrow(Y)){
for (j in 1:ncol(Y)) {
Csa[i,j] = cs + C[i,j]/(MU[i]^2)
}
}
for(i in 1:nrow(Y)){
for(j in 1:ncol(Y)){
Vi[i,j] <- (Y[i,j] - MU[i])^2/(Csa[i,j] * MU[i]^2) + log(Csa[i,j] * MU[i]^2)
}
}
V = apply(Vi, FUN=sum, MARGIN=1)
return(sum(V))
}

#Fill this in the gap
cs_it[[c]] = optim(par=cs_it[[c]], fn=PLStep2,  params=list(Y=gammas, MU=Muis_it[[c
    ]], C = C), method = "Brent",
lower = 0, upper = 10,
control = list(maxit=10000))$par
```

Step 1: REML

```
REMLStep1 = function(cr, params){

Y  = params$Y
a <- dim(Y)
I <- a[1]
J <- a[2]
K <- a[3]
G = params$G
G = matrix(G, nrow = I, byrow= TRUE)
Vij = matrix(NA, ncol= K , nrow=1)
V  = matrix(NA, ncol= J, nrow= I )
for(i in 1:I){
for(j in 1:J){
Vij    = (Y[i,j,] - G[i,j])^2/(cr*G[i,j]^2) + log(cr*G[i,j]^2)
V[i,j]  = sum(Vij)- log(cr) + log(K*G[i,j]^2)
}

}
return(sum(rowSums(V, dims = 1)))
}

}


#Fill this in the gap
cr_it[[c]] = optim(par=cr_it[[c]], fn=REMLStep1,  params=list(Y=Y, G=Gammas_it[[c
    ]]), method = "Brent",
lower = 0, upper = 10,
control = list(maxit=10000))$par
```

Step 2: REML

```
REMLStep2 = function(cs, params){

Y  = params$Y
MU = params$MU
C =  params$C
Vj = matrix(NA, ncol=ncol(Y), nrow=1)
Vi = matrix(NA, ncol = 1, nrow= nrow(Y))
Csa = matrix(NA, ncol=ncol(Y), nrow=nrow(Y))
D = matrix(NA, ncol = 1, nrow=nrow(Y))
for(i in 1:nrow(Y)){
for (j in 1:ncol(Y)) {
Csa[i,j] = cs + C[i,j]/(MU[i]^2)
}
}
for(i in 1:nrow(Y)){
for (j in 1:ncol(Y)) {
Vj[j] <- (Y[i,j] - MU[i])^2/(Csa[i,j] * MU[i]^2) + log(Csa[i,j] * MU[i]^2)
}
Vi[i] <- sum(Vj) - log(mean(Csa[i,])) + log(det(matrix(1,ncol=ncol(Y),nrow=1)%*%
    solve(diag(rep(MU[i]^2,ncol(Y))))%*%matrix(1,ncol=1,nrow=ncol(Y))))
}



return(sum(Vi))

}


#Fill this in the gap
cs_it[[c]] = optim(par=cs_it[[c]], fn=REMLStep2,  params=list(Y=gammas, MU=Muis_it
    [[c]], C = C), method = "Brent",
lower = 0, upper = 10,
control = list(maxit=10000))$par
```

Step 1: AR

```
ARStep1 = function(cr, params){

Y  = params$Y
a <- dim(Y)
I <- a[1]
J <- a[2]
K <- a[3]
G = params$G
G = matrix(G, nrow = I, byrow= TRUE)
Vij = matrix(NA, ncol= K , nrow=1)
V  = matrix(NA, ncol= J, nrow= I )
for(i in 1:I){
for(j in 1:J){
Vij    = abs(Y[i,j,] - G[i,j])/abs(sqrt(2/pi)* sqrt(cr) *G[i,j]) + log( abs(sqrt(2/
    pi) * sqrt(cr) *G[i,j]))
V[i,j]  = sum(Vij)
}

}
return(sum(rowSums(V, dims = 1)))
}
#Fill this in the gap in the general iteration

cr_it[[c]] = optim(par=cr_it[[c]], fn=ARStep1,  params=list(Y=Y, G=Gammas_it[[c]]),
     method = "Brent",
lower = 0, upper = 10,
```

```
control = list(maxit=10000))$par
```

Step 2: AR

```
ARStep2 = function(cs, params){

Y  = params$Y
MU = params$MU
C =  params$C #Covariance matrix
Vi = matrix(NA, ncol = ncol(Y), nrow= nrow(Y))
Csa = matrix(NA, ncol=ncol(Y), nrow=nrow(Y))

for(i in 1:nrow(Y)){
for (j in 1:ncol(Y)) {
Csa[i,j] = cs + C[i,j]/(MU[i]^2)
}
}
for(i in 1:nrow(Y)){
for(j in 1:ncol(Y)){
Vi[i,j] = abs(Y[i,j] - MU[i])/abs(sqrt(2/pi)* sqrt(Csa[i,j])*MU[i]) + log( abs(sqrt
    (2/pi) * sqrt( Csa[i,j])*MU[i]))
}




}
V = apply(Vi, FUN=sum, MARGIN=1)
return(sum(V))


}

#Fill this in the gap
cs_it[[c]] = optim(par=cs_it[[c]], fn=ARStep2,  params=list(Y=gammas, MU=Muis_it[[c
    ]], C = C), method = "Brent",
lower = 0, upper = 10,
control = list(maxit=10000))$par
```

Step 3

Is the exact same as Step 2 in the two-step methods.

Relative Bias

Note again that we also calculate the MSE (hence the name), but do not use it.

```
CalculateMSEExtended <- function(Estimates, values){
MeanMSE <- (cbind(mean(Estimates[,1],na.rm = TRUE) - values[1])/values[1], var(
    Estimates[,1],na.rm = TRUE) + (mean(Estimates[,1],na.rm = TRUE)- values[1])^2)
c0MSE <- (cbind(sqrt(mean(Estimates[,2],na.rm = TRUE)) - values[2])/values[2], var(
    sqrt(Estimates[,2]),na.rm = TRUE) + (sqrt(mean(Estimates[,2],na.rm = TRUE)) -
    values[2])^2)
csMSE <- (cbind(sqrt(mean(Estimates[,3],na.rm = TRUE)) - values[3])/values3, var(
    sqrt(Estimates[,3]),na.rm = TRUE) + (sqrt(mean(Estimates[,3],na.rm = TRUE)) -
    values[3])^2)
crMSE <- (cbind(sqrt(mean(Estimates[,4],na.rm = TRUE)) - values[4])/values[4], var(
    sqrt(Estimates[,4]),na.rm = TRUE) + (sqrt(mean(Estimates[,4],na.rm = TRUE)) -
    values[4])^2)
return(cbind(MeanMSE,c0MSE,csMSE,crMSE))
}
```