Joran Jongerling

Utrecht University, Utrecht

There and Back: Modeling Inter-Individual Differences in Intra-Individual Variability

THERE AND BACK: MODELING INTER-INDIVIDUAL DIFFERENCES IN INTRA-INDIVIDUAL VARIABILITY

DAARHEEN EN TERUG: MODELLEREN VAN INTER-INDIVIDUELE VERSCHILLEN IN INTRA-INDIVIDUELE VARIABILITEIT
(met een samenvatting in het Nederlands)

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof.dr. G.J. van der Zwaan, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op vrijdag 9 december 2016 des ochtends te 10.30 uur

door

Joran Jongerling

geboren op 25 juli 1982
te Rotterdam

Promotor: Prof.dr. H.J.A. Hoijtink
Copromotor: Dr. E.L. Hamaker

Beoordelingscommissie:

Prof.dr. G. van Breukelen
Prof.dr. C.V. Dolan
Prof.dr. C. Finkenauer
Prof.dr. C.A.W. Glas
Prof.dr. F.J. Oort

Jongerling, Joran
There and Back: Modeling Inter-Individual Differences in Intra-Individual Variability
Proefschrift Universiteit Utrecht, Utrecht. - Met lit. opg. - Met samenvatting in het Nederlands.
ISBN 978-90-393-6668-4
Druk: Printservice Ede
Cover design by Joran Jongerling
Copyright © 2016, J. Jongerling. All Rights Reserved

For Lennie, my parents, and my grandparents.
Thank you so much for everything!

Acknowledgements

Now that my PhD is finished, it's about time I thanked all the people without whom, this dissertation would not have come to be. First, I'd like to thank my supervisors. Ellen, thank you for your many contributions to this dissertation and for all the editing that made me look like a decent writer. Herbert, I'm forever grateful for your guidance and support, and for the great discussions during our meetings. I also think we really were on to something with this lets-have-my-job-evaluation-in-a-different-country-every-year thing.

I would like to thank all my current and former colleagues at the M&T Department of Utrecht University for all the small talk, projects, pranks, conversations, outings, and other activities we shared. They made my time as a PhD a lot more enjoyable than it would have been otherwise. A couple of you I feel deserve special mention. Charlotte, Jesper, Noémi, and Erline, you were the best roommates I could have wished for, and I will always have fond memories of our time together. My fellow 'first generation' PhD friends: Joris, Rens, Gerko, Peter, Carel, Floryt, and Rebecca. Together with Kevin, Jesper, Linda, and Charlotte we made for one of the tightest, weirdest, funniest bunch of researchers-in-the-making that I ever saw! Thank you for all the fun times! And Kevin, thank you for all the impromptu Basket visits. Nijs, you I would like to thank for always being there for a talk, a joke, an idea, or anything else I might need or not need. It has been invaluable to me. Finally, I would like to thank Cora Maas for not only setting me up on the path of becoming, what I hope, is a passionate teacher, but for all your friendship and advice as well.

I would also like to thank my new colleagues at the Department of Pedagogical and Educational Sciences at the Erasmus University in Rotterdam. Thank you all for your warm welcome. I immediately felt at home with you guys, and really consider you to be an extended family of sorts.

I am also grateful to my family for all their support. I especially want to thank my parents and my grandparents. You supported me and helped me in al conceivable ways my whole life, and your are always there for me. I cannot begin to list what you've done for me, just like I cannot tell you enough how much I love you, and how lucky I consider myself for being your (grand)son. Of course, I also cannot leave out

my ‘parents-in-law’ here. I really hit the jackpot with you guys, and I really treasure your friendship and support.

Last but not least, the love of my life, Lennie. Where do I begin? You are my life, truly my whole world. Without you I would not have been able to do this, and I definitely would not be where I am today. You have no idea how much you do for me, and mean to me, and there is nothing that I want more than to grow old together. Thank you for making this ‘weird boy’ the happiest guy ever!

Rotterdam, April 2015
Joran Jongerling

Contents

1

Introduction

1.1 Intra-individual Differences in Process Dynamics

When it comes to processes, an important distinction can be made between *stable processes* and *developmental processes*.

Stable processes can be roughly defined as processes that are characterized by within-person reversible variability over time in the absence of a gross underlying trend (Nesselroade, 1991). Examples include the interaction between dyadic partners during a conversation, or individuals' daily fluctuations in affect. While, all of us have some days on which we feel more positive than usual, and some days on which we feel more negative, these positive and negative deviations revolve around a constant, or base, level of affect.

Developmental processes, on the other hand, are characterized by structural change over time, with intra-individual variability occurring around an (individual's) mean trend or growth curve. Examples of these type of processes are the increase in students' math ability across their time in school, or the decrease in depressive symptoms resulting over the course of a successful form of therapy. Here, there is not only moment to moment or day to day variation in how depressed an individual feels, but there is also a systematic decrease, or downward trajectory, in the average, or base, depression level of that individual.

For both types of processes, inter-individual differences in the underlying characteristics are especially important, since these differences, captured by inter-individual differences in model parameters, can be indicative of systematic variations in regulatory mechanisms, coping, and differential sensitivity and/or exposure to both modeled and unmodeled factors. However, stable and developmental processes require different sampling intensities for the study of inter-individual differences in model parameters.

J. Jongerling wrote the introduction.

This dissertation therefore discusses methods for studying inter-individual differences in both intensively and non-intensively sampled data.

1.2 Part 1: Modeling Inter-Individual differences with Intensively Sampled Data.

Models for the study of stable processes are usually based on assessing the amount of autoregression between successive measurement occasions, and modeling inter-individual differences in autoregression requires intensive data, that is, data collected over a large number of repeated measurements (that are usually close together in time), like the data obtained from diary studies. Models for these type of data are presented in the first part of this dissertation. Specifically, we will focus on a *dynamic multilevel modeling* (DMM) (Jongerling, Laurenceau, & Hamaker, 2015), that is, a model based on modeling the repeated measures of an individual at level 1 using a time series model, while allowing for individual differences in the model parameters at level 2. Specifically we will use a two level first-order autoregressive AR(1) model with random means, random autoregression, and random innovation variance (i.e., the level 1 residual variance). Note that this model is more extensive than the models usually described in the literature, as it also allows for individual differences in error variance.

Chapter 2 focuses on the various model parameters of our DMM and their interpretation. In addition, it provided an explanation of what inter-individual differences in the model parameters may reflect, and why they could be of interest to psychological researchers. Specific attention is given to inter-individual differences in the residual variance. Finally, Chapter 2 will also compare Maximum Likelihood estimation of our DMM to Bayesian estimation, and investigate the trade-off between the number of individuals N and the number of time points T in the estimation of the two level AR(1) model.

In Chapter 3, an expression for the total variance of our DMM is derived. This expression allows for a more detailed look at the variance structure of the model, and can be used to determine how (inter-individual differences in) the model parameters influence the total variance. Subsequently, this expression for the total variance is used to derive equations for some interesting proportions of explained variance, such as, the proportion of explained variance on level 1, the proportion of explained variance on level 2, the total proportion of explained variance, and the proportion of variance explained by autocorrelation. Furthermore, it is also used to derive an expression for the Intra-Class Correlation (ICC) in a two level AR(1) model.

1.3 Part 2: Modeling Inter-Individual differences with Non-Intensively Sampled Data.

In case intensively sampled data is not available, other models can provide valuable insights into the longitudinal process under investigation. These models are presented in the second part of the dissertation and focus more on developmental change and systematic change trajectories. Special attention will be given to the the autoregressive latent trajectory (ALT) model that was introduced by Curran and Bollen (2001); Bollen and Curran (2004), and which combines a latent growth curve (LGC) model (Meredith & Tisak, 1990; Curran & Bollen, 2001; Bollen & Curran, 2004) with an autoregressive model (Jöreskog, 1971, 1979).

Chapter 4 of the thesis examines two different methods suggested by Curran and Bollen (2001) to deal with the recursion inherent to the ALT model. Specifically, the effect of these "start-up" methods on the model parameters will be investigated.

Chapters 5 and 6 of this thesis, are aimed at enabling applied researchers to analyze their own data with either the ALT model, or a number of other models suitable for the study of both stable and developmental processes. Chapter 5 describes a multi-center randomized trial cost-effectiveness study for the treatment of personality disorders. This study is used as an applied example for researchers, that outlines how the Deviance Information Criterion (DIC) (Spiegelhalter, Best, Carlin, & Linde, 2002) can be used to select the best fitting model, and how this best fitting model should be interpreted.

Chapter 6 subsequently provides applied researchers with all the materials needed for analysis of their data. Specifically, it provides code for the Bayesian estimation of a two level first-order autoregressive model, a linear growth curve model, a quadratic growth curve, and a autoregressive latent trajectory models. In addition, it provides a detailed manual that gives step-by-step instruction on how to prepare data for analysis with one of the provided models, and on how to run the code for the actual analysis.

Part I

A Two Level First-Order Autoregressive Model

2

A Multilevel AR(1) Model: Allowing for Inter-Individual Differences in Trait-Scores, Inertia, and Innovation Variance

Summary. In this paper we consider a multilevel first-order autoregressive (AR(1)) model with random intercepts, random autoregression, and random innovation variance (i.e., the level 1 residual variance). Including random innovation variance is an important extension of the multilevel AR(1) model for two reasons. First, between-person differences in innovation variance are important from a substantive point of view, in that they capture differences in sensitivity and/or exposure to unmeasured internal and external factors that influence the process. Second, using simulation methods we show that modeling the innovation variance as fixed across individuals, when it should be modeled as a random effect, leads to biased parameter estimates. Additionally, we use simulation methods to compare Maximum Likelihood estimation to Bayesian estimation of the multilevel AR(1) model and investigate the trade-off between the number of individuals and the number of time points. We provide an empirical illustration by applying the extended multilevel AR(1) model to daily positive affect ratings from 89 married women over the course of 42 consecutive days.

2.1 Introduction

Over the past few decades, there has been a growing interest in the study of processes as they unfold over time. This is accompanied by an increased need for longitudinal

This chapter is accepted for publication as Jongerling, J., Laurenceau, J-P., & Hamaker, E.L. (2015). A Multilevel AR(1) Model: Allowing for Inter-Individual Differences in Trait-Scores, Inertia, and Innovation Variance. *Multivariate Behavioural Research.*

J. Jongerling wrote the article and the computer code needed for the analyses. He also ran all the analyses.

J-P. Laurenceau provided the data for the applied example and provided feedback on the article.

E.L. Hamaker supervised the study and provided feedback on the analyses and the article.

models that both capture the essence of these intra-individual processes, as well as allow for investigating any individual differences therein. While the study of *developmental processes* has blossomed with the introduction of techniques like latent growth curve modeling (Meredith & Tisak, 1990; Bollen & Curran, 2004, 2006) and latent transition models (Schmittmann, Dolan, Maas, & Neale, 2005), the statistical techniques for studying *stable processes* have only recently started to gain the attention of a wider audience of psychological researchers.

Stable processes can be roughly defined as processes that are characterized by within-person reversible variability over time in the absence of a gross underlying trend (Nesselroade, 1991). Examples include individuals' daily fluctuations in affect or the interaction between dyadic partners during a conversation. An innovative and promising modeling approach to the study of stable processes is *dynamic multilevel modeling*, which is based on modeling the repeated measures of an individual at level 1 using a time series model, while allowing for individual differences in the model parameters at level 2.

Suls, Green, and Hillis (1998) were the first to use this approach. They used a first-order autoregressive (AR(1)) model at level 1, in which each observation is regressed upon the preceding observation using an autoregressive parameter. The part that cannot be predicted from the previous observation is referred to as the *innovation* (also known as perturbation, random shock, or residual). Suls et al. (1998) conceptualized the autoregressive parameter as a measure of spillover or carryover, as it indicates the degree to which prior states affect current states. They also proposed an interpretation of the autoregressive parameter as a measure of *inertia*, because the further it is away from zero, the longer it takes the individual to restore equilibrium after being perturbed by an innovation. Hence, the autoregressive parameter can be thought of as indicating a person's regulatory weakness, being inversely related to attractor strength (Hamaker, 2012). At the second level of their dynamic multilevel model, Suls et al. (1998) established a positive relationship between inertia and neuroticism and a negative relationship between inertia and agreeableness. Recently, this innovative work by Suls et al. (1998) has received attention from Kuppens and his colleagues, who have performed a series of studies focused on emotional inertia, showing that there is a positive relationship between inertia and depression (Kuppens, Allen, & Sheeber, 2010), that emotional inertia prospectively predicts the onset of depression (Kuppens et al., 2012), and that emotional inertia is related to rumination, although both factors separately contribute to depression (Koval, Kuppens, Allen, & Sheeber, 2012).

Across all these studies, the models were characterized by a random intercept and a random autoregressive (i.e., inertia) parameter, while the residual variance at level 1 (i.e., the innovation variance) was restricted to be the same across individuals. In contrast, Wang, Hamaker, and Bergeman (2012) considered a multilevel AR(1) model that also included a random innovation variance to allow for individual differences in this aspect of the process. However, they did not consider this issue in depth, neither from a substantive nor from a statistical point of view. Therefore, the current paper is focused on the need of including a random innovation variance in the multilevel

AR(1) model. We will argue that individual differences in residual variances are meaningful from a substantive point of view and may contain important information about regulatory processes. We then investigate what the effect is of ignoring this potential source of individual differences in a simulation study. In addition, we consider the trade-off between the number of observations within each person, and the number of people in the sample, as this is clearly of interest to applied researchers.

The remainder of this paper is organized as follows. First, we introduce a multilevel AR(1) model that allows for individual differences in means, inertias, and innovation variances. Second, we discuss five different estimation methods for this model – three Maximum Likelihood (ML) based methods that can be run using standard multilevel software and two Bayesian methods that can be run using WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000). Third, we present a simulation study in which the five estimation methods are compared. Fourth, we provide an illustration in which we apply the multilevel AR(1) model to an empirical data set, consisting of 42 daily emotion ratings obtained from a larger questionnaire administered to 96 married couples in a study on intimacy in marriage (for details, see Laurenceau, Feldman Barrett, & Rovine, 2005). We end with a discussion of the findings and recommendations for applied researchers who wish to make use of this model in their own work.

2.2 A multilevel AR(1) model

In this section, we begin with presenting the level 1 or *within-person part* of the multilevel AR(1) model. This is comprised of an AR(1) model (cf., Hamilton, 1994; Chatfield, 2003) that can be expressed in two ways. We discuss the roles of the various model components in substantive terms. This is followed by the presentation of the level 2 or *between-persons part* of the multilevel AR(1) model which allows us to model individual differences in the level 1 parameters. We explain what such differences may reflect, and why they could be of interest to psychological researchers.

2.2.1 Level 1: Within-person

An AR(1) process can be expressed by using either one or two equations. Below, we present both, and discuss how the expressions are related. Let y_{it} be the observed score of individual i at time point t. If we express the AR(1) process with a single equation, we regress the observed score directly on the preceding score, that is

$$y_{it} = c_i + \phi_i y_{i,t-1} + \epsilon_{it}, \tag{2.1}$$

where c_i is the individual's intercept (i.e., the expected score, when $y_{i,t-1} = 0$), ϕ_i is the AR-parameter, and ϵ_{it} is the unpredictable part, also referred to as the innovation, residual, or random shock. It is assumed that ϕ_i lies between -1 and 1 to ensure stationarity (that is, a situation in which the mean and variance of the process do not change over time, see Hamilton, 1994; Chatfield, 2003). Furthermore, it is

assumed that the innovations are independent and normally distributed with mean 0 and variance σ_i^2.

Alternatively, when using the two equations specification, we can think of the individual's score as consisting of two parts: a mean score μ_i, which represents an individual's trait score (i.e., his/her long-run tendency, equilibrium, or long-term preferred state) and a temporal deviation from this mean, which we denote as ζ_{it}, that is

$$y_{it} = \mu_i + \zeta_{it}. \tag{2.2}$$

The temporal deviations (or states) themselves also may be characterized by autocorrelation and can be modeled with the AR(1) model

$$\zeta_{it} = \phi_i \zeta_{i,t-1} + \epsilon_{it}. \tag{2.3}$$

The two models expressed above are simply reparametrizations of each other, meaning that the actual process they describe is exactly the same. The equivalence between these two expressions can be seen by relating the mean in Equation 2.2 to the intercept in Equation 2.1 through

$$\mu_i = \frac{c_i}{1 - \phi_i}, \tag{2.4}$$

which is a standard result in time series literature (cf., Hamilton, 1994; Chatfield, 2003). Despite the equivalence of the two expressions, we feel that the latter two equation specification is the model that researchers would typically want to estimate, because it provides estimates of the AR parameter, innovation variance, and the mean (instead of a less meaningful intercept). With standard maximum likelihood software, however, this model cannot be estimated, because both equations need to be combined into one. As a result, maximum likelihood estimation leads to the single equation specification and an inability to to model individual means directly.

The inertia parameter ϕ_i in Equations 2.1 and 2.3 reflects the degree to which previous scores or states carry over into current scores or states. Suppose we have a number of daily measurements of negative affect for an individual. If the inertia parameter is close to zero, this implies that there is little or no carry over from the level of negative affect yesterday on the level of negative affect today. In contrast, when the inertia parameter is close to 1, this implies that an increased level of negative affect yesterday is likely to persist into today (and subsequent days), while decreased levels also tend to persist for several days. This is where the interpretation of inertia comes from.

The innovation ϵ_{it} represents the part of the process that cannot be predicted based on previous scores or states. Thus, it can be thought of as the collection of all unobserved (or omitted) factors that influence the process under investigation. For instance, today's negative affect not only depends on yesterday's negative affect, but also on sleep quality, recent stress experiences, caffeine and alcohol intake, hormonal levels, social obligations and interactions, et cetera. Furthermore, individuals may be

more or less sensitive to these factors. While it is possible to include measurements of such factors in our level 1 model (e.g., Suls et al., 1998), and to model an individual's sensitivity to such a factor (e.g., Wichers et al., 2009, 2010; Wichers, Lothmann, Simons, Nicolson, & Peeters, 2012; Wichers, Peeters, et al., 2012), there will always be additional factors that influence the process but that were not observed and therefore cannot be modeled explicitly. These effects are absorbed into the innovation term, and thus influence the innovation variance parameter σ_i^2.

2.2.2 Level 2: Between-person

The fixed-effect parameters of the within-person part of the model, that is the mean μ_i, the inertia ϕ_i, and the innovation variance σ_i^2, may be characterized by individual differences, which we can model at level 2 by including random effects. Before presenting the level 2 model, we take a more detailed look at the effect of individual differences in the level 1 parameters. To this end, we make use of four simulated AR(1) processes which are presented in Figure 2.1.

To draw links between the parameters of the model and the behaviour of the outcome, there are several aspects of Figure 2.1 that are worth noting. First, the two upper panels of this Figure show that differences in means are just indicative of differences in the the vertical position of the series (i.e., the equilibria or preferred states) of different individuals, but that they do not give any information about the individual dynamics. Second, comparing the two left panels, it can be seen that differences in the autoregressive parameter result in differences in the dynamics, that is, the pattern of fluctuations over time, as well as in the amount of total variance on the outcome variable. An AR-parameter closer to 1 (e.g., .9 for lower-left panel) leads to more carryover and therefore less random fluctuations over time and a wider range of fluctuations than an AR-parameter closer to 0 (e.g., .2 for upper-left panel). Third, comparing the upper-left panel with the lower-right panel, we can see that differences in innovation variances also result in differences in observed variances: When the innovation variance is larger (e.g., 3 for the upper-left panel), the total variance for the observations is also larger compared to the case where the innovation variance is smaller (e.g., 1 for the lower-right panel), despite the AR-parameter being the same.

The fact that both the innovation variance σ^2 and the AR-parameter ϕ affect the total variance also becomes apparent from the relationship between these three characteristics (e.g., Hamilton, 1994; Chatfield, 2003), that is

$$\psi_i^2 = \frac{\sigma_i^2}{1 - \phi_i^2}, \tag{2.5}$$

where ψ_i^2 is the variance of the observed variable for individual i.

The need to allow for individual differences in means is obvious: Different individuals have different trait levels or preferred states, and this can be captured by individual differences in μ_i. In addition, the importance of allowing for individual differences in the AR-parameter has been the focus of a small number of studies,

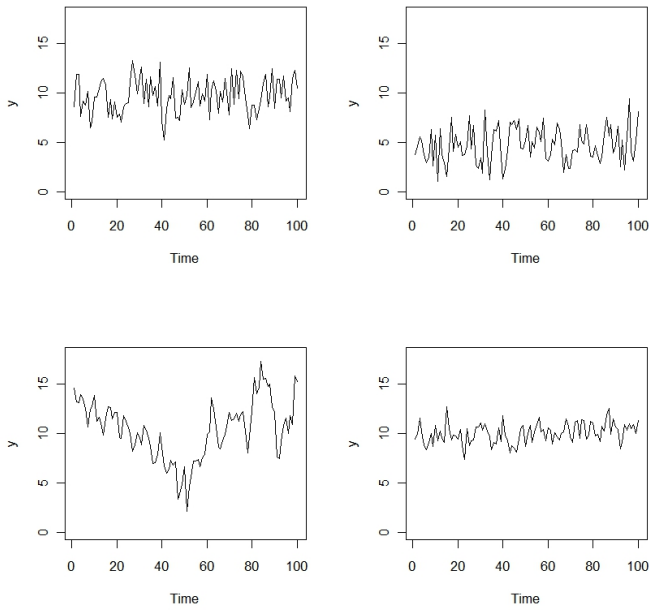


Fig. 2.1: Time series with different means, inertias, and innovation variances: The series in the upper-left panel is characterized by a mean of 10, an AR-parameter of .2, and an innovation variance of 3; the series in the upper-right panel is characterized by a mean of 5, an AR-parameter of .2, and an innovation variance of 3; the series in the lower-left panel is characterized by a mean of 10, an AR-parameter of .9, and an innovation variance of 3; and the series in the lower-right panel is characterized by a mean of 10, an AR-parameter of .2, and an innovation variance of 1.

which have shown that this measure of inertia can be meaningfully related to other person characteristics such as gender (Rovine & Walls, 2006), neuroticism (Suls et al., 1998; Wang et al., 2012), depression (Kuppens et al., 2010; Koval et al., 2012), and rumination (Koval et al., 2012). Furthermore, Koval and Kuppens (2012) have shown that inertia can be state-dependent, and that it decreases more under stress in persons vulnerable to stress than in others. In addition, it has been shown that inertia can prospectively predict the onset of depression in adolescence (Kuppens et al., 2012) and health outcomes (Wang et al., 2012). Note that for most psychological processes individuals will be characterized by a positive AR-parameter, but this is not necessarily the case. For example, Rovine and Walls (2006) found that daily drinking behavior of some individuals was actually characterized by a negative AR-parameter. This implies a regulatory process that follows a saw tooth pattern: Days on which

these persons drank more than their average, are typically followed by days on which they drink less than their average, and vice versa.

The possibility of individual differences in the innovation variance has been largely ignored in the literature thus far. While Wang et al. (2012) included a person-specific innovation variance, they have not considered the need for this potentially important feature of the model in depth. From a substantive point of view, we would like to argue that the existence of individual differences in innovation variances are to be expected for two reasons. First, there are probably individual differences in the range of fluctuation of the unobserved or omitted factors that influence the process under investigation, and this can be reflected by individual differences in the innovation variance. Second, individuals are likely to differ from each other with respect to their responsiveness to such factors. For instance, Rottenberg (2005) has indicated that depressed individuals are characterized by *emotion context insensitivity*, that is, a reduced emotional responsiveness to the environment, while Wichers et al. (2010) have shown that depressed individuals respond less strongly to positive events and more strongly to negative events. These types of individual differences in responsiveness or sensitivity to unobserved factors can also be reflected by individual differences in innovation variance.

Therefore, from a substantive point of view, we believe individual differences in innovation variance are likely the norm, rather than the exception. Relating individual differences in innovation variance to other individual differences is likely to help us to obtain more insight in the process under investigation, and in the possible sources of individual differences. Furthermore, there is also a statistical motivation for including the innovation variance as a random effect. Although ignoring random effects typically does not bias the estimated of the fixed effects in multilevel models (cf. Hox, 2010), we believe that in the current context the situation may be different because the observed variance is a function of both the innovation variance and the AR-parameter (as shown in Equation 2.5). If the innovation variance is not allowed to vary across individuals (i.e., $\sigma_i^2 = \sigma^2$), individual differences in the variance of the observed process (ψ_i) can only be accounted for by individual differences in the AR-parameter (ϕ_i). Thus, ignoring the possibility of individual differences in the innovation variance may lead to bias in the estimate of the AR-parameter.

With this line of reasoning in mind, we decided to specify the individual means, AR-parameters, and innovation variances as random effects, which may also be related to each other using a multivariate normal distribution, that is

$$\begin{bmatrix} \mu_i \\ \phi_i \\ \sigma_i^2 \end{bmatrix} \sim MVN\left(\begin{bmatrix} \mu \\ \phi \\ \sigma^2 \end{bmatrix}, \begin{bmatrix} \tau_\mu^2 & & \\ \tau_{\mu\phi} & \tau_\phi^2 & \\ \tau_{\mu\sigma^2} & \tau_{\phi\sigma^2} & \tau_{\sigma^2}^2 \end{bmatrix} \right). \tag{2.6}$$

where MVN stands for multivariate normal. The correlated random effects imply that the parameters may be influenced by the same unobserved person characteristics. For

Instead of using the innovation variance in the multivariate normal distribution, we could have decided to use the logarithm of this variance to ensure that no negative variances can occur. However, we believe this to be less intuitive than considering the variance itself,

instance, Wichers et al. (2009) showed that the affective experience of individuals suffering from depression was more sensitive to the occurrence of negative events, whereas Kuppens et al. (2010) illustrated that depressed individuals are characterized by stronger emotional inertia. If negative events are not explicitly measured and included as a predictor of affect, the effect of negative events will be absorbed into the innovation term, leading to a larger innovation variance for more sensitive (i.e., depressed) individuals. As a result, the effect of depression on both the AR-parameter and the innovation variance will lead to a positive correlation between these two random effects at the second level.

It should be noted that the level 2 model presented here is very basic, but could be readily extended to include level 2 predictors such as neuroticism or gender (see Wang et al., 2012).

2.3 Estimation

In this section, two estimation approaches are considered that can be used to estimate a multilevel AR(1) model. ML estimation using standard multilevel software and Bayesian estimation using WinBUGS. Thus far, most studies employing a multilevel AR(1) model used ML estimation in standard multilevel software. We also consider Bayesian estimation methods as a way to overcome some of the limitations associated with the use of standard multilevel software.

2.3.1 ML Estimation with standard multilevel software

There are two problems associated with estimating the multilevel AR(1) model defined in Equations 2.2, 2.3 and 2.6 with standard multilevel software. First, most multilevel software packages only allow for the single equation formulation at level 1, such that defining the model as in Equations 2.2 and 2.3 is not possible. Second, by default the level 1 residual variance (i.e., the innovation variance) is identical across individuals in standard multilevel software. We elaborate on both limitations below.

Focusing on the first limitation, in standard multilevel software, the researcher cannot use our preferred specification of the AR-process at level 1 based on the two equations (Equations 2.2 and 2.3). Instead, the equivalent specification in Equation 2.1 can be used, but this has the disadvantage that it includes estimation of the intercept c_i, not the mean μ_i: As explained above, the intercept c_i is generally less interesting and intuitive from a substantive point of view than μ_i. Based on the relationship expressed in Equation 2.4, we can derive an estimate of μ_i, if we first obtain individuals' shrinkage estimates for c_i and ϕ_i (Bryk & Raudenbush, 1992; Hox, 2010). However, these shrinkage estimates are only available after model estimation, and as a result, one could only model individual differences in means by conducting a second modeling step.

---

and moreover we do not expect computational problems because innovation variances are expected to be clearly larger than zero in the data.

As a solution to the inability to model individuals' mean scores, we could center the predictor $y_{i,t-1}$ per person: This implies we are centering level 1 predictor variables within the higher level units, and as a result the level 1 intercepts are equivalent to the level 1 cluster (i.e., person) means on the outcome variable (cf., Kreft, Leeuw, & Aiken, 1995; Enders & Tofighi, 2007). The reason for this now follows. Since an individual's true mean on $y_{i,t-1}$ is identical to his/her mean on y_{it} (i.e., it is the individual's mean over time μ_i), the person-mean centered lagged predictor can be written as $y_{i,t-1} - \mu_i = \zeta_{i,t-1}$. Using this centered predictor, and making use of the fact that $c_i = \mu_i(1 - \phi_i)$, we can write

$$\begin{aligned} y_{it} &= c_i + \phi_i\mu_i + \phi_i(y_{i,t-1} - \mu_i) + \epsilon_{it} \\ &= \mu_i(1 - \phi_i) + \phi_i\mu_i + \phi_i\zeta_{i,t-1} + \epsilon_{it} \\ &= \mu_i + \phi_i\zeta_{i,t-1} + \epsilon_{it}, \end{aligned} \tag{2.7}$$

which shows that person-mean centering does indeed allow for the direct modeling of μ_i with a single equation specification of an AR(1) process at level 1. However, there is a problem with this approach. To center $y_{i,t-1}$, we need μ_i, but we don't know the value of this parameter. In fact, the aim is to estimate μ_i using Equation 2.7 (i.e., the whole purpose of expressing the model as in Equation 2.7 is to obtain an estimate of μ_i). In short, person-mean centering requires simultaneously estimating both an individual's mean and already knowing it so it can be used for the actual centering the lagged predictor, which is obviously impossible,

We will consider two solutions to the "catch-22" we are in. First, we simply compute an individual's sample mean (i.e., the ordinary least squares estimate), and use this as an estimate for μ_i. This conforms to the usual approach when using cluster-mean centering in multilevel modeling. Second, we will consider a two-step procedure, where we begin with an empty model, with the level 1 model being: $y_{it} = \mu_i + \zeta_{it}$. From this model, shrinkage estimates of the individuals' μ_i are obtained, which are then used to center the predictor variable at level 1, such that in the second step the model in Equation 2.7 can be estimated. These two procedures are referred to as *MLpc1* and *MLpc2* respectively (where pc stands for person-centered). In addition, we will also consider estimation based on Equation 2.1, that is, without centering the predictor, which we refer to as *MLuc*. In this case, estimates of μ and μ_i will be obtained using estimates of c and ϕ and the shrinkage estimates of c_i and ϕ_i, respectively (see Equation 2.4).

The second limitation of standard ML software is that most multilevel software packages do not allow for individual differences in the residual variance at level 1, and if they do allow for individual differences, these differences would need to be fully accounted for by a level 2 predictor. That is, while it may be possible to model some individual differences, randomness of the innovation variance is not included as an option. As argued in the previous section, individual differences in innovation

In MLWin (Rasbash, Charlton, Browne, Healy, & Cameron, 2009), this second limitation can be circumvented by using syntax, but this implies one can no longer make use of the user friendly interface of the program.

variance are be expected to be the norm, rather than the exception, so the assumption that this variance is the same for everyone (i.e., a fixed effect), an assumptions that is implicitly made in standard software by only allowing a single error variance term, is unrealistic and undesirable: Not only does this assumption prevent us from studying individual differences in this part of the process, but since the variance of an AR(1) process is a function of both the AR-parameter and the innovation variance as shown in Equation 2.5, it may also lead to bias in the estimation of the AR-parameter. This possible source of bias is what we will investigate in the simulation study below.

2.3.2 Bayesian Estimation with WinBUGS

WinBUGS is a free software package that can be used for Bayesian estimation (Lunn et al., 2000). In contrast to standard multilevel software, the WinBUGS program allows for a lot of freedom in specifying a model. As a result, we can define the multilevel AR(1) model using the two-equation structure at level 1 (Equations 2.2 and 2.3), and relate individual differences in individual means to other individual differences at level 2 (Equation 2.6). Furthermore, it allows us to include the innovation variance as a random effect which may be related to other random effects.

We will consider two models when using WinBUGS: In the first Bayesian estimation method (*B1*), all the level 1 parameters of the multilevel AR(1) model (μ_i, ϕ_i, and σ_i^2) will be included as random effects, while in the second Bayesian estimation method (*B2*), only μ_i and ϕ_i will be random, thus implying that all individuals have the same innovation variance (i.e., $\sigma_i^2 = \sigma^2$). This latter model is included in the simulation study because it is the Bayesian equivalent of the ML estimation methods, which also contain a single residual variance term. Therefore, by not only comparing the Bayesian methods to the ML methods, but also comparing the two Bayesian methods to each other, we can differentiate between performance differences resulting from (erroneously) modeling the innovation variance as a fixed effect, and differences resulting from the use of Bayesian versus ML analyses.

Since WinBUGS is based on Bayesian estimation of the model, several steps are required before the model can be estimated by the program. While a thorough discussion of Bayesian statistics is beyond the scope of this article (interested readers are referred to Gelman, Carlin, Stern, and Rubin (2004), Hoijtink (2009), and Hamaker and Klugkist (2011)), there is one feature of Bayesian analysis that needs to be discussed here briefly – the prior distribution. In Bayesian statistics, researchers need to specify prior distributions for all model parameters, where these prior distributions represent a researcher's prior beliefs or knowledge about these parameters by assigning probabilities to their different possible values. These prior distributions are then combined with the distribution of the data using Bayes theorem in the following way,

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{f(y)}, \tag{2.8}$$

where $f(\theta|y)$ is the posterior distribution of a parameter that represents the combined information from both the prior and the data about this parameter, $f(y|\theta)$ is the dis-

tribution of the data (y) conditional on parameter θ, $f(\theta)$ is the prior distribution for parameter θ, and $f(y)$ is the distribution of the data. Posterior distributions of parameters of interest are subsequently used for model estimation. That is, the mean, median, or mode of a posterior distribution can be used as the point estimate of a parameter, while the standard deviation of the posterior distribution can be seen as a measure of the sample variability of this estimate (analogues to the standard error in standard maximum likelihood estimation). If one has little or no prior knowledge, *uninformative* priors can be used, which are characterized by assigning low and (approximately) equal probabilities to a very large range of possible values of a parameter. The results obtained with such priors depend almost exclusively on the data, and are therefore often close to ML estimates.

Specifically, for estimation method *B1*, we need to specify priors for all 9 parameters that are defined in Equation 2.6, that is, for the three fixed effect (i.e., μ, ϕ, and σ^2), for the three random effects (i.e., τ^2_μ, τ^2_ϕ, and $\tau^2_{\sigma^2}$), and the three covariances between these random effects (i.e., $\tau_{\mu\phi}$, $\tau_{\mu\sigma^2}$, and $\tau_{\phi\sigma^2}$). For the fixed effects, normal distributions with 0 means and variances equal to 10,000 were chosen as priors. That is,

$$\mu \sim \mathcal{N}(0, 10,000) \tag{2.9}$$

$$\phi \sim \mathcal{N}(0, 10,000) \tag{2.10}$$

$$\sigma^2 \sim \mathcal{N}(0, 10,000) \tag{2.11}$$

These large variances spread the normal distribution out over a very large range of values, making the distribution uninformative. In addition, we use an Inverse Wishart (IW) distribution as the prior for the covariance matrix in Equation 2.6 containing the random effects and their covariances, so that,

$$\begin{bmatrix} \tau^2_\mu & & \\ \tau_{\mu\phi} & \tau^2_\phi & \\ \tau_{\mu\sigma^2} & \tau_{\phi\sigma^2} & \tau^2_{\sigma^2} \end{bmatrix} \sim IW(R, df), \tag{2.12}$$

where R is a scale matrix that positions the distribution in multivariate space, and df are the degrees of freedom of the distribution which determine how informative it is. Usually, an identity matrix is used for scale matrix R, but given the fact that the variance of the AR-parameter is expected to be much smaller than 1 (since most individuals will have a AR-parameter between say 0 and .5), this may not be an appropriate prior in this case. In fact, preliminary analyses showed that the identity matrix is not appropriate here. Therefore, we will use ML estimates of the random effects τ^2_μ and τ^2_ϕ in the scale matrix of the IW prior, setting the off-diagonal elements to 0. Since there is no ML estimate of the random effect $\tau^2_{\sigma^2}$, we set the value for

Note that this approach of using the ML estimates in the IW prior is equivalent to the training sample approach suggested by O'Hagan (1995), where part of the data is used to obtain a prior for the analysis of the rest of the data.

this parameter to 1, as would have been the case had an identity matrix been used. Finally, to make the IW an uninformative prior, the degrees of freedom need to be set to the number of random effects in the model, which in this case is 3.

For estimation method *B2*, the same priors are used, with the exception that the IW distribution is now only for the random effects τ_μ^2 and τ_ϕ^2, and their covariance $\tau_{\mu\phi}$, thus the degrees of freedom are set to 2.

2.4 Simulation study

The purpose of the simulation study is twofold. On the one hand, we wanted to compare the performance of the different estimation methods in order to determine the effect of modeling a random innovation variance as a fixed effect. On the other hand, we were also interested in identifying the minimum number of individuals and/or time points required by the different methods for accurate parameter estimation. To this end, we varied: a) the number of individuals in the sample N; b) the number of time points T; and c) the correlation between the innovation variance and the AR-parameter.

2.4.1 Data generation and model estimation

Specifically, we generated data with 20, 50, and 100 individuals, and with 10, 20, and 50 time points. These samples sizes were considered realistic for these kinds of models based on the literature.

For the fixed and random effects parameters we used

$$\begin{bmatrix} \mu_i \\ \phi_i \\ \sigma_i^2 \end{bmatrix} \sim MVN \left(\begin{bmatrix} 10 \\ .2 \\ 3 \end{bmatrix}, \begin{bmatrix} \tau_\mu^2 & & \\ 0 & .01 & \\ 0 & \tau_{\phi\sigma^2} & 1 \end{bmatrix} \right). \tag{2.13}$$

Note that a key focus in this simulation is on whether ignoring randomness in the innovation variance results in bias in the estimation of ϕ_i. Because the relationship between ϕ_i and σ_i^2 may have an effect on this bias, the correlation between these random effects $\rho_{\phi\sigma^2}$ was varied from -.6, 0, to .6. The mean and variance for the autoregressive parameter were chosen based on Wang et al. (2012). The mean of the innovation variance is arbitrary in the sense that rescaling the observed variable y_{it} will also result in rescaling of the innovations, and thus in its variance. However, this is not independent of the variance of the innovation variance: We chose the variance of the innovation variance such that individual negative variances are unlikely to occur. Finally, the fixed effect for the mean is also an arbitrary value (here set at 10), and the variance of the mean proved arbitrary in preliminary simulations. In the current

This is relevant in the creation of the datasets as negative variances will make it impossible to generate data. However, it has no further practical consequences.

simulations, the variance of μ_i was chosen such that the total between-person variance was the same as the average within-person variance.

Datasets were created by randomly drawing N cases from the distribution defined in Equation 2.13, and subsequently generating a time series of T observations for each of the N cases using the AR(1) model defined in Equations 2.2 and 2.3.. In each scenario, 1,000 replications were created. The three ML methods were run in R, using Restricted Maximum Likelihood (REML) estimation in the `lme4` package (Bates, Maechler, & Bolker, 2011). The two Bayesian estimation methods were run by calling WinBUGS from R using the `R2WinBUGS` package (Sturtz, Ligges, & Gelman, 2005). Based on preliminary convergence checks, the number of iterations for the Bayesian estimation procedures was set to 10,000 with a burn-in of 5,000.

2.4.2 Evaluating performance

We evaluate performance with respect to the fixed effects (i.e., μ, ϕ, and σ^2), the random effects (i.e., τ^2_μ, τ^2_ϕ, and $\tau^2_{\sigma^2}$) and their correlations (i.e., $\rho_{\mu\phi} = \tau_{\mu\phi}/(\tau_\mu\tau_\phi)$, $\rho_{\mu\sigma^2} = \tau_{\mu\sigma^2}/(\tau_\mu\tau_{\sigma^2})$, and $\rho_{\phi\sigma^2} = \tau_{\phi\sigma^2}/(\tau_\mu\tau_{\sigma^2})$), and the individual parameter estimates (i.e., μ_i's, ϕ_i's and σ^2_i's). Performance with respect to the fixed effects is based on evaluating: a) the bias, which is determined by taking the difference between the true parameter value and the average parameter estimate; and b) the coverage rate of the 95% confidence or credibility interval (CI), which is determined by computing the proportion of replications for which the true parameter value lies inside the associated interval. These coverage rates should be about .95; coverage rates lower than .90 will be considered to be too low, while coverage rates over .99 will be considered to be too high. In addition, performance with respect to the random effects and their correlations is assessed on the basis of bias, while performance with respect to the individual parameter estimates is based on: a) the coverage rates of individual 95%

We do not have an analytical expression of the within-person variance. Instead, we simulated a dataset consisting of 10,000,000 persons, given a particular set of parameter values, and determined the average within-person variance. This way we determined that when $\phi = .20$, $\tau^2_\phi = .01$, $\sigma^2 = 3.00$, and $\tau_{\sigma^2} = 1.00$, the total within-person variation is equal to 3.198, 3.170, and 3.141 for $\rho_{\phi\sigma^2}$ values of .6, 0, and -.6 respectively.

During data generation it was evaluated whether all parameter values fell into a permissible range, that is, no individual innovation variances smaller than 0 and no values of ϕ_i greater than | 1 | to ensure stationarity. If parameter values fell outside these ranges, those data were discarded and a new dataset was generated; however this was rarely needed (i.e., for less than 4-5% of generated datasets).

A problem we encountered with the use of the ML estimates in the IW prior is that when the ML estimates are very inaccurate, the WinBUGS analysis crashes. This occurred in one out of every 300 to 500 datasets. In practice, if this problem occurs, the user should change the scale values; in the current simulation study, we solved this by preventing the τ^2_ϕ estimate in the scale matrix (R) of the IW prior to become too small (by substituting the value .005 for the ML estimate of τ^2_ϕ if this estimate is smaller than this boundary value) and by producing a new dataset in case WinBUGS crashed.

CIs; and b) the average correlation between the true individual parameters and the estimated individual parameters.

2.4.3 Results

Fixed effects

The results reported on the left side of Table 2.1 show that while the five methods perform similarly with respect to bias in μ, the two ML methods based on person-centering the lagged predictor (i.e., methods *MLpc1* and *MLpc2*) lead to considerable bias in the estimation of ϕ and σ^2. This is especially true when T becomes smaller, while the effect of sample size at the between-person level (i.e., N) has a negligible effect. This issue has been studied in more detail by Hamaker and Grasman (2015), and we will therefore refrain from pursuing this issue here. Instead, we conclude that the bias for the ϕ-parameter obtained with methods *MLpc1* and *MLpc2* disqualifies the two centering procedures as a proper approach to multilevel AR(1) modeling.

When comparing the remaining three methods, it can be seen that in general, the bias reported for the uncentered ML method (i.e., *MLuc*) and its Bayesian equivalent in which the innovation variance is also modeled as a fixed effect (i.e., *B2*), is of a similar size, unless $T = 10$, when the Bayesian estimation procedure outperforms the ML method with respect to ϕ. The bias obtained with the true model (i.e., method *B1*) is much smaller in comparison to the other two methods when ϕ is considered, but larger when σ^2 is considered. Overall, the three methods seem to overestimate σ^2 regardless of the correlation between ϕ_i and σ_i^2. For ϕ, a negative correlation tends to result in a negative bias, while a positive correlation is associated with a positive bias.

On the right side of Table 2.1, the coverage rates for the 95% CIs of the fixed effects are reported. The two centered ML options resulted in extremely low coverage rates for the CI of ϕ (e.g., even .014 and .040 when $N = 100$ and $T = 10$), which could be expected given the bias in this estimate. In contrast, the coverage rates for μ obtained with *MLuc* are too high (i.e., it is equal to 1), which is the result of extremely large standard errors for this estimate. The ML approach does not provide a standard error for the level 1 residual variance (here, the innovation variance σ^2), which is the reason we could not obtain coverage rates for the innovation variance obtained with the ML methods. Finally, the coverage rates of the Bayesian methods clearly outperformed the ML results, with the *B1* method (which was based on the true model) resulting in coverage rates close to the target value of .95.

The reason for this large standard error is that it had to be computed from the standard errors of c and ϕ, since μ is not directly estimated in this approach: To this end, we used the following equation from Mood, Graybill, and Boes (1985) for the variance of a quotient $\text{var}[\frac{X}{Y}] = (\frac{\mu_X}{\mu_Y})^2(\frac{\text{var}[X]}{\mu_X^2} + \frac{\text{var}[Y]}{\mu_Y^2} - \frac{2\text{cov}[X,Y]}{\mu_X \mu_Y})$. This extra estimation step forms an additional source of uncertainty, leading to large standard errors and thus coverage rates that are always 1.

Random effects

The bias in the estimation of the random effects and their correlations are summarized in Tables 2.2 and 2.3, respectively. Table 2.2 contains the parameters estimated by all five procedures, that is, τ^2_μ, τ^2_ϕ, and the correlation $\rho_{\mu\phi}$. Table 2.3 contains the additional parameters which are only estimated with B1, that is, $\tau^2_{\sigma^2}$, $\rho_{\mu\sigma^2}$, and $\rho_{\phi\sigma^2}$. Table 2.2 shows that the ML methods performed quite similarly, and that they are quite comparable to the Bayes methods when the bias for τ^2_μ is considered. With respect to the bias for τ^2_ϕ, the Bayesian methods performed less well than the ML methods, especially when both N and T are small. The more complex (and correct) model estimated with method *B1* led to slightly more bias in the estimation of τ^2_ϕ than the incorrect model estimated with *B2*. The bias associated with the Bayesian methods is always positive.

With respect to the bias in $\rho_{\mu\phi}$, the five methods performed quite similarly when $N = 100$ and $T = 50$. When T decreases, the bias obtained with the ML methods increases. Of the Bayesian methods, method *B2* seemed least affected by changes in N and/or T, while method *B1* performed quite similarly when N is 100 or 50, but clearly performs less well when $N = 20$, especially when this is also combined with a smaller T.

Table 2.3 shows that method *B1* resulted in little bias in the estimation of $\rho_{\mu\sigma^2}$. In addition, it can be seen that the estimates of $\tau^2_{\sigma^2}$ generally show positive bias, with the amount of bias being mostly affected by N, while T has little effect. For $\rho_{\phi\sigma^2}$ there generally is a bias towards zero (i.e., a positive bias when the correlation is negative and a negative bias when the correlation is positive). For this random effect, the amount of bias is especially affected by T, while decreasing N also has a detrimental, although less stark, effect.

Individual parameters

In addition to the model parameters discussed above, we also consider the individual parameter estimates. That is, for each individual, an estimate of μ_i and ϕ_i can be obtained, and in case of method *B1*, also an estimate of σ^2_i. In the Bayesian analyses, we obtained posterior distributions of both the model parameters (as discussed above) and the individual parameters. From these posterior distributions, we can obtain point estimates as well as CIs. For the ML analyses, individual parameter estimates can be obtained after the analysis is run, using the function `ranef()` from the R-package `lme4` to obtain individual point estimates, and the function `se.ranef()` from the R-package `arm` (Gelman et al., 2011), to obtain the individual standard errors with which to construct the individual CIs.

First, we computed the correlation between the estimated and the true individual parameter values in order to see to what extent the rank order of individuals would be correct if these estimates were used. In order to save space, these results are only

briefly summarized here. The correlations obtained for the ML estimation methods were always higher than the ones obtained with the Bayesian methods. For μ_i, the minimum correlation obtained with the centered ML methods was .89 for both method MLpc1 and MLpc2, while the minimum correlation obtained with the MLuc method was equal to .87. The lowest correlations with the Bayesian methods for this estimate were .78 for method *B1* and .79 for method *B2*. In addition, when $T \geq 20$ all three the ML estimation methods show correlations between the real and estimated values of μ_i that are higher than .90, while the Bayesian methods need 50 time points for this correlations to be exceed .90.

For ϕ_i, the correlations between the estimated and the true individual values were much lower than for μ_i. The maximum correlation obtained with the three ML methods was .53 (when $N = 100$ and $T = 50$), while the minimum correlation was .06 (at $N = 20$ and $T = 10$). The maximum correlation for the Bayesian methods was .43 (for method *B1* at $N = 100$ and $T = 50$), while the minimum value was .05 (for method *B2* when $N = 100$ and $T = 10$).

Estimation method *B1* also provided correlations between the true and estimated values of σ_i^2: The maximum correlation was .74 (when $N = 100$ and $T = 50$), and the minimum correlation was .32 (when $N = 20$ and $T = 10$).

Taken together, the results show that for the ML estimation methods, the correlations between the true and estimated values of μ_i and ϕ_i increase as N and (particularly) T increases. For the Bayesian estimation methods, these correlations also increase when T increases, however the effect of an increase in N is less consistent. The correlations obtained with method *B1* tend to decrease with N, except for the estimates obtained when $\rho_{\phi\sigma^2} = 0$. If the AR-parameter and innovation variance are not correlated, the correlation between the true and estimated values of the parameters tend to increase when N goes from 50 to 20. For method *B2* this trend of increased performance at lower sample sizes is even stronger, and the correlations tend to increase as N decreases regardless of the value of $\rho_{\phi\sigma^2}$.

Second, we considered the coverage rates of the individual 95% CIs. With respect to these, the ML methods performed less well than the Bayesian methods. The two centered ML methods always led to coverage rates below .90 for both ϕ_i and μ_i. Notably, the coverage rates obtained with *MLuc* for ϕ_i were also rather low (i.e., always below .90), and quite similar to the the ones obtained with the other two ML methods. The coverage rates for μ_i obtained with *MLuc* were acceptable however (i.e., always above .90). In contrast, the Bayesian methods resulted in coverage rates of the individual CIs that were always above .90.

Taken together, these results indicate that the ML estimation methods were a little better at retaining the rank order of the individual estimates with respect to ϕ_i, while the Bayesian methods were better for making individual inferences for all individual parameters.

Tables containing these correlations and the individual coverage rates can be obtained from the first author.

2.4.4 Conclusion

The first aim of the simulation study was to determine whether ignoring the randomness in innovation variance leads to bias in the estimation of the other parameters, particularly the AR-parameter. Because the observed variance is a function of both the innovation variance and the AR-parameter, it could be expected that ignoring randomness in the former, leads to problems concerning the latter. The second aim was to determine the trade off between T and N.

We compared five estimation methods, of which only one included the innovation variance as a random effect. When comparing the results from these five methods, we can conclude the following. First, person-mean centering the predictor leads to considerable bias in the estimation of ϕ, and should therefore not be used (see for a more elaborate discussion, Hamaker & Grasman, 2015). Second, when comparing the results obtained from methods *MLuc* and *B2* to the results obtained with estimating the true model (method *B1*), we can conclude that ignoring the randomness in innovation variance leads to bias in the estimation of ϕ. The direction of this bias depends on the actual correlation between ϕ_i and σ_i^2: When there is a positive correlation, ϕ tends to be overestimated, and when there is a negative correlation, ϕ tends to be underestimated. These results are also reflected by coverage rates that regularly drop below .90. Third, including the innovation variance as a random effect (i.e., method *B1*) is not associated with a specific pattern of bias, and in comparison to the other methods, the estimated bias for ϕ is very small, although the random effects are generally overestimated. Furthermore, the coverage rates obtained with method *B1* are always above .90, and are often close to the target value .95. This is also true for the coverage rates of the individual CIs. The only criterion on which the ML methods outperformed method *B1*, was with respect to the correlation between the true and estimated individual ϕ_is.

Focusing on the effect of sample size on the results obtained with method *B1*, we can conclude the following. For the fixed effects, the bias increases when either N or T decreases, while the bias for the random effects, which is always positive, seems more strongly affected by N than by T. Still, the coverage rates were always above .90, indicating that this approach can be effectively used for making inferences even with small sample sizes such as $N = 20$ and $T = 10$.

2.5 Empirical application

To further illustrate the Bayesian estimation of a multilevel AR(1) model, we apply estimation method B1 to data collected in a study by Laurenceau et al. (2005). In this study, spouses from 96 married couples independently completed a structured diary each evening over a period of 42 consecutive days. Based on the partial overlap in the affective items in this dataset and the items of the PANAS-X (Watson & Clark, 1999), we selected three items (i.e., excited, enthusiastic, and energetic rated on 5-point Likert scales), to comprise a single positive affect (PA) score. Focusing on the

women only, there were 127 out of the total of 4032 (= 96*42) PA scores missing. Based on individual sequence plots (i.e., plots of the repeated measurements of each woman), we removed seven women who had none or very little variability over time, such that the final dataset contained 89 female participants.

To analyze the data using method B1, we began by analyzing the data using standard ML analysis with person centering based on observed mean scores and listwise deletion to get estimates for τ_μ^2 and τ_ϕ^2, which are needed for the scale matrix of the IW prior. Next, the scores of the 89 females were analyzed using estimation method B1. To evaluate whether the analysis converged, we ran three separate MCMC chains with different starting values and considered the mixing of the trace plots and the values of the Gelman-Rubin statistic for each parameter. Starting values for the fixed effects and the covariance matrix were based on random draws from a standard normal distribution (chain 1 and 2), or based on the ML analysis (chain 3). We used a burn-in of 5,000 iterations and total number of 10,000 iterations. Following initial convergence checks, we decided to use a thinning rate of 10. As a result, we ran the analysis for a total of 100,000 iterations (10,000 * 10). With these settings, the analysis of the trace plots and the Gelman-Rubin statistic indicated convergence.

The results obtained with method B1 are summarized in Table 4 and Figure 2.2. The first column contains the point estimates (i.e., the means of the posterior distribution) and the standard deviation between parentheses (i.e., the standard deviation of the posterior distribution), while the second column contains the lower and upper bounds of the 95% CIs. Since the 95% CI for ϕ lies above zero, we can conclude that–on average–the women are characterized by a carryover of yesterday's PA on today's PA.

The point estimate of the variance of the mean (i.e., τ_μ^2) is equal to 4.442 (95% CI ranges from 3.260 to 6.026) indicating there is considerable variation in the average PA of individuals over time. The point estimate of the variance of the inertia (i.e., τ_ϕ^2) is .008 (95% CI ranges from .002 to .018). While this may seem like a small variance, it should be noted that the ϕ parameter itself is likely to be small as for stationary processes it must lie in the range of -1 to 1; in practice it will be much more often between 0 and .5 or so (cf., Wang et al., 2012). The point estimate of the variance of the innovation variance (i.e., $\tau_{\sigma^2}^2$) is 4.389 (95% CI ranging from 2.948 to 6.391), suggesting there is considerable between-person variation in this source of variance. This corresponds to the idea that individuals differ in their sensitivity, reactivity, and exposure to external events that influence the process under investigation. Here, it seems to imply that individuals differ in the amount and/or severity of positive and negative events that they encounter in daily life, as well as their sensitivity and reactivity to such events. Note that the 95% CIs of the variances cannot include zero because we are using an IW prior, such that we cannot use the CIs as an informal test of whether the parameter should be considered to differ from zero. However, since the lower bounds are (relatively) far away from zero, we believe it is safe to conclude that all three parameters are characterized by a meaningful level of individual differences.

Finally, when considering the covariances between the random effects, each of the CIs includes zero, such that we cannot conclude that these parameters are truly dif-

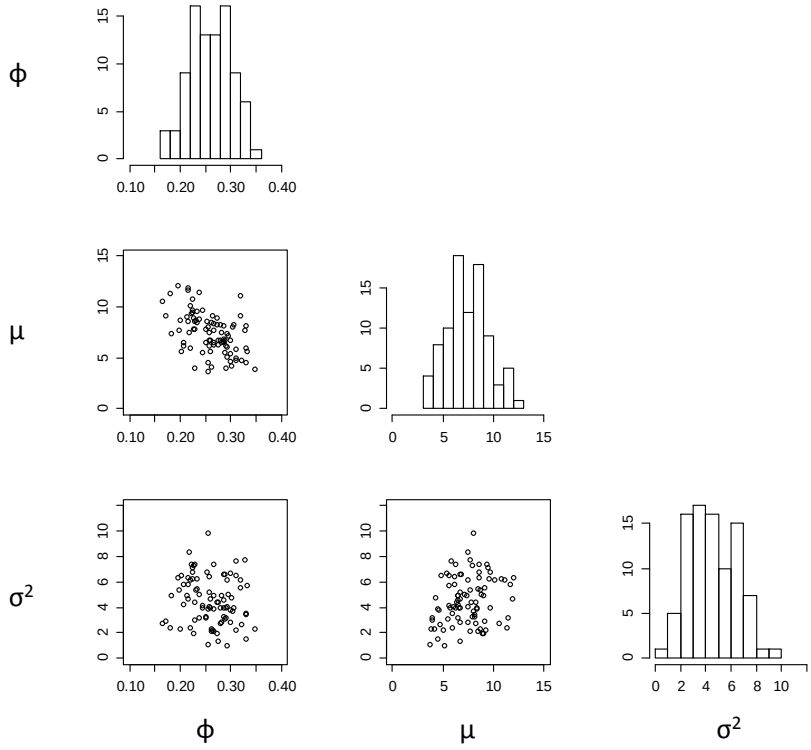


Fig. 2.2: The histograms show the estimated posterior distributions of the three random parameters ϕ, μ, and σ^2. The scatterplots show the bivariate relation between the random variables of the corresponding row and column (e.g, the scatterplot on the second row of the first column shows the relation between μ and ϕ).

ferent from zero. This would imply that the unobserved factors that influence the individuals' means, their inertias, and their innovation variances do not overlap. For example, if the trait extraversion were to have a positive effect on the average PA level of individuals (and thus be predictive of μ_i), it is unlikely to affect the individuals' inertia or their exposure and/or reactivity to time-varying factors that influence PA (i.e., ϕ_i and σ_i^2). Although we have not considered the CIs for correlations or covariances in the current study in detail, preliminary results suggested to us that these tend to be too wide, such that they may not be that appropriate for the current purpose. Thus, at this stage it is too early to conclude that the individual means, inertias and innovation variances are affected by the same factors, even though we have found no evidence for relatedness between these random effects.

2.6 Discussion

In this paper, we presented a multilevel extension of the AR(1) model and compared several ways to estimate it. The model we considered here is more extensive than typically considered in the literature as it includes a random (rather than fixed) innovation variance as well as a random autoregressive parameter. We argued that there are both substantive and statistical reasons for preferring this extended multilevel AR(1) model. First, between-person differences in innovation variances may form an important source of information. The innovation can be conceptualized as a collection of all unobserved temporal factors that influence the process under investigation, both internally (e.g., hormonal levels, alcohol intake, cognitions, associations, appraisal of events) and externally (e.g., social obligations, personal interactions, weather, political developments). Allowing for individual differences in the innovation variance implies we allow not only for individual differences in sensitivity and/or responsiveness to these factors, but also for individual differences in exposure to these factors (or, more specifically, individual differences in the variability of these factors).

Second, using a simulation study, we showed that when the innovation variance is in fact random, ignoring this source of individual differences leads to bias in the estimation of the AR-parameter (where the direction of the bias depends on the correlation between the innovation variance and the AR-parameter). This can be explained by the fact that the variance of an AR(1) process is a function of both the innovation variance and the AR-parameter, and when one of these is fixed across individuals, the other is the only random source that can account for individual differences in observed variance. The impact, or cost, of this bias in the AR-parameter depends on the amount of autocorrelation. Our simulation study showed that the maximum bias is likely around -.12, so if the true value of the AR parameter is far away from 0, the consequences of the bias are probably not that severe. If the true value is close to 0 however, the bias could change the estimate from positive to negative. The latter possibility is a more severe problem, as a negative AR parameter describes a qualitatively different process than a positive one.

Based on these arguments, we advise researchers interested in applying a multilevel AR(1) model to use an approach that allows for the inclusion of the innovation variance as a random effect. This can be done in WinBUGS, which has the additional advantage that it allows for defining the AR(1) process in two equations, such that we can estimate the individual mean rather than the intercept. The mean has a more meaningful interpretation in terms of an individual's long-term tendency (i.e., it is the score a person would turn to if there would be no more random input to the process), whereas the intercept is generally less meaningful (i.e., the expected score when the score at the preceding occasion is equal to zero). Furthermore, while person-mean centering the AR predictor implies that the intercept at level 1 becomes the individual's mean on the dependent variable, it should be discouraged as it leads to a negative bias in the estimation of the AR-parameter (as has been shown in the simulation study).

The results from our simulation study also indicated that it was difficult to retain the rank order in the individual innovation variances and the individual AR-parameters (especially if fewer than 50 time points are available). Future research should focus on how well between-person differences in innovation variances and AR-parameters can be predicted at the second level using person characteristics. This is particularly important because regressing the innovation variance at the second level on measurements of, for instance, sensation seeking behavior, neuroticism, and/or sensitivity (e.g., sensory-processing sensitivity, Aron & Aron, 1997) should help determine what factors play a role in the (individual differences in) variability of a particular process. Such an approach could also be used as a first step in determining which factors should be considered as candidates to be included as level 1 predictors in subsequent studies in order to model their effects on the process more explicitly.

Also, it should be noted that we focused on one particular form of heterogeneity in this paper: inter-individual variability. However, researchers might also be interested in other forms of variability, like variability in parameters across time, or (qualitative) differences in the kind of process that best describes the repeated measurements of individuals. As an example of the first of these other types of variability, one could think of a situation in which the process under investigation depends on the current state of an individual, with different states leading to different parameter values (e.g., different amounts of inertia). Data from this type of scenario can be analyzed with Treshold Autoregressive (TAR) models (De Haan-Rietdijk, Gottman, Bergeman, & Hamaker, 2014), in which additional variability in model parameters is possible through regime switching. An example of the second alternative type of variability would be a sample in which the repeated measures of some individuals can be characterized by an AR(1) process, while others may be better described by an AR(2). In this case, researchers might choose to run separate analyses for each individual in the sample, or use a mixed model approach in which different (level 1) processes are allowed for different individuals within the larger multilevel model.

Note that these alternative types of variability can be combined with the form of inter-individual variability examined in this paper. The TAR model could be extended by allowing different amounts of innovation variance for different individuals for example, with the amount of innovation variance of each individual also varying across time. This could be a important extension since the reason that erroneously modeling the innovation variance as fixed leads to bias in the AR-parameter likely applies to this type of model as well. Similarly, random model parameters (i.e., the innovation variance) can also be incorporated into mixed models that allow different types of processes for different individuals. This can be done by specifying random innovation variances for every individual in the sample to prevent bias in the parameter estimates of these mixed models, or by specifying random innovation variances for a subset of the sample to distinguish between individuals for who all factors of interest are explicitly modeled, and those for who some factors are still unknown or unmeasured.

Table 2.1: Bias and Coverage Rates of Fixed Effect Estimates

		Bias									Coverage Rates								
		μ			ϕ			σ^2			μ			ϕ			σ^2		
	$\rho_{\phi\sigma^2}$	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6
N=100	MLpc1	.010	.003	.012	-.011	-.025	-.039	.001	-.002	.000	.961	.954	.946	.904	**.690**	**.397**			
T=50	MLpc2	.010	.003	.012	-.010	-.025	-.038	.001	-.002	.000	.959	.951	.946	.910	**.702**	**.407**			
	MLuc	.011	.003	.012	.016	.001	-.013	.007	.003	.003	**1.000**	**1.000**	**1.000**	**.827**	.932	**.888**			
	B1	.013	.005	.014	.001	.000	-.001	.008	.006	.006	.966	.961	.948	.955	.950	.955	.943	.945	.945
	B2	.011	.003	.013	.016	.001	-.013	.007	.003	.003	.967	.963	.952	**.864**	.944	**.898**	**.704**	**.680**	**.723**
T=20	MLpc1	.004	-.006	.006	-.052	-.063	-.081	-.018	-.017	-.007	.950	.932	.946	**.477**	**.295**	**.117**			
	MLpc2	.004	-.006	.006	-.048	-.059	-.078	-.017	-.020	-.007	.949	.930	.943	**.534**	**.341**	**.149**			
	MLuc	.004	-.008	.006	.020	.007	-.012	.003	.000	.005	**1.000**	**1.000**	**1.000**	**.829**	.905	**.877**			
	B1	.008	-.003	.010	.002	.002	-.004	.009	.006	.016	.954	.942	.950	.957	.945	.932	.955	.950	.941
	B2	.005	-.005	.008	.017	.005	-.014	.004	.000	.008	.954	.940	.949	.903	.941	.900	**.821**	**.824**	**.812**
T=10	MLpc1	.005	-.003	.005	-.126	-.139	-.154	-.074	-.065	-.055	.939	.950	.953	**.070**	**.039**	**.014**			
	MLpc2	.005	-.004	.005	-.112	-.124	-.139	-.072	-.064	-.055	.937	.949	.954	**.139**	**.076**	**.040**			
	MLuc	.004	-.003	.007	.041	.026	.009	.009	.010	.010	**1.000**	**1.000**	**1.000**	**.696**	**.810**	**.860**			
	B1	.010	.002	.009	.007	.002	-.005	.012	.021	.021	.952	.957	.960	.936	.950	.940	.945	.947	.954
	B2	.008	.000	.007	.020	.006	-.012	-.007	.000	.000	.952	.955	.959	**.889**	.929	.923	**.870**	**.887**	**.886**
N=50	MLpc1	.007	.004	.007	-.011	-.025	-.038	-.002	.008	.003	.951	.957	.941	.923	**.847**	**.662**			
T=50	MLpc2	.007	.004	.007	-.011	-.024	-.038	-.002	.008	.006	.950	.956	.937	.926	**.853**	**.674**			
	MLuc	.007	.005	.006	.016	.001	-.013	.003	.012	.006	**1.000**	**1.000**	**1.000**	**.875**	.937	.900			
	B1	.008	.005	.008	.001	-.001	-.002	.011	.022	.016	.965	.963	.959	.957	.954	.954	.933	.935	.931
	B2	.008	.005	.008	.015	.001	-.013	.005	.014	.008	.959	.962	.954	.912	.955	.923	**.691**	**.657**	**.660**
T=20	MLpc1	.000	.010	.007	-.052	-.066	-.083	-.013	-.009	-.010	.954	.953	.946	**.692**	**.558**	**.387**			
	MLpc2	.000	.010	.007	-.048	-.062	-.079	-.013	-.008	-.010	.952	.949	.944	**.723**	**.599**	**.421**			
	MLuc	.001	.011	.004	.019	.004	-.014	.006	.008	.003	**1.000**	**1.000**	**1.000**	**.859**	.908	**.898**			
	B1	.002	.010	.008	.000	-.003	-.010	.028	.030	.027	.961	.961	.955	.949	.944	.958	.936	.945	.948
	B2	.001	.010	.007	.017	.002	-.016	.011	.010	.007	.957	.955	.950	.910	.936	.934	**.816**	**.822**	**.814**
T=10	MLpc1	.002	.011	.016	-.130	-.143	-.156	-.070	-.056	-.060	.942	.952	.936	**.315**	**.214**	**.155**			
	MLpc2	.003	.011	.016	-.115	-.128	-.141	-.068	-.055	-.059	.940	.947	.930	**.415**	**.300**	**.240**			
	MLuc	.005	.011	.018	.036	.020	.006	-.068	.020	.008	**1.000**	**1.000**	**1.000**	**.796**	**.848**	**.863**			
	B1	.006	.011	.017	-.001	-.009	-.015	.045	.056	.048	.957	.959	.952	.954	.946	.938	.952	.943	.936
	B2	.004	.010	.015	.018	.003	-.011	.003	.013	.005	.956	.956	.948	.913	.937	.932	**.893**	**.889**	**.862**
N=20	MLpc1	.015	.015	-.015	-.012	-.025	-.040	.004	.002	.005	.942	.919	.935	.928	.902	**.804**			
T=50	MLpc2	.015	.015	-.015	-.011	-.024	-.040	.004	.002	.005	.942	.919	.935	.928	.902	**.806**			
	MLuc	.015	.017	-.014	.015	.001	-.016	.009	.006	.008	**1.000**	**1.000**	**1.000**	**.893**	.935	.913			

Continued on next page

Table 2.1 – continued from previous page

		Bias									Coverage Rates								
		μ			ϕ			σ^2			μ			ϕ			σ^2		
	$\rho_{\phi\sigma^2}$	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6	.6	0	−.6
	B1	.013	.014	-.016	.005	.003	-.002	.036	.035	.038	.965	.942	.951	.965	.965	.972	.947	.951	.956
	B2	.017	.017	-.013	.012	-.001	-.016	.014	.010	.013	.963	.938	.950	.947	.958	.938	**.669**	**.698**	**.700**
T=20	MLpc1	.007	-.001	.012	-.053	-.067	-.083	-.029	-.039	-.023	.938	.921	.942	**.856**	**.795**	**.718**			
	MLpc2	.007	-.001	.012	-.050	-.063	-.079	-.029	-.039	-.023	.937	.919	.938	**.869**	**.816**	**.735**			
	MLuc	.008	.000	.010	.016	.001	-.016	-.010	-.024	-.009	**1.000**	**1.000**	**1.000**	.906	.918	.924			
	B1	.006	-.001	.012	.011	.006	-.002	.062	.047	.064	.957	.949	.962	.976	.955	.965	.963	.953	.962
	B2	.010	.002	.015	.009	-.004	-.020	.002	-.012	.004	.953	.944	.960	.965	.946	.947	**.832**	**.832**	**.816**
T=10	MLpc1	-.005	.017	.014	-.130	-.148	-.157	-.068	-.075	-.054	.943	.935	.924	**.677**	**.585**	**.557**			
	MLpc2	-.005	.017	.013	-.114	-.132	-.141	-.067	-.074	-.053	.938	.934	.925	**.744**	**.652**	**.622**			
	MLuc	.000	.020	.004	.035	.012	.002	.017	-.001	.016	**.994**	**.994**	**.995**	**.837**	**.865**	**.886**			
	B1	-.004	.015	.016	.022	.009	.005	.145	.128	.151	.965	.965	.958	.963	.955	.970	.955	.961	.961
	B2	.000	.019	.020	.008	-.013	-.022	.022	.006	.026	.955	.955	.954	.952	.943	.962	**.894**	**.879**	**.881**

Bias and coverage rates of the fixed effects estimates of the five estimation methods. Where μ is the mean parameter, ϕ is the AR-parameter, σ^2_ϵ is the innovation variance, and $\rho_{\phi\sigma^2_\epsilon}$ is the correlation between the AR-parameter and the innovation variance. The real values for μ, ϕ, and σ^2_ϵ were 10, .2 and 3 respectively. The coverage rates are for the 95% confidence (ML methods) and credibility intervals (Bayesian methods). Values lower than .90 and values of .99 or higher are printed in bold.

Table 2.2: Bias in Variance and Correlation Estimates for Mean and AR-parameter

		τ^2_μ			τ^2_ϕ			$\rho_{\mu\phi}$		
	$\rho_{\phi\sigma^2}$	.6	0	−.6	.6	0	−.6	.6	0	−.6
N=100	MLpc1	.044	.045	.029	-.001	.000	.000	-.008	-.003	-.004
T=50	MLpc2	.020	.023	.009	-.001	.000	.000	-.008	-.003	-.000
	MLuc	-.048	-.043	-.054	-.002	-.001	.000	-.008	-.004	-.000
	B1	.096	.100	.086	.001	.000	.002	-.006	.001	.003
	B2	.073	.075	.061	-.001	.001	.001	-.003	.002	.001
T=20	MLpc1	.098	.084	.074	.000	.000	.000	.005	.007	.010
	MLpc2	.050	.040	.037	.000	.000	.000	.004	.006	.010
	MLuc	-.079	-.100	-.093	-.001	.000	.002	.004	.006	.011
	B1	.087	.080	.079	.004	.003	.005	.004	.009	.002
	B2	.063	.054	.055	.001	.003	.003	.005	.008	.001
T=10	MLpc1	.211	.191	.175	.000	.000	-.001	.014	.005	-.010
	MLpc2	.156	.144	.138	.000	.000	-.001	.016	-.004	-.003
	MLuc	-.154	-.170	-.174	.002	.005	.007	.012	.006	-.012
	B1	.084	.080	.087	.008	.009	.009	.001	.007	.001
	B2	.058	.055	.066	.005	.007	.007	.003	.009	-.003
N=50	MLpc1	.024	.039	.052	.000	.000	.000	-.002	-.010	.016
T=50	MLpc2	.000	.017	.032	.000	.000	.000	-.002	-.012	.016
	MLuc	-.070	-.047	-.025	-.001	-.001	.000	-.001	-.010	.016
	B1	.181	.200	.215	.003	.001	.003	.004	-.004	.007
	B2	.116	.134	.149	.000	.001	.001	.001	-.007	.007
T=20	MLpc1	.124	.111	.101	.000	.000	.000	.001	-.003	.012
	MLpc2	.076	.068	.064	.000	.000	.000	.004	.001	.012
	MLuc	-.060	-.059	-.060	.000	.000	.001	.001	-.003	.012
	B1	.226	.217	.216	.006	.006	.003	.006	.007	.017
	B2	.155	.148	.150	.003	.004	.005	-.002	-.002	.013
T=10	MLpc1	.220	.188	.174	.002	.002	.002	-.029	.026	.027
	MLpc2	.166	.143	.137	.001	.002	.002	-.028	.020	.026
	MLuc	.166	-.183	-.183	.001	.006	.008	-.028	.023	.028
	B1	.228	.201	.206	.011	.012	.012	.003	.006	.006
	B2	.143	.120	.124	.009	.010	.011	-.008	-.005	-.006
N=20	MLpc1	-.016	.025	.067	.000	.000	.000	-.007	-.017	-.010
T=50	MLpc2	-.040	.003	.047	.000	.000	.000	-.006	-.018	-.010
	MLuc	-.102	-.059	-.017	-.001	.000	.000	-.007	-.017	-.010
	B1	.423	.469	.515	.007	.006	.006	-.011	-.016	-.015
	B2	.308	.355	.402	.004	.004	.004	.007	-.004	-.001
T=20	MLpc1	.110	.123	.085	.002	.002	.003	.016	-.000	.001
	MLpc2	.063	.081	.048	.002	.002	.003	.013	-.002	-.000
	MLuc	-.079	-.051	-.086	.001	.002	.004	.015	-.000	-.000
	B1	.515	.536	.489	.013	.012	.015	-.018	-.017	-.011
	B2	.393	.414	.373	.010	.011	.012	.005	.003	.007
T=10	MLpc1	.204	.190	.165	.007	.006	.006	.005	-.002	.056
	MLpc2	.149	.146	.128	.006	.006	.006	-.014	.013	.059
	MLuc	-.248	-.227	-.222	.008	.010	.011	.006	-.003	.056
	B1	.534	.519	.489	.023	.022	.022	-.022	-.017	-.015
	B2	.402	.373	.367	.019	.020	.020	.003	.003	.008

Bias in the variance and correlation estimates of the five estimation methods for μ and ϕ. Where τ^2_μ is the variance of the mean parameter, τ^2_ϕ is the variance of the AR-parameter, $\rho_{\mu\phi}$ is the correlation between the mean parameter and the AR-parameter, and $\rho_{\phi\sigma^2}$ is the correlation between the AR-parameter and the innovation variance. The real values for τ^2_μ are 2.188, 2.160, and 2.131 for $\rho_{\phi\sigma^2}$ values of .6, 0, and -.6 respectively. The real values for τ^2_ϕ and $\rho_{\mu\phi}$ were always equal to .01 and 0 respectively.

Table 2.3: Bias in Variance and Correlation Estimates for the Innovation Variance

		$\tau^2_{\sigma^2}$			$\rho_{\mu\sigma^2}$			$\rho_{\phi\sigma^2}$		
	$\rho_{\phi\sigma^2}$	.6	0	−.6	.6	0	−.6	.6	0	−.6
N=100	T=50	.037	.026	.038	.004	.005	.004	-.124	.004	.127
	T=20	.021	.025	.039	.007	.002	.003	-.286	.018	.300
	T=10	.005	.003	.015	.004	.001	-.002	-.427	.030	.456
N=50	T=50	.078	.061	.088	.001	-.011	.004	-.210	.008	.214
	T=20	.076	.072	.093	.001	.007	.002	-.377	.004	.392
	T=10	.074	.088	.108	-.003	.005	.000	-.492	.028	.530
N=20	T=50	.286	.270	.289	.005	-.000	-.007	-.335	-.007	.355
	T=20	.313	.313	.313	-.002	.002	-.006	-.466	.021	.502
	T=10	.146	.477	.495	-.004	-.006	.002	-.514	.029	.581

Bias in the variance and correlation estimates of estimation methods B1 for σ^2. Where $\tau^2_{\sigma^2}$ is the variance of the innovation variance, $\rho_{\mu\sigma^2}$ is the correlation between the mean parameter and the innovation variance, and $\rho_{\phi\sigma^2}$ is the correlation between the AR-parameter and the innovation variance. The real value of $\rho_{\phi\sigma^2}$ differs between scenarios as indicated in the Table. The real values for τ_{σ^2} and $\rho_{\mu\sigma^2}$ were always equal to 1 and 0 respectively.

Table 2.4: Results from the empirical application of the multilevel AR(1) model to data from Laurenceau et al. (2005)

	Parameter Estimate	95% Credibility Interval
Fixed Effects		
μ	7.406 (.230)	6.955 - 7.857
ϕ	.260 (.020)	.220 - .300
σ^2	4.533 (.249)	4.052 - 5.034
Random Effects		
τ^2_μ	4.442 (.711)	3.260 - 6.026
τ^2_ϕ	.008 (.004)	.002 - .018
$\tau^2_{\sigma^2}$	4.389 (.880)	2.948 - 6.391
$\tau_{\mu\phi}$	-.234 (.216)	-.625 - .216
$\tau_{\mu\sigma^2}$	.199 (.115)	-.036 - .417
$\tau_{\phi\sigma^2}$	-.057 (.246)	-.532 - .419

The table shows the parameter estimates and 95% Credibility Intervals of the Bayesian analysis of the daily positive affect data from Laurenceau et al. (2005). The standard deviations of the posterior distributions of the parameters are given between brackets.

3

Explained Variance and Intra-Class Correlation in a Two Level AR(1) Model

Summary. The total variance of a first order autoregressive AR(1) time series is well known in time series literature. However, despite the increased use and interest in two level AR(1) models, an equation for the total variance of these model does not exist. This paper presents an approximation of this total variance. It will be used to compute the unexplained and explained variance at each level of the model, the proportion of explained variance, and the intra-class correlation (ICC). The use of these variances and the ICC will be illustrated using an example concerning structured diary data about the positive affect of 96 married woman.

3.1 Proportion of Variance Explained in the AR(1) Model

3.1.1 The AR(1) Model

The AR(1) model was developed to model longitudinal data in which observations that are closer together in time are more highly correlated than observations that are further apart (Jöreskog, 1971, 1979). If we use this model to analyze repeated measurements of an individual that were collected on $t = 1, ..., T$ consecutive measurement occasions, then we can think of the score y_t as consisting of two parts: a mean score μ, which represents an individual's trait score (i.e., his/her long-run tendency, equilibrium, or long-term preferred state) and a error term ζ_t that represents a temporal deviation from this mean:

$$y_t = \mu + \zeta_t. \tag{3.1}$$

This chapter is submitted at Multivariate Behavioral Research.

J.Jongerling derived the expressions in this study, wrote the article and necessary computer code, and ran the analyses.

H. Hoijtink supervised the study and gave input and feedback on the study and the paper.

The temporal deviations (or states) can subsequently be modeled with the AR(1) model:

$$\zeta_t = \phi\zeta_{t-1} + \epsilon_t, \tag{3.2}$$

where ϕ is the AR-parameter used for the regression of each temporal deviation on its immediate preceding value, and ϵ_t is the unpredicted part, also referred to as the innovation, residual, or random shock. It is assumed that ϕ lies between -1 and 1 to ensure stationarity, that is, a situation in which the mean and variance of the process do not change over time (Hamilton, 1994, p. 53; Chatfield, 2003, pg. 12). Furthermore, it is assumed that the innovations are independent and normally distributed with mean 0 and variance σ^2, and that the temporal deviations ζ_t are independent and normally distributed with mean 0 and variance σ^2_ζ.

3.1.2 Explained Variance

For the AR(1) model the total variance is:

$$\sigma^2_y = \frac{\sigma^2}{1-\phi^2}, \tag{3.3}$$

where σ^2_y denotes the total variance of the time series (Hamilton, 1994, pg. 53), ϕ is the AR-parameter, and σ^2 is the variance of the innovation ϵ_t. Since the innovation is the unpredicted part of the AR(1) model, σ^2 can also be interpreted as the unexplained variance in an AR(1) model:

Definition 1 *The unexplained variance in an AR(1) model is equal to the variance of the innovation, σ^2.*

The proportion of explained variance R^2 in the AR(1) model can then be found by dividing the unexplained variance by the total variance, to get the proportion of unexplained variance, and subtracting the result from 1:

$$\begin{aligned} R^2 &= 1 - \frac{\sigma^2}{\frac{\sigma^2}{1-\phi^2}}, \\ &= 1 - \frac{\sigma^2(1-\phi^2)}{\sigma^2} \\ &= \phi^2. \end{aligned} \tag{3.4}$$

3.2 Total Variance in the Two Level AR(1) Model

3.2.1 The Two Level AR(1) model

The two level AR(1) model can be used to model time series for $i = 1, ..., N$ individuals, where N denotes the total number of individuals. In this section, we begin

with presenting the level 1 or *within-person part* of the two level AR(1) model. On this level, the scores of each of the N individuals are modeled using the AR(1) model presented in the previous section. This is followed by the presentation of the level 2 or *between-persons part* of the model, in which individual differences in the level 1 parameters are modeled.

Level 1: Within-person

Like the AR(1) model presented in the previous section, the within-person part of a two level AR(1) model can be though of as consisting of two parts; a mean score and a temporal deviation from this mean. The only difference is that, in the two level AR(1) model, all parameters in the equations for these two parts have a subscript i, to identify the individual to which the model applies:

$$y_{it} = \mu_i + \zeta_{it}, \tag{3.5}$$

where,

$$\zeta_{it} = \phi_i \zeta_{i,t-1} + \epsilon_{it}. \tag{3.6}$$

Level 2: Between-person

The parameters of the within-person part of the model, that is the mean μ_i, the inertia ϕ_i, and the innovation variance σ_i^2, may be characterized by individual differences, which we can model at level 2:

$$\begin{bmatrix} \mu_i \\ \phi_i \\ \sigma_i^2 \end{bmatrix} \sim MVN \left(\begin{bmatrix} \mu_\mu \\ \mu_\phi \\ \mu_{\sigma^2} \end{bmatrix}, \begin{bmatrix} \tau_\mu^2 & & \\ \tau_{\mu\phi} & \tau_\phi^2 & \\ \tau_{\mu\sigma^2} & \tau_{\phi\sigma^2} & \tau_{\sigma^2}^2 \end{bmatrix} \right), \tag{3.7}$$

where MVN denotes a multivariate normal distribution, τ_μ^2, τ_ϕ^2, and $\tau_{\sigma^2}^2$ are the inter-individual variances in the model parameters, $\tau_{\mu\phi}$ is the covariance between the mean and the AR-parameter, $\tau_{\mu\sigma^2}$ is the covariance between the mean and the innovation variance, and $\tau_{\phi\sigma^2}$ is the covariance between the AR-parameter and the innovation variance. Furthermore, μ_μ, μ_ϕ, and μ_{σ^2} are the expected values of μ_i, ϕ_i, and σ_i^2 respectively. More information on this specification of a multilevel AR(1) model and the reasons for allowing for individual differences in the innovation variance is given in Jongerling, Laurenceau, and Hamaker (2015).

3.2.2 Total Variance

Until now, no expression was available for the total variance of a two level AR(1) model. An approximation for the total variances across all time points t and all individuals i is derived in Appendix A:

$$\sigma_y^2 \approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2. \tag{3.8}$$

Note, that inter-individual variance in the innovation variance does not directly influence the total variance. This follows from the fact that Equation 3.8 does not contain the term $\tau_{\sigma^2}^2$. Instead, random variance in the innovation variances $\tau_{\sigma^2}^2$ only influences σ_y^2 through its covariance $\tau_{\phi\sigma^2}$ with the AR-parameter ϕ.

To test the accuracy of this approximation we undertook a small simulation study in which we compared the estimate of the total variance obtained with Equation 3.8 to the sample variance in a set of generated datasets. These data were generated with different sample sizes and different parameter values, and the sample variances of the generated data were compared to two different total variance estimates calculated with Equation 3.8: one estimate based on the parameter values used to generate the data (the population values), and one estimate based on parameter estimates obtained by analyzing the corresponding generated dataset using method B1 form the study by Jongerling, Laurenceau, and Hamaker (2015).

Specifically, we generated data with a realistic sample size of 100 individuals and with a very large sample of 10,000,000. The very large datasets were used to check the asymptotic performance of Equation 3.8. In samples of 10,000,000 individuals, the sample variance will be really close to the true population variance. In addition, the total variance estimate obtained by entering the population parameter values (i.e., the parameter values used to generate the data) into Equation 3.8 can be viewed as an estimate of the population variance, and if Equation 3.8 is correct, this total variance estimate should be really close to the sample variance in the large sample.

The sample size of 100 individuals was subsequently used to determine the performance of Equation 3.8 in more realistic situations. In these smaller samples, the sample variance is a less accurate estimate of the population variance, so comparing it to a estimate of the population variance obtained with Equation 3.8 provides less information about the accurateness of the equation than it did in the larger datasets. Therefore, instead of comparing the sample variance to a total variance estimate obtained by entering the population parameter values (i.e., the values used to generate the datasets) into Equation 3.8, we will compare the sample variance to an estimate of the total variance obtained by first estimating the parameter values using the current sample data, and subsequently using these parameter-estimates in Equation 3.8. This will give an indication of how accurate Equation 3.8 can estimate the total sample variance based on information obtained from the corresponding sample data.

For both the datasets with $N = 100$ and with $N = 10,000,000$ we used $T = 20$ time points to generate our data. Note that this is not a large number of measurements, so our approximation of the total variance will be thoroughly tested. The parameter values chosen to generate the data are given in Table 3.1. Note that all covariances not listed in this table were set to 0. These parameter values were chosen based on values found in applications of the two level AR(1) model (Wang et al., 2012). The only exception being the correlation between the AR-parameter and innovation variance $\rho_{\phi\sigma^2}$ of -.95, that was used in the last two simulations. This extreme value was chosen

to (again) thoroughly test our approximation, and the parameter values for these last two simulations where chosen so that this extreme correlation value could be obtained. The results of this simulation study are shown in the right hand panel of Table 3.1. They show that the relative difference between the sample variance of the large sample and the total variance estimate obtained by entering population values into Equation 3.8 is 1% at most (in case of the extreme correlation of $-.95$, where $(3.2029 - 3.1711)/3.2029 = .01$), and that the largest relative difference between the small sample variance and the total variance estimate obtained by entering parameter estimates into Equation 3.8 is equal to 3.15% (for $\rho_{\phi\sigma^2} = -.600$, where $(5.3173 - 5.4849)/5.3173 = -.0315$). Note that it is not surprising that the relative difference is larger when comparing the small samples variances to the total variance calculated using parameter estimates, since the fact that we use parameter estimates in this comparison, instead of the true population values, will result in an extra source of bias. Taken together, we feel these results show that Equation 3.8 provides accurate and useful estimates of the total sample variance.

Table 3.1: Results from the simulation study comparing the total variance estimated using Equation 3.8 to the sample variance of generated datasets with varying sample sizes and parameter estimates.

μ_μ/τ^2_μ	μ_ϕ/τ^2_ϕ	$\mu_{\sigma^2}/\tau^2_{\sigma^2}$	$\rho_{\phi\sigma^2}$	N	Sample Variance	Eq. 3.8 pop.	Eq. 3.8 samp.
10/2.188	.20/.010	3.00/1.00	.600	100	5.4526	5.3788	5.6104
10/2.188	.20/.010	3.00/1.00	.600	10,000,000	5.3846	5.3788	
10/2.160	.20/.010	3.00/1.00	.000	100	5.2365	5.3242	5.3598
10/2.160	.20/.010	3.00/1.00	.000	10,000,000	5.3297	5.3242	
10/2.131	.20/.010	3.00/1.00	-.600	100	5.3173	5.2686	5.4849
10/2.131	.20/.010	3.00/1.00	-.600	10,000,000	5.2697	5.2686	
10/ .220	.40/.031	2.50/1.40	-.950	100	3.1561	3.1711	3.2065
10/ .220	.40/.031	2.50/1.40	-.950	10,000,000	3.2029	3.1711	

Note: $\rho_{\phi\sigma^2}$ denotes the correlation between ϕ and σ^2, which is $\frac{\tau_{\phi\sigma_2}}{\tau_\phi \tau_{\sigma^2}}$. The column *Eq. 3.8 pop.* contains the total variance estimates obtained by entering the population values into Equation 3.8. The column *Eq. 3.8 samp.* contains the total variance estimates obtained by entering the parameter estimates obtained in each generated dataset into Equation 3.8. Data were simulated and (where necessary) parameters were estimated using the methods described in Jongerling, Laurenceau, and Hamaker (2015). For $N = 10,000,000$ no parameters could be estimated because the software can not handle such large sample sizes. However, for samples this large the parameters estimates will be very close to the population values, and the total variance based on population values will therefore be virtually the same as the total variance based on sample estimates.

3.3 Proportion of Variance Explained and Intra-Class Correlation in the Two Level AR(1) Model

In the following sections we will provide expressions for different proportions of explained variance. Specifically, we will provide expressions for 1) the proportion of explained variance at level 1, 2) the proportion of explained variance at level 2, 3) the total proportion of explained variance, and for 4) the proportion of variance explained by autocorrelation. In addition, we will provide an expression for the Intra-Class Correlation (ICC) of the two level AR(1) model.

3.3.1 Proportion of Explained Variance at Level 1

As was the case for the AR(1) model, the innovations represent the unpredicted part, and so the innovation variance can be interpreted as the unexplained variance. In contrast to the AR(1) model however, the amount of innovation variance may differ across individuals in a two level AR(1) model, meaning that there is an overall mean amount of innovation variance on level 1 (μ_{σ^2}), and some inter-individual variability in the amount of innovation variance on level 2 ($\tau^2_{\sigma^2}$). In other words, the total innovation (or unexplained) variance is divided into two parts or parameters. One for each level of the model. We therefore define unexplained variance at level 1 of a multilevel AR(1) model as follows:

Definition 2 *The unexplained variance at the first level of a two level AR(1) model is that part of the variance of the innovations that is located at level 1. This implies,*

$$\sigma^2_{y,un1} = \mu_{\sigma^2}, \tag{3.9}$$

where $\sigma^2_{y,un1}$ denotes the unexplained variance in y at level 1. This result is obtained by setting μ_ϕ, τ^2_μ and τ^2_ϕ equal to 0 in Equation 3.8.

Following the same logic/reasoning, we define the total variance at level 1 of a two level AR(1) model as that part of the total variance that is *not* due to between-person (i.e., level 2) variance in the model parameters:

Definition 3 *The total variance at the first level of a two level AR(1) model is that part of the total variance not due to τ^2_μ and τ^2_ϕ. Removing these level 2 variance terms from Equation 3.8 renders,*

$$\sigma^2_{y,tot1} = \frac{\mu_{\sigma^2}}{1 - \mu^2_\phi}, \tag{3.10}$$

where $\sigma^2_{y,tot1}$ denotes the total variance in y at level 1.

Note that this expression is very similar to the expression for the total variance of the AR(1) model (Equation 3.3). The only difference is that in the two level AR(1) model the total variance at level 1 is determined using μ_ϕ and μ_{σ^2}, whereas in the AR(1)

model the total variance is determined using ϕ and σ^2. Further note that we did not include inter-individual variance in the innovation variance ($\tau^2_{\sigma^2}$) in the definition of the total level 1 variance, since Equation 3.8 shows that this term does not affect the total variance (i.e., Equation 3.8 does not include the term $\tau^2_{\sigma^2}$).

The explained variance at level 1 can now be obtained by dividing the unexplained variance at level 1 by the total variation at level 1, to get the proportion of unexplained variance, and subtracting the result from 1:

$$\begin{aligned} R^2_{level\ 1} &= 1 - \frac{\mu_{\sigma^2}}{\frac{\mu_{\sigma^2}}{1-\mu^2_\phi}} \\ &= 1 - \frac{\mu_{\sigma^2}\left(1-\mu^2_\phi\right)}{\mu_{\sigma^2}} \\ &= \mu^2_\phi. \end{aligned} \tag{3.11}$$

In multilevel literature the concept of explained variance is complex (Hox, 2010, pg. 70), especially in the presence of random slopes, because maximum likelihood estimates of level 1 and level 2 variance terms might be biased (Hox, 2010; Snijders & Bosker, 1994). However, since this is more a computational issue than a conceptual one, we will nevertheless use this theoretical conceptualization of level 1 variance being the variance associated with level 1 parameters.

3.3.2 Proportion of Explained Variance at Level 2

Definition 4 *We define the total variance at the second level of a two level AR(1) model as the difference between the total variance of a two level AR(1) model (Equation 3.8) and the total variance at the first level of the model (Equation 3.10):*

$$\begin{aligned} \sigma^2_{y|l2} \approx &\left(\frac{1}{1-(\mu^2_\phi+\tau^2_\phi)} + \frac{(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau^4_\phi}{(1-(\mu^2_\phi+\tau^2_\phi))^3} - \frac{1}{1-\mu^2_\phi}\right)\mu_{\sigma^2} \\ &+ \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu^2_\phi+\tau^2_\phi))^2} + \tau^2_\mu. \end{aligned} \tag{3.12}$$

Since there are no predictors at level 2 in our model, and since there are no level 1 predictors on which individuals differ in their mean scores (since $\mathbb{E}_t\left[\zeta_{i,t}\right] = \mathbb{E}_t\left[\zeta_{i,t-1}\right] = 0, \forall i$), the unexplained variance is also equal to Equation 3.12. Therefore,

$$R^2_{level2} = 0. \tag{3.13}$$

3.3.3 Proportion of Variance Related to Autocorrelation

Apart from the explained variance on the first and second level, a third type of variance can be determined for the two level AR(1) model; the variance not attributable to ϕ. We define this variance as follows:

Definition 5 *The variance in a two level AR(1) model that is not attributable to ϕ is obtained when $\mu_\phi = 0$ and $\tau_\phi^2 = 0$. Applying this to Equation 3.8 renders:*

$$\mu_{\sigma^2} + \tau_\mu^2. \tag{3.14}$$

The proportion of the variance that is related to autocorrelation can now be determined by dividing the amount of variance that is not related to ϕ (Equation 3.14) by the total variance in a two level AR(1) model (Equation 3.8), and subtracting the result from 1:

$$R_\phi^2 \approx 1 - \frac{\mu_{\sigma^2} + \tau_\mu^2}{\frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2}. \tag{3.15}$$

3.3.4 Total Proportion of Variance Explained at Level 1 and Level 2

From Equation 3.11 and Equation 3.13 it follows that the proportion of explained variance at level 1 is μ_ϕ^2 while the proportion of explained variance at level 2 is 0. This means that the total amount of explained variance is:

$$R_{total}^2 \approx \frac{\mu_\phi^2 * \sigma_{y,tot1}^2}{\frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2}. \tag{3.16}$$

Since the proportion of explained variance at level 1 is equal to μ_ϕ^2, this expression is closely related to the proportion of the variance that is related to ϕ (Equation 3.15). However, the two are not the same. This is because the inter-individual variance in the AR-parameter τ_ϕ^2 is not part of Equation 3.16, even though it is part of the variance associated with the AR-parameter in Equation 3.15.

3.3.5 Intra-Class Correlation for a Two Level AR(1) Model

Using Equation 3.12 we can also determine the intra-class correlation (ICC) for a two level AR(1) model, that is, the percentage of the total variance of this two level model that is located on the higher levels. The expression for the ICC is obtained by dividing the total variance on level 2 (Equation 3.12) by the total variance of the two level AR(1) model (Equation 3.8) to get,

$$ICC \approx \frac{\left(\frac{1}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4}{(1-(\mu_\phi^2+\tau_\phi^2))^3} - \frac{1}{1-\mu_\phi^2}\right)\mu_{\sigma^2} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \tau_\mu^2}{\frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2}. \quad (3.17)$$

3.4 Empirical Illustration

To illustrate the estimation of the proportions of explained variance and the ICC in the context of a two level AR(1) model, we will analyze data collected in a study by Laurenceau, Feldman Barrett, and Rovine (2005), that were also previously analyzed in Jongerling, Laurenceau, and Hamaker (2015). In this study, spouses from 96 married couples independently completed a structured diary each evening over a period of 42 consecutive days. We summed the scores on four items labeled excited, enthusiastic, energetic, and happy (all rated on 5-point Likert scales) to comprise a single positive affect PA score. Focusing on the women only, there were 127 out of the total of 96*42=4032 PA scores missing. Based on individual sequence plots (i.e., plots of the repeated measurements of each woman), we removed seven women who had none or very little variability over time, such that the final dataset contained 89 female participants.

To analyze the data, we used a Bayesian estimation method with uninformative normal priors for μ_μ, μ_ϕ, and μ_{σ^2} and an uninformative Inverse Wishart prior for the variance covariance matrix of these three parameters. A thorough explanation of this estimation method is available in Jongerling, Laurenceau, and Hamaker (2015), where it is referred to as method B1. The results are summarized in Tables 3.2 and 3.3, and in Figure 3.1. The first column of Table 3.2 contains the means and sd's of the posterior distribution displayed in Figure 3.1, while the second column contains the lower and upper bounds of the 95% central credibility intervals.

In Table 3.3 different types of variance and the ICC are presented. Using the parameter estimates from Table 3.2 it follows that the total variance is equal to 9.324, of which 4.862 is on level 1 and 4.463 is on level 2. The explained variance on level 1 is equal to 6.80%, while the total percentage of explained variance is equal to 3.50%. These amounts of explained variance are not very large, and can be seen as a strong indication that the inclusion of level 1 and level 2 predictors in the model is important (even though small amounts of explained variance obviously don't necessarily mean that the model is of little substantial interest. Model parameters might still be meaningfully related to important theoretical concepts). The importance of adding predictors on level 1 is further illustrated by the fact that the proportion of variance related to autoregression is equal to .037. Apparently, the only level 1 predictor currently in the model explains just 6.80% of the variance on level 1, and is related to just 3.70% of the total variance in the model. This shows that there is a lot of variance still unaccounted for. In addition, the importance of adding predictors on

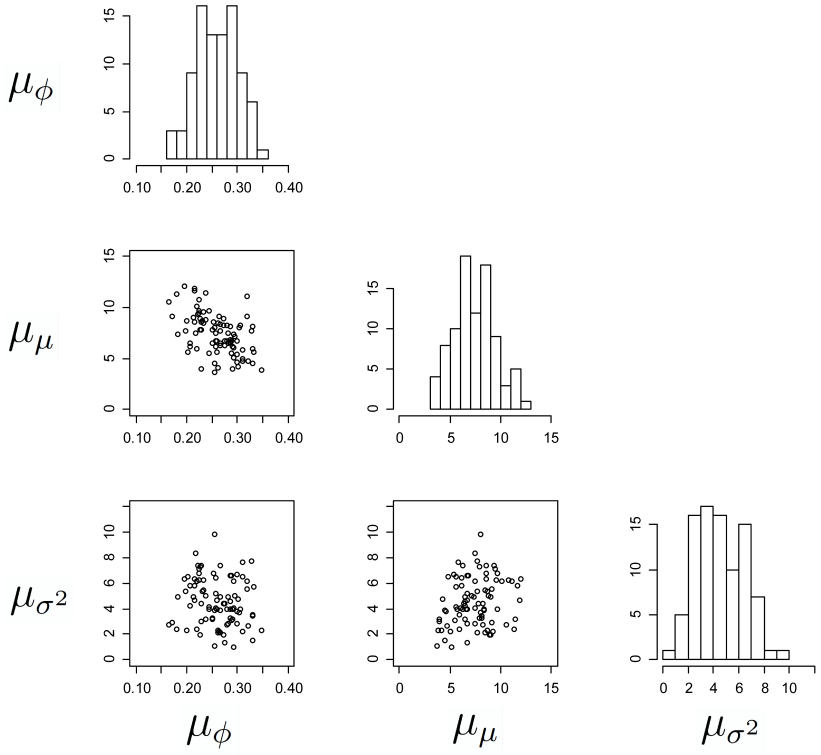


Fig. 3.1: The histograms show the posterior distributions of μ_ϕ, μ_μ, and μ_{σ^2}. The scatterplots show the bivariate relation between the parameters of the corresponding row and column (e.g, the scatterplot on the second row of the first column shows the relation between μ_μ and μ_ϕ).

level 2 is further illustrated by the ICC value of .479 (indicating that 47.9% of the total variance is located on the second level). Without predictors on this level of the model we are currently not explaining any variance at level 2, despite the fact that the amount of variance at this level is nearly half of the total variance. With such large parts of the variance on level 1 and level 2 being unrelated to the predictor in the model, it is no wonder it can explain only 3.50% of the total variance in the data.

Table 3.2: Estimates, standard deviations, and credible intervals for the parameters of the two level AR(1) model obtained using the data of Laurenceau, Feldman Barrett, and Rovine (2005)

	Parameter Estimate	95% Credibility Interval
Fixed Effects		
μ_μ	7.406 (.230)	6.955 - 7.857
μ_ϕ	.260 (.020)	.220 - .300
μ_{σ^2}	4.533 (.249)	4.052 - 5.034
Random Effects		
τ^2_μ	4.442 (.711)	3.260 - 6.026
τ^2_ϕ	.008 (.004)	.002 - .018
$\tau^2_{\sigma^2}$	4.389 (.880)	2.948 - 6.391
$\tau_{\mu\phi}$	-.234 (.216)	-.625 - .216
$\tau_{\mu\sigma^2}$	.199 (.115)	-.036 - .417
$\tau_{\phi\sigma^2}$	-.057 (.246)	-.532 - .419

The table shows the parameter estimates, standard deviations, and 95% Credibility Intervals of the Bayesian analysis of the daily positive affect data from Laurenceau, Feldman Barrett, and Rovine (2005). The standard deviations of the posterior distributions of the parameters are given between brackets.

Table 3.3: Variances and ICC for the data from Laurenceau, Feldman Barrett, and Rovine (2005)

	Unexplained	Explained	Total	R^2
Level 1	4.533 (Eq.3.9)	.329 (Eq.3.10 and 3.11)	4.862 (Eq.3.10)	.068 (Eq.3.11)
Level 2	4.463 (Eq.3.12)	.000 (Eq.3.12 and 3.13)	4.463 (Eq.3.12)	.000 (Eq.3.13)
Related to ϕ	.023	.329 (Eq.3.8 and 3.16)	.349 (Eq.3.8 and 3.15)	.037 (Eq.3.15)
Total	8.996	.329 (Eq.3.8 and 3.16)	9.324 (Eq. 3.8)	.035 (Eq.3.16)
ICC			.479 (Eq. 3.17)	

Between brackets the equations used to obtain the corresponding quantities are given. Note that for the explained variances information from two equations needs to be combined. For example, the explained variance at level 1 is calculated by first determining the total variance on this level using Equation 3.10, and the proportion of explained variance on this level using Equation 3.11. Subsequently, the specific amount of variance corresponding to this proportion of explained variance is determined by multiplying the results of Equations 3.10 and 3.11. The explained variance on level 2, the amount of explained variance related to ϕ, and the total amount of explained variance can be determined in a similar manner. The total amount of variance related to ϕ is calculated by first determining the total variance using Equation 3.8, and determining the proportion of variance related to ϕ using Equation 3.15. Subsequently, the specific amount of variance corresponding to this proportion is determined by multiplying the results of Equations 3.8 and 3.15. The amount of explained variance related to ϕ is again determined in a similar manner by multiplying the results of Equations 3.8 and 3.16. Finally, the unexplained variance related to ϕ is obtained by subtracting the explained variance related to ϕ from the total variance related to ϕ.

3.5 Conclusion

In this paper we derived an expression for the total variance of a two level AR(1) model. This expression is an elaboration on the well known, single-level expression for AR(1) models. Based on this expression we derived the proportion of explained variance (both total and on level 1 and level 2 separately), the proportion of variance related to the AR-parameter ϕ, and the intra-class correlation of a two level AR(1) model using parameter estimates. This was demonstrated with the diary data from the positive affect study by Laurenceau, Feldman Barrett, and Rovine (2005).

Unexpectedly, the expressions derived in the study also revealed that random variance in the innovation variance does not directly influence the total variance of a two level AR(1) model. Instead, inter-individual variances in the innovation variance only influences the total variance through their correlation with the AR-parameter. To us, this was an unexpected result, that has important implications. In Jongerling, Laurenceau, and Hamaker (2015) we argued that individual differences in the innovation variance τ_σ^2 are indicative of differential sensitivity and/or exposure to unmodeled factors. Taken together with the results found here, we can now conclude that such differential sensitivity and/or exposure will not show up in the total variance of an AR(1) model. Specifically, any inter-individual differences in these areas that are independent of the AR-parameter will go unnoticed when only looking at the total variance of the time series. This shows that the separate detection and modeling of individual differences in innovation variances is very important. A point already made in Jongerling, Laurenceau, and Hamaker (2015) as well.

For now we assumed that there were no (higher level) predictors in the model. Extending our two level AR(1) model to include predictors is straightforward as it only involves writing the means of the model parameters as functions of these predictors. However, doing so would make the variance structure of the model more complicated. Expressions for the proportion of explained variance and ICC for two level AR(1) models that include predictors are therefore the topic of future research.

3.6 Appendix A

For a simple univariate AR(1) model the well known expression for the total variance can be written as follows,

$$\sigma_y^2 = \frac{\sigma^2}{1-\phi^2}, \tag{3.18}$$

where σ_y^2 is the total variance of the time series, ϕ is the AR-parameter used to regress the current state on the previous one(s), and σ^2 is the error-variance, referred to as the innovation variance in time series literature, that represent variance that could not be predicted based on previous scores or states (and as such can be thought of as the collection of all unmodeled factors that influence the process under investigation Jongerling, Laurenceau, and Hamaker (2015)).

To extend this expression to a two level AR(1) model, we start with rewriting the total variance as the difference between two expected values (i.e., the expected value over time (t) of the squared repeated observations ($\mathbb{E}_t(y_t^2)$) minus the squared expectation over time of the repeated observations ($\mathbb{E}_t(y_t)^2$))(Equation 3.19):

$$\mathbb{E}_t(y_t^2) - \mathbb{E}_t(y_t)^2 = \frac{\sigma^2}{1-\phi^2}. \tag{3.19}$$

Next, we take the expected value over all individuals on both sides of the equality sign (Equation 3.20):

$$\begin{aligned}
\mathbb{E}_i[\mathbb{E}_t(y_{it}^2) - \mathbb{E}_t(y_{it})^2] &= \mathbb{E}_i[\frac{\sigma_i^2}{1-\phi_i^2}] \\
\mathbb{E}_{it}(y_{it}^2) - \mathbb{E}_i(\mu_i^2) &= \mathbb{E}_i[\frac{\sigma_i^2}{1-\phi_i^2}] \\
\mathbb{E}_{it}(y_{it}^2) - [\mathbb{E}_i(\mu_i)^2 + \tau_\mu^2] &= \mathbb{E}_i[\frac{\sigma_i^2}{1-\phi_i^2}] \\
\mathbb{E}_{it}(y_{it})^2 + \sigma_y^2 - \mathbb{E}_i(\mu_i)^2 - \tau_\mu^2 &= \mathbb{E}_i[\frac{\sigma_i^2}{1-\phi_i^2}] \\
\sigma_y^2 &= \mathbb{E}_i[\frac{\sigma_i^2}{1-\phi_i^2}] + \tau_\mu^2.
\end{aligned} \tag{3.20}$$

Since there are no simple exact formulas for the mean of a quotient, we eliminate the expected value of the quotient on the right side of the equation by using Taylor-series (Mood et al., 1985, pg. 181). A Taylor-series can be used to approximate the value of a function around a specific value, and, in general, the Taylor-series for function *f(x)* around the value a can be written as,

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n, \tag{3.21}$$

where $f^{(n)}(a)$ denotes the nth derivative of f evaluated at the value a. The more derivatives of *f(x)* used in the series, the more precise the approximation of *f(a)*. Here, second order Taylor-series (i.e., Taylor series in which the highest used derivative is the second order derivative) were used. The Taylor series expansion for the product of two random variables is equal to the product of the Taylor-series that can be derived for the variables separately. So, to get the Taylor-series for the quotient $\frac{\sigma_i^2}{1-\phi_i^2}$ about $(\mu_{\sigma^2}, \mu_{[1-\phi^2]})$, we first consider σ_i^2 and $1-\phi_i^2$ our two random variables, and construct separate second order Taylor-series for the function $f(\sigma_i^2) = \sigma_i^2$ and the function $g(1-\phi_i^2) = \frac{1}{1-\phi_i^2}$. For $f(\sigma_i^2)$, the second order Taylor-series around μ_{σ^2} is equal to,

$$f(\mu_{\sigma^2}) = \sigma_i^2 + (\sigma_i^2 - \mu_{\sigma^2}). \tag{3.22}$$

Note that the second- and higher order terms in this Taylor-series are 0 because we take σ_i^2 as our parameter of interest. The first derivative of σ_i^2 then is equal to 1, while the second and higher order derivatives are 0.

For $g(1-\phi_i^2)$, the second order Taylor-series around $\mu_{[1-\phi^2]}$ is equal to,

$$g(\mu_{[1-\phi^2]}) \approx \frac{1}{(1-\phi_i^2)} - \frac{1}{(1-\phi_i^2)^2}((1-\phi_i^2) - \mu_{[1-\phi^2]}) + \frac{1}{(1-\phi_i^2)^3}((1-\phi_i^2) - \mu_{[1-\phi^2]})^2. \tag{3.23}$$

By subsequently multiplying these two Taylor-series term by term we get the following second order Taylor-series for the quotient of σ_i^2 and $1-\phi_i^2$,

$$\begin{aligned} f(\mu_{\sigma^2})g(\mu_{[1-\phi^2]}) \approx & \frac{\sigma_i^2}{(1-\phi_i^2)} - \frac{\sigma_i^2}{(1-\phi_i^2)^2}((1-\phi_i^2) - \mu_{[1-\phi^2]}) \\ & + \frac{\sigma_i^2}{(1-\phi_i^2)^3}((1-\phi_i^2) - \mu_{[1-\phi^2]})^2 \\ & + \frac{(\sigma_i^2 - \mu_{\sigma^2})}{(1-\phi_i^2)} - \frac{(\sigma_i^2 - \mu_{\sigma^2})}{(1-\phi_i^2)^2}((1-\phi_i^2) - \mu_{[1-\phi^2]}) \\ & + \frac{(\sigma_i^2 - \mu_{\sigma^2})}{(1-\phi_i^2)^3}((1-\phi_i^2) - \mu_{[1-\phi^2]})^2. \end{aligned} \tag{3.24}$$

When we now take the expectation of both sides of Equation 3.24 this simplifies to,

$$\mathbb{E}_i\left[f(\mu_{\sigma^2})g(\mu_{[1-\phi^2]})\right] \approx \frac{\mu_{\sigma^2}}{\mu_{[1-\phi^2]}} - \frac{\tau_{(\sigma^2,1-\phi^2)}}{(\mu_{[1-\phi^2]})^2} + \frac{\mu_{\sigma^2}\tau_{[1-\phi^2]}^2}{(\mu_{[1-\phi^2]})^3}. \tag{3.25}$$

Substituting this expression for the expectation of the quotient in Equation 3.20 we get,

$$\begin{aligned} \sigma_y^2 &\approx \frac{\mu_{\sigma^2}}{\mu_{[1-\phi^2]}} - \frac{\tau_{(\sigma^2,1-\phi^2)}}{(\mu_{[1-\phi^2]})^2} + \frac{\mu_{\sigma^2}\tau_{[1-\phi^2]}^2}{(\mu_{[1-\phi^2]})^3} + \tau_\mu^2 \\ &\approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{\rho_{(\sigma^2,\phi^2)}\tau_{\sigma^2}\tau_{\phi^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}\tau_{\phi^2}^2}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2. \end{aligned} \tag{3.26}$$

This takes care of all the expected values, but the expression still contains some random variables that are not directly modeled, like $\tau_{\phi^2}^2$ instead of τ_ϕ^2 for example. To make sure our expression for the total variance of a two level AR(1) model only contains parameters that are directly modeled, we view the square of the AR-parameters

as the product of two random variables, that is, as the product of the AR-parameters with themselves. Aroian (Aroian, 1947) showed that the standard deviation of the product of two random variables θ_1 and θ_2 can be written as,

$$\tau_{\theta_1\theta_2} = \tau_{\theta_1}\tau_{\theta_2}\sqrt{\left(\frac{\mu_{\theta_1}}{\tau_{\theta_1}}\right)^2 + \left(\frac{\mu_{\theta_2}}{\tau_{\theta_2}}\right)^2 + 2\rho\left(\frac{\mu_{\theta_1}}{\tau_{\theta_1}}\right)\left(\frac{\mu_{\theta_2}}{\tau_{\theta_2}}\right) + 1 + \rho^2}, \tag{3.27}$$

where ρ is the correlation between the two random variables. Applying this to our "product" of AR-parameters this expression becomes,

$$\begin{aligned}\tau_{\phi\phi} &= \tau_\phi^2\sqrt{\left(\frac{\mu_\phi}{\tau_\phi}\right)^2 + \left(\frac{\mu_\phi}{\tau_\phi}\right)^2 + 2\left(\frac{\mu_\phi}{\tau_\phi}\right)\left(\frac{\mu_\phi}{\tau_\phi}\right) + 2} \\ &= \tau_\phi^2\sqrt{4\left(\frac{\mu_\phi}{\tau_\phi}\right)^2 + 2}.\end{aligned} \tag{3.28}$$

Substituting this expression for τ_{ϕ^2} (and the square of this expression for $\tau_{\phi^2}^2$) in Equation 3.26 results in,

$$\sigma_y^2 \approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{\rho_{(\sigma^2,\phi^2)}\tau_{\sigma^2}\sqrt{(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2. \tag{3.29}$$

Finally, the correlation between the innovation variance and the square of the AR-parameter ($\rho_{(\sigma^2,\phi^2)}$) can also be interpreted as the correlation between a random variable (the innovation variance) and the product of a random variable (the AR-parameter multiplied by itself). According to Bohrnstedt and Goldberger (1969), the covariance between the product of the two random variables x and y, and a third random variable v, can be written as,

$$\tau_{xy,v} = \mathbb{E}(x)\tau_{y,v} + \mathbb{E}(y)\tau_{x,v} + \mathbb{E}\left[(\Delta x)(\Delta y)(\Delta v)\right], \tag{3.30}$$

where $\Delta x = x - \mathbb{E}(x)$, $\Delta y = y - \mathbb{E}(y)$, and $\Delta v = v - \mathbb{E}(v)$. Since the product in our correlation is the product of one random variable with itself, and since $\tau_x^2 = (x-\mathbb{E}(x))^2$, the correlation between the innovation variance and the square of the AR-parameter can be written as,

$$\rho_{(\sigma^2,\phi^2)} = \frac{2\mu_\phi\tau_{\phi\sigma^2} + \mathbb{E}_i(\tau_\phi^2(\sigma_i^2 - \mu_{\sigma^2}))}{\tau_{\sigma^2}\sqrt{(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4}}, \tag{3.31}$$

where the term under the square root in the denominator can again be recognized from Equation 3.28 as the standard deviation of ϕ^2, and the term $\mathbb{E}_i(\tau_\phi^2(\sigma_i^2 - \mu_{\sigma^2}))$

will always be relatively small (since μ_{σ^2} is the mean value of σ^2) and can therefore be ignored.

Substitution in Equation 3.29 results in the following expression for the total variance of a two level AR(1) model,

$$\sigma_y^2 \approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2. \tag{3.32}$$

Note, that inter-individual variance in the innovation variance does not directly influence the total variance. This follows from the fact that Equation 3.32 does not contain the term $\tau_{\sigma^2}^2$. Instead, random variance in the innovation variances only influences σ_y^2 through its correlation with the AR-parameter as can be seen by the covariance term $\tau_{\phi\sigma^2}$ included on the right-hand side of Equation 3.32.

Part II

The Autoregressive Latent Trajectory Model

4

On the Trajectories of the Predetermined ALT Model: What are we really modeling?

Summary. This paper shows that the mean and covariance structure of the predetermined autoregressive latent trajectory (ALT) model are very flexible. As a result, the shape of the modeled growth curve can be quite different from what one may expect at first glance. This is illustrated with several numerical examples which show that, for example, a linear trajectory may be present among the model predicted scores even though no latent change parameter was included in the model. In addition, two examples are given that show that the predetermined ALT model can fit to data generated by models with model structures that are rather different from that of the ALT model itself. The practical relevance of these findings is demonstrated using an empirical example. We end by providing recommendations for researchers considering the use of the predetermined ALT model.

4.1 Introduction

The *autoregressive latent trajectory (ALT) model* was introduced by Curran and Bollen (2001) (also see Bollen & Curran, 2004) and forms a combination of two popular longitudinal models, that is, the *simplex model* and the *latent growth curve (LGC) model*. While both these models can be easily fitted to longitudinal data using standard Structural Equation Modeling (SEM) software, the simplex model and LGC model represent different change processes. The simplex model was developed to

This chapter is published as Jongerling, J., & Hamaker, E.L. (2011). On the Trajectories of the Predetermined ALT Model: What are we really modeling? *Structural Equation Modeling: A Multidisciplinary Journal*, *18*(3), 370-382.

J. Jongerling wrote the article, wrote the computer code needed for the analyses, and ran the analyses.

E.L. Hamaker supervised the study and provided feedback on the analyses and the paper.

model longitudinal data in which observations that are closer together in time are more highly correlated than observations that are further apart (Jöreskog, 1971, 1979). In this model an autoregressive (AR) parameter is estimated that, when standardized, represents the stability in the rank order of scores between different measurement occasions. Although it is possible to extend the model with structured means (e.g., Browne & Du Toit, 1991; Mandys, Dolan, & Molenaar, 1994), typically only the covariance structure is analyzed. In contrast, the LGC model is used to model individual trajectories of change over time (Meredith & Tisak, 1990). The focus is on the average trajectory, or growth curve, and on individual deviations in the parameters that describe this curve (Bollen & Curran, 2004). What is estimated are the means and (co)variances of these individual growth parameters across all individuals, and hence the LGC model is based on modeling both the covariance and the mean structure.

The ALT model is an integration of these two models, and can be described as a LGC model with AR relationships between the observed variables (Curran & Bollen, 2001; Bollen & Curran, 2004). Because of the way that the AR relationships are incorporated in the ALT model, the model is recursive and needs to be "started up" in order to be used in practice (Curran & Bollen, 2001; Bollen & Curran, 2004). Bollen and Curran (2004) suggest two ways for doing this, but in this article we mainly focus on the start up method that treats the first observation as a predetermined, exogenous variable. Advantages of this method, in comparison to the alternative start up procedure, are that it does not require constraining the AR relations to be equal over time, and that it does not require non-linear constraints on some of the model parameters at the first measurement occasion. We will show that in an *predetermined* ALT model, that is, an ALT model with a predetermined first observation, the shape of the growth curve that is being modeled can be quite different from what one may expect at first glance. For example, a linear mean trajectory may be present even though no change parameter was included in the model.

This paper is organized as follows. First, a short introduction to the simplex and LGC model is given. In addition, the ALT model is presented and both start up methods suggested by Bollen and Curran (2004) are discussed. Second, several numerical examples are given which illustrate that the growth curve in the predetermined ALT model may differ from what one may initially suspect. Third, we show that, due to their flexibility, predetermined ALT models without a quadratic term can adequately fit covariance structures generated by a quadratic growth model. Fourth, an empirical example is presented. We end by giving recommendations for researchers who consider using the ALT model.

4.2 Three Models for Longitudinal Data

4.2.1 The Simplex Model

The simplex model was developed to model longitudinal data that are characterized by decreasing correlations as the distance between observations increases. The top

half of Figure 4.1 shows the simplex model for four waves of data. It shows that each observation is the weighted sum of its preceding value and a random error term that represents variation that cannot be predicted from the previous observation. This unpredicted part is also referred to as innovation in time series literature. The ρ's are the AR parameters, which are used for the regression of each observation on its immediate preceding value. The standardized values of these AR parameters indicate the stability of the rank order of the scores.

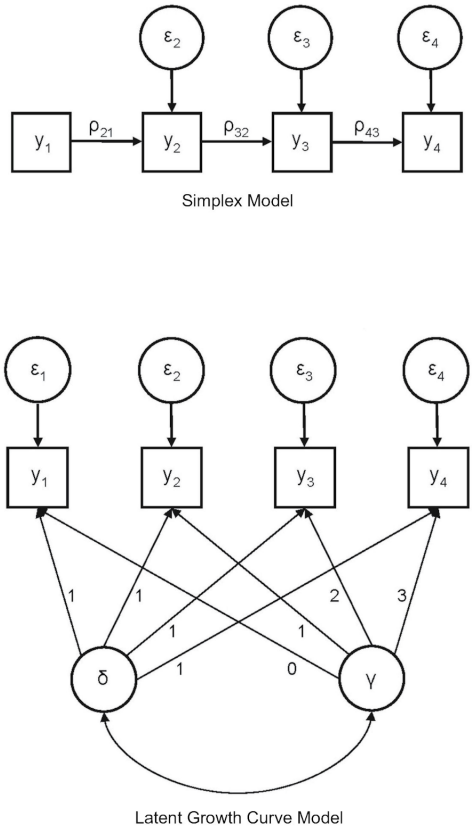


Fig. 4.1: Simplex model and LGC model for four repeated measures

Following Bollen and Curran (2004), the simplex model with mean structure can be written as

$$y_{i,t} = \delta_t + \rho_{t,t-1} y_{i,t-1} + \varepsilon_{i,t}, \tag{4.1}$$

where $y_{i,t}$ is the score of person i at time t, δ_t is the intercept at time t, $\rho_{t,t-1}$ is the AR parameter between the observations at time t and time $t-1$, and $\varepsilon_{i,t}$ is a random error term with zero mean and variance of $\sigma^2_{\varepsilon t}$. The error terms are assumed uncorrelated across individuals and time, and uncorrelated with previous observations. Sometimes the AR parameters are fixed across time (i.e., $\rho_{t,t-1} = \rho$), which implies that the relation between adjacent time points is assumed to be the same across the whole range of time points considered.

The mean trajectory of the simplex model is given by

$$\mu_1 = \delta_1 \qquad \text{for } t = 1 \tag{4.2a}$$

$$\mu_t = \rho_{t,t-1}\mu_{t-1} + \delta_t \qquad \text{for } t = 2, \ldots, T, \tag{4.2b}$$

where T is the total number of time points. Note that Equations 4.2a and 4.2b shows that the mean structure of the simplex model is saturated.

4.2.2 Latent Growth Model

The LGC model was introduced to model individual differences in trajectories through analyzing the mean and covariance structure simultaneously. The bottom half of Figure 4.1 shows a LGC model with an intercept and linear slope for four waves of data. Each observation in this model is a function of the individual's intercept (δ_i), his/her (linear) slope parameter (γ_i), and a random error term ($\varepsilon_{i,t}$). Following Bollen and Curran (2004), we fix the factor loading of the first measurement occasion to 0, making this our reference point. A linear curve is subsequently captured by fixing the other factor loadings at $t - 1$.

The LGC model with a linear slope can be expressed as

$$y_{i,t} = \delta_i + \gamma_i(t-1) + \varepsilon_{i,t}, \tag{4.3}$$

where $\varepsilon_{i,t}$ has a zero mean and variance of $\sigma^2_{\varepsilon t}$. The residuals, $\varepsilon_{i,t}$, are assumed uncorrelated across individuals and time, and uncorrelated with the random intercept and random slope. If the mean intercept and mean slope across all individuals are represented by μ_δ and μ_γ respectively, then the individual intercept and slope can be written as

$$\delta_i = \mu_\delta + \zeta_{\delta i}, \text{ and} \tag{4.4a}$$

$$\gamma_i = \mu_\gamma + \zeta_{\gamma i}, \tag{4.4b}$$

where $\zeta_{\delta i}$ and $\zeta_{\gamma i}$ are individual deviations from the mean intercept and slope. These individual deviations have zero means and are uncorrelated with the error term $\varepsilon_{i,t}$. However, they are allowed to be correlated with each other.

It follows from Equations 4.3, 4.4a, and 4.4b that the mean trajectory for this model can be expressed as

$$\mu_t = \mu_\delta + (t-1)\mu_\gamma \text{ for } t = 1, \ldots, T. \tag{4.5}$$

Note that if μ_γ and σ^2_γ are both equal to 0, the model in Equation 4.3 simplifies to an intercept only LGC model, which implies there is no structural linear change over time for any of the individuals.

4.2.3 ALT model

The ALT model combines the two models introduced above and can be described as a LGC model with AR relations between the observed variables (Bollen & Curran, 2004). It differs from LGC models with autocorrelated disturbances, such as proposed by Chi and Reinsel (1989), although it can be shown that these two models are algebraically equivalent when the AR parameter is invariant over time (Hamaker, 2005). The ALT model can be expressed as

$$y_{i,t} = \alpha_i + \beta_i(t-1) + \rho_{t,t-1}y_{i,t-1} + \varepsilon_{i,t}, \tag{4.6}$$

where α_i is a constant, β_i is a change parameter, $\rho_{t,t-1}$ is an AR parameter, and $\varepsilon_{i,t}$ is a random error term that is subject to the same assumptions as under the simplex and LGC model. It is crucial to note that (unless all $\rho_{t,t-1} = 0$) the parameters α_i and β_i do not represent the intercept and slope of the individual growth curves. Why this is the case and what the consequences are will become apparent below.

Equation 4.6 shows that $y_{i,t}$ is a function of α_i, β_i, and $y_{i,t-1}$. However, this last term is a function of α_i, β_i and $y_{i,t-2}$, and so on. Due to this recursion, the ALT model has to be started up if it is to be used in practice. Curran and Bollen (2001) suggested two ways to do this (see Hamaker, 2005), which result in two different forms of the ALT model that we refer to as the *constrained* ALT model and the *predetermined* ALT model. While the focus here is on the predetermined ALT model, we briefly introduce the constrained form for comparison and to indicate how the parameters α_i and β_i are related to the intercept and slope of the individual trajectories.

Constrained ALT Model

The first start up method consists of restricting the AR parameters $\rho_{t,t-1}$ to be fixed across time (i.e., $\rho_{t,t-1} = \rho$ for all t) and smaller than 1 in absolute value (i.e., $|\rho| < 1$). Under these restrictions, Bollen and Curran (2004) show that the first observation can be written as,

$$y_{i,1} = \alpha_i(1-\rho)^{-1} - \beta_i\rho(1-\rho)^{-2} + z_{i,1}, \text{ where} \tag{4.7}$$

$$\begin{aligned} z_{i,1} &= \varepsilon_{i,1} + \rho\varepsilon_{i,0} + \rho^2\varepsilon_{i,-1} + \cdots + \\ &= \sum_{j=0}^{\infty} \rho^j \varepsilon_{i,1-j}. \end{aligned} \tag{4.8}$$

If the residual variation is equal across time, the term $z_{i,1}$ can be seen as an infinite weighted sum of previous residuals with zero mean and variance σ^2_z which is a function of σ^2_ε and ρ (Hamaker, 2005). Note that this start up method puts nonlinear

constraints on α_i (i.e., $(1-\rho)^{-1}$) and β_i (i.e., $\rho(1-\rho)^{-2}$) at the first measurement occasion. After the process is started up at $t = 1$, the model defined in Equation 4.6 can be used for the subsequent occasions (i.e., for $t = 2, 3, ...$). The top half of Figure 4.2 shows the constrained ALT model for four waves of data.

Hamaker (2005) showed that the intercept and slope of the growth trajectories of the constrained ALT model can be written as

$$\delta_i = \alpha_i(1-\rho)^{-1} - \beta_i\rho(1-\rho)^{-2}, \text{ and} \tag{4.9a}$$

$$\gamma_i = \beta_i(1-\rho)^{-1}, \tag{4.9b}$$

which shows that the model parameters α_i and β_i cannot be interpreted as the intercept and slope of the growth trajectories in the constrained ALT model.

Predetermined ALT Model

The second start up method suggested by Bollen and Curran (2004) involves treating the first observation as a predetermined, exogenous variable. In that case, the first observation ($y_{i,1}$) can be written as an unconditional mean (ν) and an individual deviation from this mean ($\varepsilon_{i,1}$), that is

$$y_{i,1} = \nu + \varepsilon_{i,1}. \tag{4.10}$$

To account for the recursiveness in the process prior to $t = 1$, $y_{i,1}$ is allowed to correlate with α_i and β_i. After the process is started up at $t = 1$, Equation 4.6 can be used to model subsequent occasions. The predetermined ALT model is depicted in the bottom half Figure 4.2. Two advantages of this second method, in comparison to the previous one, are that it does not require the assumption of time invariant ρ parameters, and that it does not put nonlinear constraints on the factor loadings of α_i and β_i at the first measurement occasion. On the downside however, when the first observation is treated as predetermined, at least five waves of data are needed to identify a model with both a constant and a change parameter. Moreover, as Bollen and Curran (2004) pointed out, treating $y_{i,1}$ as predetermined has the effect that the standard LGC model is no longer nested under the ALT model.

In contrast to the constrained ALT model, where the analytic relationship between the parameters of the ALT model and the intercept and slope of the linear trajectory can be derived (Hamaker, 2005), in the predetermined ALT model no such derivation is possible. However, the conclusion that the parameters α_i and β_i cannot be interpreted as the intercept and slope of the growth trajectories applies to this form of the ALT model as well. This becomes clear when looking at the mean structure of the predetermined ALT model,

$$\mu_1 = \nu \qquad \text{for} \quad t = 1 \tag{4.11a}$$

$$\mu_t = \mu_\alpha + (t-1)\mu_\beta + \rho_{t,t-1}\mu_{t-1} \qquad \text{for} \quad t = 2, \ldots, T., \tag{4.11b}$$

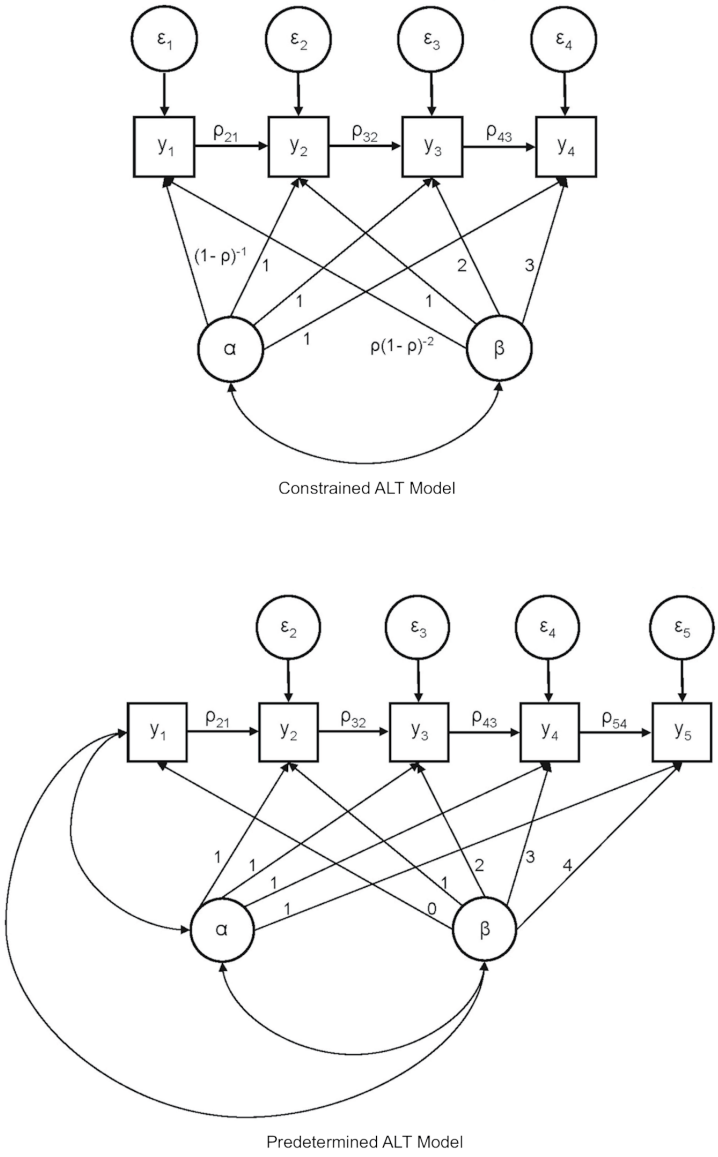


Fig. 4.2: Constrained ALT model for four repeated measures and Predetermined ALT model for five repeated measures

which shows that μ_α and μ_β recursively enter into the formula for μ_t through μ_{t-1}.

In the next section we will show that the mean structure of the predetermined ALT model is rather flexible, even if the model contains just a latent constant (i.e., $\beta_i = 0$ for all i), and that the mean trajectory in this model can be quite different from what one may initially expect.

4.3 Mean Trajectories in the Predetermined ALT Model

We begin with a simple predetermined ALT model which contains no change parameter but only a constant. This model seems analogous to the intercept only LGC model, and therefore one may expect a horizontal curve is modeled. However, the following numerical examples show that this is not necessarily the case.

First, suppose that there are 5 measurement occasions ($t = 1, \ldots, 5$) and that the mean of the first (predetermined) observation ν is equal to 50. Furthermore, suppose that the mean of the latent constant μ_α is 40 and that the AR parameters ρ are time invariant and equal to .2. Using these parameter values in Equations 4.11a and 4.11b (from which the terms involving μ_β have been dropped), we find that the modeled mean for each measurement occasions is equal to 50. These means are plotted in panel a of Figure 4.3. Note that even though the model does not involve change (as we may have expected since the change parameter μ_β was set to zero), $\mu_\alpha = 40$ does not represent the mean intercept in this model. If ν is changed to 55, while the other parameters remain unchanged, the mean trajectory is decreasing at a diminishing rate, much like a quadratic trend, as depicted in panel b of Figure 4.3. If $\nu = 45$, while μ_α and ρ remain unchanged, the mean trajectory, shown in panel c of Figure 4.3, is increasing and quadratic in appearance. Finally, changing the values of the AR parameters also affects the modeled means. For example, if $\nu = 50$ and $\mu_\alpha = 40$ (as in the first example), but the AR parameter is allowed to vary across time and set to .2, .3, .2, and .3 respectively, then Equation 4.11a results in modeled means of 50, 50, 55, 51, and 55.3. The corresponding mean growth curve is depicted in panel d of Figure 4.3, and shows an oscillating process as a result of the increasing and decreasing AR parameters.

Next we consider the mean trajectory of an ALT model with a constant as well as a change parameter. If $\nu = 56$, $\mu_\alpha = 40$, $\mu_\beta = 1$, and the time invariant AR parameter ρ is equal to .3, the mean growth curve is an almost straight line as depicted in panel a of Figure 4.4. However, if ν is changed to 46, while all other parameters remain unchanged, the mean trajectory is an increasing curve that seems more like a quadratic trend than a straight line. This can be seen in panel b of Figure 4.4. If $\nu = 60$, while μ_α, μ_β, and ρ are unchanged, the mean trajectory becomes downward sloped and quadratic in appearance, as is shown in panel c of Figure 4.4. Finally, if $\nu = 56$, $\mu_\alpha = 40$, and $\mu_\beta = 1$ (as in panel a of Figure 4.4), but the AR parameter varies across time and is set to .2, .3, .2, and .3 respectively, the mean growth curve, depicted in panel d of Figure 4.4, is characterized by repeated increases and decreases as a result of the increasing and decreasing AR relations.

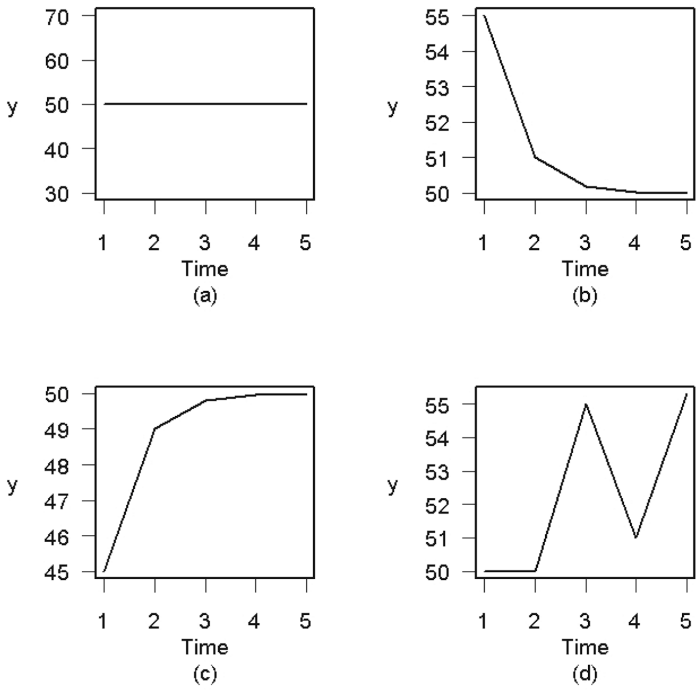


Fig. 4.3: Actual Growth Trajectories in an ALT model with only a latent intercept. Panel (a) shows actual (model predicted) growth trajectory for $\nu = 50$, $\mu_\alpha = 40$, and $\rho = 0.2$. Panel (b) shows the growth trajectory for $\nu = 55$, $\mu_\alpha = 40$, and $\rho = 0.2$. Panel (c) shows the growth trajectory for $\nu = 45$, $\mu_\alpha = 40$, and $\rho = 0.2$. Panel (d) shows the growth trajectory for $\nu = 50$, $\mu_\alpha = 40$, and $\rho_{2,1} = \rho_{4,3} = 0.2$ and $\rho_{3,2} = \rho_{5,4} = 0.3$

These examples shows that, despite the apparent resemblance, the shape of the mean trajectory in a predetermined ALT model may deviate considerably from a seemingly analogue LGC model.

4.4 Covariance Structures in the Predetermined ALT Model

In the previous section we showed that a predetermined ALT model with just a constant, or with a constant and change parameter can generate mean growth curves that are quadratic in appearance (panels b and c of Figures 4.3 and 4.4). This raises the question whether this model fits data that come from a quadratic growth model. To this end we generate a population covariance matrix and population means for 5 waves of data using a growth model with an intercept, linear slope, and quadratic

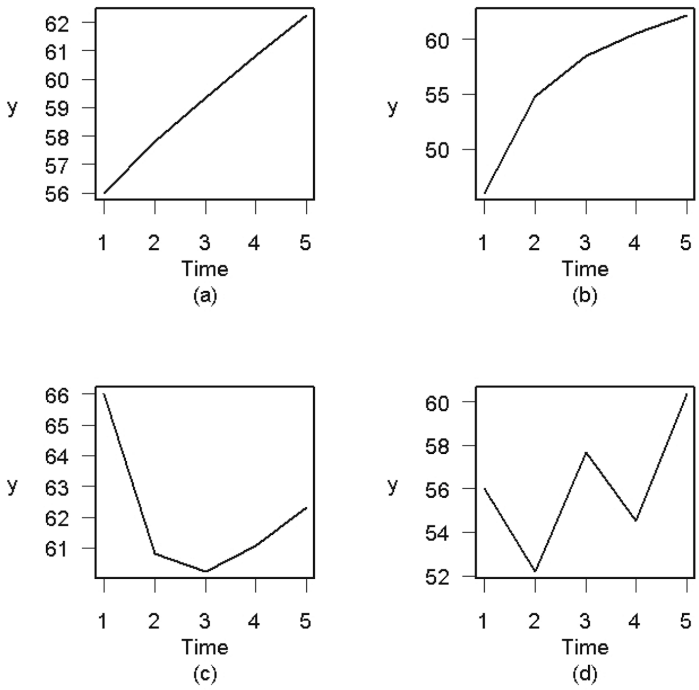


Fig. 4.4: Actual Growth Trajectories in an ALT model with latent intercept and linear growth parameter. Panel (a) shows actual (model predicted) growth trajectory for $\nu = 56$, $\mu_\alpha = 40$, $\mu_\beta = 1$, and $\rho = 0.3$. Panel (b) shows the growth trajectory for $\nu = 46$, $\mu_\alpha = 40$, $\mu_\beta = 1$, and $\rho = 0.3$. Panel (c) shows the growth trajectory for $\nu = 66$, $\mu_\alpha = 40$, $\mu_\beta = 1$, and $\rho = 0.3$. Panel (d) shows the growth trajectory for $\nu = 56$, $\mu_\alpha = 40$, $\mu_\beta = 1$, and $\rho_{2,1} = \rho_{4,3} = 0.2$ and $\rho_{3,2} = \rho_{5,4} = 0.3$

slope parameter. The equation for this model is

$$y_{i,t} = \delta_i + \gamma_i(t-1) + \xi_i(t-1)^2 + \varepsilon_{i,t}, \tag{4.12}$$

where δ_i is the random intercept, γ_i is the random linear slope parameter, and ξ_i is the random quadratic slope parameter.

First, suppose that the means of the latent intercept, linear slope, and quadratic slope are 130, 40, and -5 respectively, while their variances are equal to 240, 50, and 4.5. In addition, suppose that $\sigma_{\delta\gamma} = .32$, $\sigma_{\delta\xi} = -3$, $\sigma_{\gamma\xi} = .83$, and that the residual variances at the five measurement occasions are 45, 48, 55, 70, and 110. With these parameter values the mean trajectory is increasing with a decreasing rate of change (panel a of Figure 4.5). The covariances are given in Table 4.1. When a predetermined ALT model with a constant and a change parameter is fitted to these means and

this covariance matrix, using a sample size of 215, we obtain $\chi^2(3) = 6.22, p = .10$, CFI = .997, SRMR = .016, and RMSEA = .071 (90% confidence interval .00 - .15). Furthermore, all parameters are significantly different from 0 with the exception of $\hat{\rho}_{2,1}$, $\hat{\sigma}^2_{\alpha}$, and $\hat{\sigma}_{\alpha\beta}$. The significant parameter estimates (with standard errors between parentheses) are $\hat{\rho}_{3,2} = 0.34$ (0.17), $\hat{\rho}_{4,3} = 0.47$ (0.17), $\hat{\rho}_{5,4} = 0.55$ (0.18), $\hat{\nu} = 130.00$ (1.15), $\hat{\mu}_{\alpha} = 169.94$ (26.85), $\hat{\mu}_{\beta} = -18.28$ (8.11), $\hat{\sigma}^2_{y_1} = 283.68$ (27.36), and $\hat{\sigma}^2_{\beta} = 61.13$ (28.80), while $\hat{\sigma}^2_{\varepsilon_2} = 68.06$ (23.13), $\hat{\sigma}^2_{\varepsilon_3} = 79.39$ (14.39), $\hat{\sigma}^2_{\varepsilon_4} = 121.11$ (30.47), $\hat{\sigma}^2_{\varepsilon_5} = 174.71$ (33.96), $\hat{\sigma}_{y_1\alpha} = 246.85$ (69.19) and $\hat{\sigma}_{y_1\beta} = -45.71$ (17.07).

Table 4.1: Covariance matrix, means, and residual variances for $\mu_{\delta} = 130$, $\mu_{\gamma} = 40$, $\mu_{\xi} = -5$, $\sigma^2_{\delta} = 240$, $\sigma^2_{\gamma} = 50$, $\sigma^2_{\xi} = 4.5$, $\sigma_{\delta\gamma} = .32$, $\sigma_{\delta\xi} = -3$, $\sigma_{\gamma\xi} = .83$

	Y1	Y2	Y3	Y4	Y5
Y1	285.00				
Y2	237.32	338.80			
Y3	228.64	348.94	557.56		
Y4	213.96	411.74	689.50	1117.24	
Y5	193.28	479.20	909.76	1484.96	2314.80
Means	130	165	190	205	210
Residual Variances	45	48	55	70	110

Next, suppose that $\mu_{\delta} = 225$, $\mu_{\gamma} = -86.4$, and $\mu_{\xi} = 12.3$, while $\sigma^2_{\delta} = 70$, $\sigma^2_{\gamma} = 170$, and $\sigma^2_{\xi} = 2.7$. Further presume that $\sigma_{\delta\gamma} = -.5$, $\sigma_{\delta\xi} = 3.5$, $\sigma_{\gamma\xi} = -.83$, $\sigma^2_{\epsilon 1} = 39$, $\sigma^2_{\epsilon 2} = 58$, $\sigma^2_{\epsilon 3} = 88$, $\sigma^2_{\epsilon 4} = 118$, and $\sigma^2_{\epsilon 5} = 138$. These parameter values result in a mean trajectory that is decreasing with a diminishing rate of change (panel b Figure 4.5). The corresponding covariances are given in Table 4.2. When the predetermined ALT model with a constant and change parameter is fitted to this data (again using a sample size of 215), the model fits moderately well with $\chi^2(3) = 7.806, p = .05$, CFI = .996, SRMR = .020, and RMSEA = .086 (90% confidence interval .00 - .163). The significant parameter estimates (with standard errors between parentheses) are $\hat{\rho}_{2,1} = 0.38$ (0.11), $\hat{\rho}_{3,2} = 0.25$ (0.13), $\hat{\nu} = 225.00$ (0.71), $\hat{\mu}_{\alpha} = 68.88$ (28.00), $\hat{\sigma}^2_{y_1} = 108.49$ (10.46), $\hat{\sigma}^2_{\beta} = 139.74$ (55.23), $\hat{\sigma}_{y_1\beta} = 23.95$ (9.50), $\hat{\sigma}^2_{\varepsilon_2} = 75.63$ (19.57), $\hat{\sigma}^2_{\varepsilon_3} = 103.71$ (16.15), $\hat{\sigma}^2_{\varepsilon_4} = 145.00$ (33.83), and $\hat{\sigma}^2_{\varepsilon_5} = 208.89$ (40.47). All other parameter estimates are non-significant.

Hence, while the predetermined model does not fit the data of these two examples as well as a quadratic growth model would have (i.e., the quadratic growth model results in perfect model fit since we are modeling the population covariance matrix and population means), we can nevertheless conclude that the predetermined ALT model provides a good description of the covariance and mean structures generated by the quadratic growth model of Equation 4.12. Note that, like in the numerical

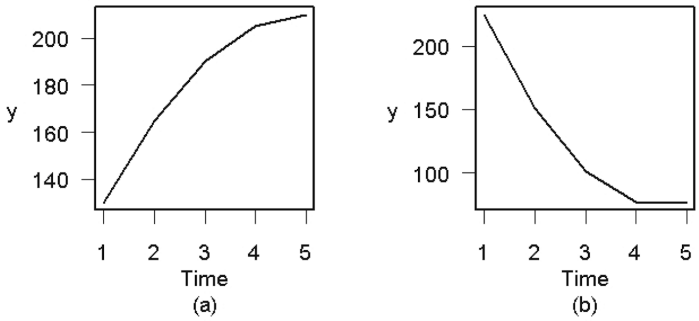


Fig. 4.5: Growth Curves Generated with a Quadratic Growth Curve Model. Panel (a) shows the actual (model predicted) growth trajectory for $\mu_\delta = 130$, $\mu_\gamma = 40$, $\mu_\xi = -5$, $\sigma^2_\delta = 240$, $\sigma^2_\gamma = 50$, $\sigma^2_\xi = 4.5$, $\sigma_{\delta\gamma} = .32$, $\sigma_{\delta\xi} = -3$, $\sigma_{\gamma\xi} = .83$, $\sigma^2_{\epsilon_1} = 45$, $\sigma^2_{\epsilon_2} = 48$, $\sigma^2_{\epsilon_3} = 55$, $\sigma^2_{\epsilon_4} = 70$, and $\sigma^2_{\epsilon_5} = 110$. Panel (b) shows the actual (model predicted) growth trajectory for $\mu_\delta = 225$, $\mu_\gamma = -86.4$, $\mu_\xi = 12.3$, $\sigma^2_\delta = 70$, $\sigma^2_\gamma = 170$, $\sigma^2_\xi = 2.7$, $\sigma_{\delta\gamma} = -.5$, $\sigma_{\delta\xi} = 3.5$, $\sigma_{\gamma\xi} = -.83$, $\sigma^2_{\epsilon_1} = 39$, $\sigma^2_{\epsilon_2} = 58$, $\sigma^2_{\epsilon_3} = 88$, $\sigma^2_{\epsilon_4} = 118$, and $\sigma^2_{\epsilon_5} = 138$.

examples in the previous section, the parameter estimates for the constant and change parameter do not represent the intercept and slope of the mean growth curves. In the first example, $\hat{\mu}_\beta = -18.28$. Based on this estimate, one might expect the mean trajectory for these data to be decreasing, but panel a of Figure 4.5 clearly shows the model predicted mean growth curve to be increasing. Furthermore, despite the fact that the mean change parameter $\hat{\mu}_\beta$ was estimated to be zero in the second example, panel b of Figure 4.5 shows that the mean growth curve for these data is not a horizontal line.

4.5 Empirical Illustration

To illustrate that researchers may not always be aware of the extreme flexibility of the predetermined ALT model, we discuss the application by Zyphur, Chaturverdi, and Arvey (2008), who analyzed a job performance data set found in Ployhart and Hakel (1998). This data set contained information on the job performance of 303 securities brokers, where performance was operationalized as the square root of gross sales commissions averaged across an annual quarter. In total, job performance was measured at eight different occasions. Zyphur et al. (2008) were interested in the effect of previous performance on future performance as well as in individual-specific

Table 4.2: Covariance matrix, means, and residual variances for $\mu_\delta = 225$, $\mu_\gamma = -86.4$, $\mu_\xi = 12.3$, $\sigma^2_\delta = 70$, $\sigma^2_\gamma = 170$, $\sigma^2_\xi = 2.7$, $\sigma_{\delta\gamma} = -.5$, $\sigma_{\delta\xi} = 3.5$, $\sigma_{\gamma\xi} = -.83$

	Y1	Y2	Y3	Y4	Y5
Y1	109.00				
Y2	73.00	305.04			
Y3	83.00	431.82	893.92		
Y4	100.00	627.34	1205.30	1117.24	
Y5	124.00	833.60	1629.96	2513.08	3620.96
Means	225	150.9	101.4	76.5	76.2
Residual Variances	39	58	88	118	138

performance trajectories, and therefore they analyzed the data using a predetermined ALT model with a constant and change parameter. The authors also compared the fit of their ALT model to that of several alternative models, including a quadratic model, but these models fitted the data less well. When reporting the results, Zyphur et al. (2008) interpreted the latent variables in their ALT model as a latent intercept and linear slope, and in accordance with this assumption presented linear performance trajectories (see Figure 2 in Zyphur et al. (2008)). However, the actual mean growth curve for this predetermined ALT model is as depicted in Figure 4.6. The deviation from linearity of this mean trajectory is not as large as in the numerical examples presented earlier, but it does show that a predetermined ALT model with a constant and a change parameter can result in a nonlinear trajectory with empirical data as well.

4.6 Discussion

This paper showed that the predetermined ALT model is very flexible and may lead to unexpected trajectories. Specifically, we presented several numerical examples that showed that the mean trajectory in this model can be quite different from what one may initially expect. For example, even with a negative estimates for the mean of the change parameter the model predicted growth curve can be increasing. In addition, we gave two examples that showed that a predetermined ALT model with a constant and change parameter can adequately fit covariance structures generated by models with divers growth processes (e.g., linear growth, quadratic growth). This shows that any discrepancies between the actual mean trajectory and the trajectory that the growth factors appear to indicate cannot necessarily be detected by a lack of model fit. Finally, we gave an empirical illustration that shows that discrepancies between the actual shape of the mean growth curve and the shape suggested by the latent variables can occur with real data as well.

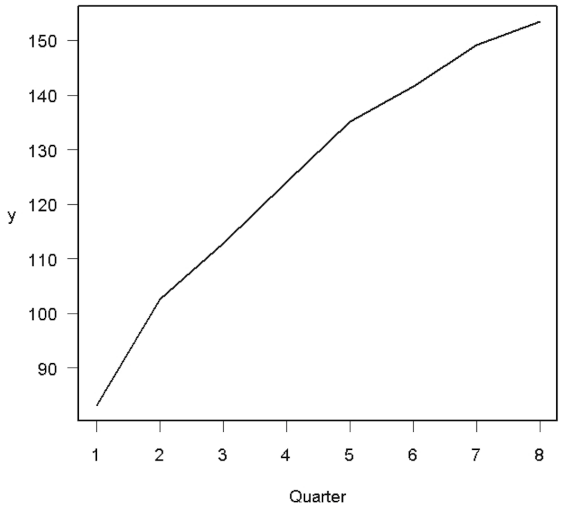


Fig. 4.6: Trajectory of model predicted scores from the article by Zyphur et al. (2008)

It is important to note that this flexibility of the mean and covariance structure is not a flaw of the predetermined ALT model. However, it is a model property that researchers using the model should be aware of. These researchers are therefore well advised to check their model predicted mean scores, either by entering the appropriate parameter values into the equations of the ALT model, or by checking a plot of the model predicted scores, in order to determine the shape of the trajectory that is being modeled. Merely eyeballing the parameter estimates of the constant and change parameter is not sufficient.

5

Bayesian Net Benefit Regression on Longitudinal Cost-Effectiveness Data from a Multi-Center RCT

Summary. In this paper, Bayesian multilevel net benefit regression methods are applied to longitudinal cost-effectiveness data from a multi-center randomized clinical trial. Specifically, it is tested whether the development of net benefit over time is best described by 1) autoregressive models, 2) latent growth curve models, 3) quadratic growth curve models, or 4) autoregressive latent trajectory models. We offer a coherent framework in which we outline a systematic strategy to find the best fitting model, that can then be used to directly produce meaningful Cost-Effectiveness Acceptability Curves without first having to resort to additional procedures for dealing with missing data.

5.1 Introduction

Randomized clinical trials on interventions for complex health problems often require extensive follow-up time periods before their full impact is revealed. These follow-up time periods may exceed the time span for which it is still reasonable to rely on patient recall. Therefore, collecting longitudinal data through patient report (e.g. interviews or questionnaires) requires that assessments are performed repeatedly over time. An

This chapter is being finished as Wetzelaer, P., Jongerling, J., Arntz, A., & Evers, S. (2015). Bayesian Net Benefit Regression on Longitudinal Cost-Effectiveness Data from a Multi-Center RCT.

P. Wetzelaer came up with the idea for this study and was responsible for the Introduction and the Discussion sections.

J. Jongerling wrote the computer code for the analyses, ran the analyses, and wrote the Methods section, the Result section, and the Appendices. He also contributed to the Introduction and the Discussion sections.

A. Arntz and S. Evers provided general input on the study and its topics, and made the data available (together with P. Wetzelaer)

advantage of this repeated assessment is that it can show (individual differences in) participant's development over time. This can provide important insight into the dynamics of the health problems under study, that would otherwise remain obscured. However, repeated assessment may also cause a specific methodological challenge. It inherently creates opportunities for missing data to occur. Whenever a trial participant misses a planned assessment, this will create a missing value in the dataset, and this loss of information subsequently adds uncertainty to the results that should not be ignored (otherwise results might be (seriously) biased). Unfortunately, most approaches for dealing with missing data are quite complex and require much time and effort. A notable exception to this rule is multilevel modeling.

Multilevel modeling was developed to analyze *hierarchical data* in which observations are "nested" within overarching units. Examples of such nested data are repeated measurements, in which the consecutive measures are nested within individuals, or clinical trials, in which participants are nested in treatment centers (and/or countries). Multilevel analysis efficiently uses all available data when estimating parameters. For example, assume that we measure a sample of 10 individuals on 5 different measurement occasions, and that one of these individuals has missing values at the third and fifth measurement occasion. The data for the other 9 individuals are complete. With these data, multilevel analysis will estimate the mean score at time points 1, 2, and 4, using all 10 individuals in the sample. To estimate the mean scores at time points 3 and 5, multilevel analysis will use the observed scores of the 9 people with complete data, and the *expected* scores at these time points for the person with the missing data, where these expected scores are determined based on his/her observed scores at time points 1, 2, and 4. This way, all information provided by the individuals in the sample is used, while the fact that there is missing data will show up in larger standard errors for the estimates at time points 3 and 5 (because the estimates for those time points are based on fewer *observed* scores than the estimates at time points 1, 2, and 4). This way of handling missing data implies that multilevel analysis will easily produce unbiased results (under certain assumptions) when missing data is present.

The above implies that multilevel analysis is ideally suited for the analysis of longitudinal data from multi-center randomized clinical trials (RCTs); it has the ability to model hierarchical data, and can easily deal with the missing data that is likely to occur with this type of data. In this study, multilevel regression models are therefore described for longitudinal *economic evaluations* from multi-center RCTs. These economic evaluations are usually performed in parallel to RCTs, and provide evidence on the value-for-money of health care interventions. Specifically, patient-level data on health outcomes (e.g. percentage of recovery) are combined with economic data on resource use in what is called a cost-effectiveness analysis (CEA). A convenient framework for these CEA's is net benefit regression (NBR) (Hoch, Briggs, & Willan, 2002). In this framework, the net benefit (NB) is calculated for each trial participant, by subtracting the costs made by this individual from the amount that we are willing to pay for his/her recovery, and subsequently used as the outcome variable in an regression analysis. The advantage of this approach is that the relative benefit

of different treatments can be determined by also adding the type of treatment as a predictor in the regression model.

Ultimately, the goal of these CEA's is to give probabilistic statements about which intervention is the most cost-effective. However, with the standard (frequentist) analysis methods available in popular statistical software packages such as SPSS, STATA, and SAS, this is not possible. The reason for this is that hypothesis testing, the method of inference used in classical statistics, always works under the assumption that a given null-hypothesis is true. This means that it can only determine how likely the observed sample data are, in a situation where an intervention of interest is assumed not to be cost-effective. What researcher usually want however is the opposite. They want to know how likely it is that a given treatment is cost-effective given the sample data at hand. These two probabilities are not the same, and importantly, one can not be determined from the other. A solution to this problem is provided by Bayesian statistics (discussed in more detail in the *Bayesian Estimation of the Multilevel Models* section of Appendix A). Bayesian statistics is a subset of the field of statistics that has a different approach to probability and statistical inference than classical statistics. Due to these differences, the outcome of a Bayesian analysis does not consist of parameter-estimates and corresponding standard errors, but of entire probability distributions for the parameters under study. These distributions, called *posterior distributions*, give the likelihood for different parameter-values *given the data*, and can be used for statistical inference. The mean of the posterior distribution of a parameter can be used as a point estimate for example, while the standard deviation can be used as a measure equivalent to standard errors. Moreover, the likelihood of certain ranges of parameter-values (e.g., values larger than 0) can be estimated by determining the proportion of the posterior distributions that fall within this range. So, in the context of a CEA, a Bayesian estimate of the average NB for a given treatment is determined by using the mean of the posterior distribution of a treatments' average NB. In addition, the probability of relative cost-effectiveness between treatments (i.e., the probability that one treatments is more cost-effective than the other) is obtained by first estimating the posterior distribution for the difference in average NB between treatments, and subsequently determining the proportion of this posterior that lies above 0.

The rest of this article is organized as follows. First, we will give a more detailed description of NBR. We'll describe how individual's NB scores can be calculated in longitudinal studies, and we'll give short descriptions of the NBR models used in this paper. These description are deliberately kept short to keep the readability as high as possible. More detailed (statistical) descriptions are given in the appendix to this paper, along with a short introduction to how these models can be estimated with Bayesian analysis. Next, we will discuss how the best fitting model can be selected from among this set of models provided, and we'll describe how the output of these models will be presented. In the second section, we present an empirical example concerning data from a multi-center RCT on psychotherapy for personality disorders (PDs). This example is used to illustrate how the approach outlined in this study can be applied in real life. We end with a discussion and conclusion.

5.2 Methods

5.2.1 Net benefit regression

In longitudinal studies, the net benefit (NB) can be determined for $i = 1, ..., N$ individuals (where N denotes the total sample size), at $t = 1, ..., T$ time points (where T is the total number of repeated measurements) using,

$$NB_{it} = \left(\frac{(\lambda * E_i)}{T} \right) - C_{it}, \tag{5.1}$$

where E_i is a dichotomous variable indicating effectiveness of the treatment for patient i (with $E_i = 0$ indicating no recovery at follow-up, and $E_i = 1$ indicating recovery at follow-up), λ is the amount of money (in euros) we are willing to spend on a participants recovery (referred to as the willingness-to-pay in CEA literature), and C_{it} are the treatment costs for patient i at assessment t. Note that by dividing by T in Equation 5.1, we are "spreading out" the NB of an individual across all time points equally. This NB is subsequently used as the dependent variable in NBR models.

In these models, conclusions about the relative effectiveness of treatments are based on differences in their average NB scores (across all individuals), such that a positive difference in average NB for one treatment over another indicates relative cost-effectiveness for that treatment. Here, the average NB of a treatment, across all individuals receiving that treatment, is simply determined by summing the model predicted (average) NB at the different measurement occasions (see the *QGC Model with Group as Predictor for the Random Parameters* section of Chapter 5 for a further explanation). Using Bayesian statistics (discussed in more detail in Appendix A), estimates of the average NB scores for the different treatments are determined using posterior distributions. As mentioned above, we'll use the mean of the posterior distribution of a treatments' average NB as our point estimate. In addition, the probability of relative cost-effectiveness between treatments (i.e., the probability that one treatments is more cost-effective than the other) is obtained by first estimating the posterior distribution for the difference in average NB between treatments, where this difference in average NB is simply obtained by subtracting the average NB of the first treatment from that of the second, and subsequently determining the proportion of this posterior that lies above 0. Both the presence and probability of relative cost-effectiveness can be calculated for different amounts of willingness-to-pay λ.

Extensive instructions and computer code for Bayesian NBR analysis and the estimation of treatments' average NB and relative cost-effectiveness are available in Chapter 5.

5.2.2 Model Specification

In this paper we use Bayesian estimation to test whether the development of NB over time is best described by 1) a first-order autoregressive (AR1) model (Jöreskog, 1971, 1979), 2) a latent growth curve (LGC) model (Meredith & Tisak, 1990; Bollen &

Curran, 2004, 2006), 3) a quadratic growth curve (QGC) model, or 4) an autoregressive latent trajectory (ALT) model (Curran & Bollen, 2001;Bollen & Curran, 2004). These longitudinal multilevel models are well suited for modeling the development of NB over time. Not only do they allow for the estimation of individual change trajectories (or growth curves) for NB over time, but for the estimation of inter-individual differences in these trajectories as well.

To facilitate the appreciation of the relevance of the suggested approach by applied researchers, the use of technical terms is kept to a minimum throughout the main article. A detailed statistical account of all the models is therefore reserved for an extensive appendix to this manuscript (Appendix A), while model code is given in Chapter 5 of this thesis. In short, the underlying rationale for each of the models is the following: AR-models assume that NB is stable over time, LGC models test for linear change in net benefit over time, QGC models test for curvi-linear change, and ALT models combine a LGC model with a first-order autoregressive (AR(1)) model, and can therefore test for linear change while accounting for the possibility of correlation between repeated measurements obtained from the same individual.

5.2.3 Model Comparison

The different models are compared based on their deviance information criterion (DIC) values to determine which model fits the data best (Spiegelhalter et al., 2002). An extensive description of this information citerion is again presented in the Appendix (Appendix B). For now, it suffices to know that, similar to other Information Criteria like the AIC (Akaike, 1973) and BIC (Schwarz, 1978), the DIC can be seen as consisting of two parts; one part that measures model misfit, and a second part that quantifies the dimensionality, or complexity of a model. This implies that model selection based on the DIC is based on a trade-off between model fit and model complexity. If two models fit the data equally well, then the model with the lowest complexity (i.e., the lowest number of model parameters) will be selected. Lower values on the DIC imply a 'better' model fit, and, as a rule of thumb, differences in DIC values larger than 5 are usually considered relevant. Once it is decided which model fits the data best, treatment condition is added as a predictor for the inter-individual variances in the intercept and the slope, in order to estimate separate growth curves for each condition.

5.2.4 Presentation of the results

The change in treatments' average NB is presented using growth curves, that is, curves that show a treatments' model predicted average NB at each of the measurement occasions. These curves will be constructed using our best fitting model, and will be presented for all treatments and for several different values of willingness-to-pay λ. Instructions on how these curves can be obtained are given in Appendix A (in the *Second Level Model* section) and in Chapter 5 of this thesis. In addition, the relative cost-effectiveness of one treatment over another will be given. This information

will be presented using cost-effectiveness acceptability curves (CEACs), that plot a treatments' probability of relative cost-effectiveness as a function of willingness-to-pay (λ). Specifically, the amount of money we are willing to spend on recovery is given on the x-axis of these CEACs, while the probability of relative cost-effectiveness is given on the y-axis. Computer code for the construction of these curves is again given in Chapter 5.

5.3 Empirical example

To demonstrate how to use the provided code and choose between the different models in practice, the suggested approach is illustrated with data from a multi-center RCT on psychotherapy for personality disorders (PDs). The study protocol has been described in detail elsewhere (Bamelis, Arntz, Wetzelaer, Verdooren, & Evers, 2013), but in short, this RCT included 320 patients with various PDs recruited from twelve mental health centers in the Netherlands, who were randomized over three treatment conditions: schema therapy (ST; n=145), clarification-oriented psychotherapy (COP; n=41) and treatment as usual (TAU; n=134). Descriptives for the data are given in Table 5.1. Overall, the dataset contained 22% missing data (from 0% at baseline to 32% at follow-up).

Patients were measured a total of six times. After an initial baseline assessments (and subsequent start of treatment), patients were assessed every six months for two years. A final follow-up assessment took place one years later. Since this 1 year interval between the final (fifth) measurement and the follow-up is twice as long as the interval between the first five measurements, we "inserted" an additional measurement occasion halfway between the last measurement and the follow-up when analyzing the data (resulting in a total of 7 measurement occasions). This inserted 6th measurement did not actually occur (and therefore contains only missing values) but was included to evenly space the measurements. Patients' recovery from PD at follow-up and their total costs at each measurement were determined in order to calculate patients' NB at each of the six measurements.

The models discussed in the section *Model Specification* were fitted to these data using the priors and data distributions given in Appendix A, and the R-code provided in Chapter 5. For the AR(1) model and the ALT model, we did not estimate models with random AR-parameters, because the number of repeated measures is to small for meaningful estimation of inter-individual differences in this parameter. For the AR(1) model, Chapter 5 provides code for data analysis with random AR-parameter. For the ALT model, no such code is provided because this model is usually not applied to data with enough repeated measures for random AR-parameters.

5.3.1 Results

The DIC values of all models that were tested are listed in Table 5.2. In depth descriptions of the models are available in Appendix A. In addition, the pD-values for

all the models, which are indicative of model complexity (higher values indicate more complexity), are given in this table. In case a model did not converge, this is also listed. These models are either too complex for the amount of information present in the data, or they they have a really bad fit to the data. A QGC model (presented in the *First Level Models* section of Appendix A) with random intercept, random slope, and fixed quadratic term had the lowest DIC value and was therefore the best fitting model (see Table 5.2). Subsequently, type of treatment was added as a predictor for the inter-individual variances in the intercept and the slope to enable estimation of separate growth curves for each treatment (see the *Second Level Model* section of Appendix A for instructions, and Chapter 5 for relevant computer code). These growth curves where estimated for λ values (in euros) of 0, 2500, 5000, 7500, 10000, 12500, 15000, 17500, 30000, and 37500, and are presented in Figure 5.1. This figure shows that, although the NB of the COP treatment seems to be structurally lower than that of the other two treatments, the average development of NB is quite similar for all three treatments. The parameter estimates and their corresponding credibility intervals (listed in Table 5.3) show a similar picture. The slope of the COP treatment is quite a bit lower than those of the TAU and ST treatments, while the intercepts of the treatments are comparable, but the credibility intervals of the parameters estimates of the different treatments show considerable overlap. However, since the credibility intervals are quite wide, the absence of substantial differences between conditions may be due to low statistical power.

Next, the probability of relative cost-effectiveness (i.e. the probability that one treatment has a higher NB than another) was determined for the ST treatment compared to the TAU and the COP treatments, and for the COP treatment compared to the TAU treatment (using the method described in the introduction and *Net Benefit Regression* section, and the code presented in Chapter 5). Like the growth curves, the probability of relative cost-effectiveness was determined for λ values of 0, 2500, 5000, 7500, 10000, 12500, 15000, 17500, 30000, and 37500, and the corresponding CEAC's are shown in Figure 5.2. The CEAC's show that the ST treatment is always likely to be more cost-effective than the COP treatment, while it is also likely to be more cost-effective than the TAU treatment (relative probability $> 70\%$) for willingness-to-pay values of 7500 euros or more. The COP treatment is always likely to be less cost-effective (relative probability between 20% and 30%) than the TAU treatment. Based on these results, the ST treatment would be recommended over the other two.

5.4 Discussion

Bayesian multilevel NBR for longitudinal data is a useful extension to the NBR framework, that holds much promise for applied researchers in the field of health economics. Following a systematic strategy a best fitting model can be determined, that is then used to directly produce growth curves and meaningful CEAC's without first having to resort to additional procedures for dealing with missing data.

It should be noted that, for now, we limited our analyses to normally distributed NB data with just two-levels (observations nested in individuals). In CEA's in the context of longitudinal RCT, both data with more than two levels, and data with non-normal distributions can occur. In RCT's people are usually assessed in several participating centers for example, leading to three-level data in which observations are nested in participants, and participants are nested in centers. In addition, NB is a function of both an effect variable and a cost variable, each of which has its own parametric characteristics. Because of this, the actual distribution of the NB scores is likely to be skewed. Fortunately, extending our approach to nested data with three or more levels and/or non-normally distributed data is straightforward, and information on how to do so is provided in Appendix C and in Chapter 5.

Table 5.1: Descriptives

	ST		**COP**		**TAU**	
Gender	**n**	**%**	**n**	**%**	**n**	**%**
Males	66	45.50	18	43.90	55	41.00
Females	79	54.50	23	56.10	79	59.00
Total	145	100.00	41	100.00	134	100.00
Measurement Occasions	**Mean Costs**	**%Missing**	**Mean Costs**	**%Missing**	**Mean Costs**	**%Missing**
Occasion 1	4615.21	0.00	4759.22	0.00	5657.13	0.00
Occasion 2	5620.31	9.66	4767.32	14.60	4350.48	23.13
Occasion 3	5005.10	20.69	4687.37	14.63	3657.76	29.10
Occasion 4	3526.72	24.83	4843.15	19.51	3238.50	34.33
Occasion 5	2883.03	26.90	3997.76	17.07	3513.05	38.06
Occasion 6	-	100.00	-	100.00	-	100.00
Occasion 7	1992.30	29.66	2748.82	19.51	3179.91	38.81

Descriptive statistics for the data from the study by Bamelis, Arntz, Wetzelaer, Verdooren, and Evers (2013). For the 6th measurement occasion no data is available, because it did not actually occur. Patients were assessed every six months for the first 5 measurements, but the final follow-up took place 1 year after the the fifth measurement. This 1 year interval is twice as long as the interval between the first five measurements so we inserted an additional measurement halfway between the last measurement and the follow-up when analyzing the data to evenly space the measurements. This results in a total of 7 measurements ($T = 1, \ldots, 7$).

Table 5.2: DIC

Model Family	Variant	DIC	pD
AR	Fixed Mean	29767.90	61.30
	Random Mean	Did not Converge	
LGC	Fixed Intercept & Slope	30152.20	3.00
	Random Intercept, Fixed Slope	29944.30	497.20
	Fixed Intercept, Random Slope	30335.00	302.80
	Random Intercept & Slope	29737.70	984.40
QGC	Fixed Intercept, Slope, and Quadratic Term	30154.20	4.00
	Random Intercept, Fixed Slope and Quadratic Term	29959.40	511.20
	Random Slope, Fixed Intercept and Quadratic Term	30348.10	313.30
	Random Quadratic Term, Fixed Intercept and Slope	Did not Converge	
	Random Intercept and Slope, Fixed Quadratic Term	29728.00	972.90
	Random Intercept and Quadratic Term, Fixed Slope	30196.30	990.30
	Random Slope and Quadratic Term, Fixed Intercept	30589.20	588.40
	Random Intercept, Slope, and Quadratic Term	Did not Converge	

Continued on next page

Table 5.2 – continued from previous page

Model Family	Variant	DIC	pD
ALT (constrained)	Fixed Alpha, Beta, and AR	29738.80	56.50
	Random Alpha, Fixed Beta and AR	Did not Converge	
	Random Beta, Fixed Alpha and AR	30369.50	973.30
	Random Alpha and Beta, Fixed AR	Did not Converge	
ALT (LGC with AR(1) Errors)	Fixed Intercept, Slope, and AR	29739.30	56.80
	Random Intercept, Fixed Slope and AR	Did not Converge	
	Random Slope, Fixed Intercept and AR	Did not Converge	
	Random Intercept and Slope, Fixed AR	Did not Converge	

The DIC values and pD values for the different Bayesian multilevel models presented in this paper. Lower values for the DIC indicate better model fit. Lower values for pD indicate lower model complexity.

Table 5.3: Parameter Estimates for the QGC model with Random Intercept and Slope, Fixed Quadratic term, and Treatment as a predictor for inter-individual variability

		Parameter Estimates			95% Credibility Interval		
Lambda	Group	Intercept	Slope	Quadratic Term	Intercept	Slope	Quadratic Term
0	TAU	-5054.85	385.40	4.55	-6194.69 - -3932.09	-19.10 - 787.93	-42.03 - 53.56
0	ST	-5329.62	530.08	4.55	-7976.38 - -2689.58	-281.04 - 1328.92	-42.03 - 53.56
0	COP	-5056.56	268.84	4.55	-8381.39 - -1669.64	-724.48 - 1245.76	-42.03 - 53.56
2500	TAU	-4794.92	397.20	5.14	-5883.37 - - 3694.39	7.83 - 800.26	-41.66 - 52.93
2500	ST	-5019.95	533.23	5.14	-7633.61 - -2420.82	-251.05 - 1344.75	-41.66 - 52.93
2500	COP	-4843.91	275.55	5.14	-8178.35 - -1492.57	-707.35 - 1247.03	-41.66 - 52.93
5000	TAU	-4573.07	423.11	4.08	-5678.40 - -3416.32	16.84 - 824.96	-43.55 - 52.61
5000	ST	-4742.44	551.01	4.08	-7404.85 - -2050.64	-263.35 - 1350.04	-43.55 - 52.61
5000	COP	-4606.87	286.22	4.08	-7984.63 - -1206.82	-712.63 - 1284.34	-43.55 - 52.61
7500	TAU	-4315.34	432.84	4.06	-5417.44 - -3183.88	45.15 - 826.23	-42.65 - 53.36
7500	ST	-4439.69	558.64	4.06	-7028.16 - -1780.44	-22.48 - 1361.72	-42.65 - 53.36
7500	COP	-4376.14	292.08	4.06	-7738.85 - -972.21	-681.90 - 1247.90	-42.65 - 53.36
10000	TAU	-4056.92	446.74	3.44	-5153.76 - -2938.37	51.92 - 841.89	-45.48 - 51.21
10000	ST	-4142.90	567.56	3.44	-6783.01 - -1532.48	-223.30 - 1357.37	-45.48 - 51.21
10000	COP	-4118.67	307.27	3.44	-7456.44 - -795.72	-659.61 - 1281.23	-45.48 - 51.21
12500	TAU	-3820.85	463.42	3.73	-5000.81 - -2676.27	55.74 - 864.52	-42.77 - 51.46
12500	ST	-3854.83	576.56	3.73	-6590.78 - -1194.34	-216.89 - 1383.69	-42.77 - 51.46
12500	COP	-3915.13	314.23	3.73	-7312.58 - -476.91	-678.56 - 1301.56	-42.77 - 51.46
15000	TAU	-3570.82	477.74	3.51	-4704.79 - -2473.86	80.40 - 868.02	-43.64 - 49.61
15000	ST	-3570.82	587.02	3.51	-6256.30 - -908.65	-218.56 - 1369.07	-43.64 - 49.61
15000	COP	-3702.26	324.45	3.51	-7178.54 - -369.96	-653.31 - 1286.63	-43.64 - 49.61
17500	TAU	-3331.68	493.11	2.66	-4461.26 - -2190.55	92.12 - 895.50	-45.76 - 52.39
17500	ST	-3262.72	598.81	2.66	-5977.80 - -574.23	-190.78 - 1400.65	-45.76 - 52.39
17500	COP	-3432.81	325.53	2.66	-6825.27 - 27.44	-695.13 - 1301.52	-45.76 - 52.39
30000	TAU	-2083.47	539.39	2.17	-3274.63 - -883.20	148.66 - 942.53	-45.32 - 48.69
30000	ST	-1800.27	637.18	2.17	-4685.45 - 1047.31	-170.55 - 1445.21	-45.32 - 48.69
30000	COP	-2246.44	348.72	2.17	-5811.32 - 1443.04	-621.65 - 1324.90	-45.32 - 48.69
37500	TAU	-1351.97	565.48	0.72	-2603.62 - -95.82	171.92 - 974.64	-49.76 - 48.51
37500	ST	-917.74	659.44	0.72	-3833.65 - 2011.63	-129.29 - 1469.21	-49.76 - 48.51
37500	COP	-1520.92	361.37	0.72	-5345.05 - 2322.34	-610.03 - 1371.38	-49.76 - 48.51

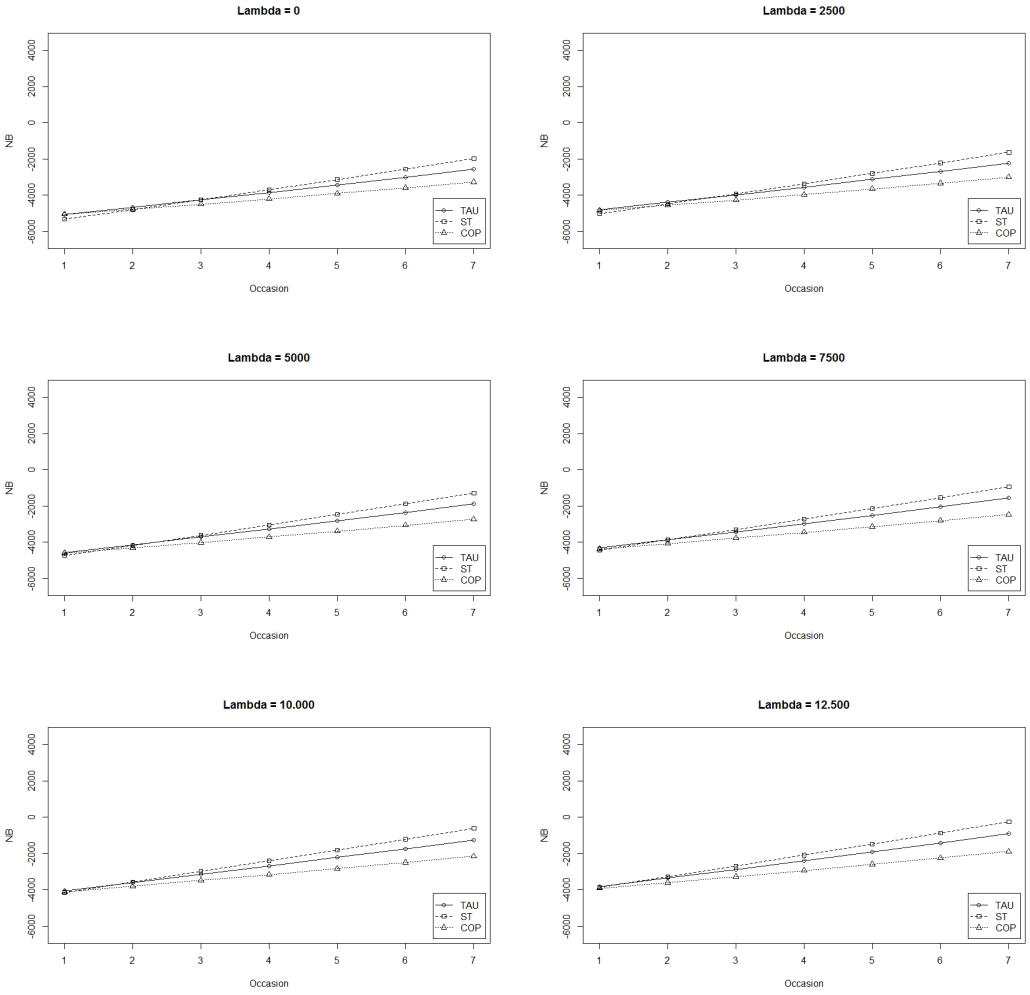


Fig. 5.1: Growth Curves

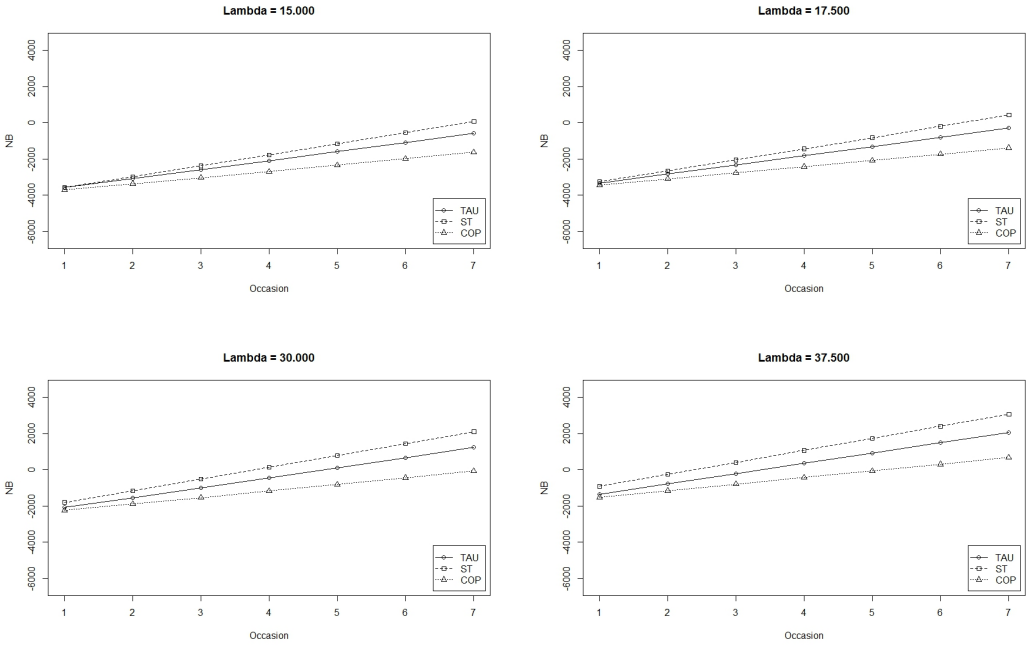


Fig. 5.1: Growth Curves (Continued from previous page)

5.5 Appendix A: Bayesian Multilevel Models

5.5.1 First Level Models

The First-Order Autoregressive Model

The first order autoregressive (AR(1)) model (Hamilton, 1994; Chatfield, 2003) describes a situation where there is no systematic change (no increase or decrease) over time, and where individual's scores simply fluctuates around an individual mean. An AR(1) process can be expressed as consisting of two parts: a mean score μ, which represents an individual's trait score (i.e., his/her long-run tendency, equilibrium, or long-term preferred state) and a temporal deviation from this mean, which we denote as ζ_t, that is

$$y_t = \mu + \zeta_t. \tag{5.2}$$

The temporal deviations (or states) themselves also may be characterized by autocorrelation and can be modeled with the AR(1) model

$$\zeta_t = \phi\zeta_{t-1} + \varepsilon_t. \tag{5.3}$$

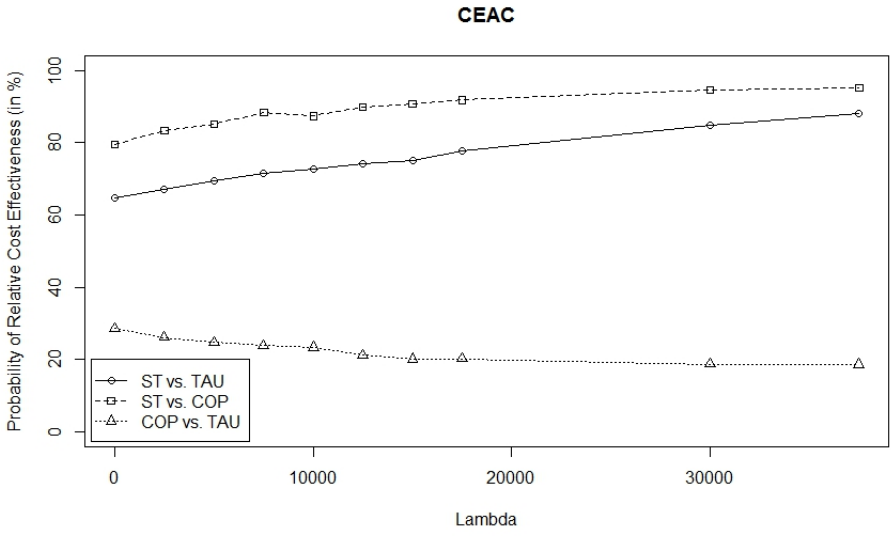


Fig. 5.2: Total Relative Probability of Cost-Effectiveness

The ST vs TAU curve shows the probability that the ST treatment is more cost-effective than the TAU treatment, the ST vs COP curve shows the probability that the ST treatment is more cost-effective than the COP treatment, and the COP vs TAU curve shows the probability that the COP treatment is more cost-effective than the TAU treatment.

Where ϕ is the AR-parameter used to regress the current state on the previous one, and ε_t is the unpredictable part, also referred to as the innovation, residual, or random shock. It is assumed that ϕ lies between -1 and 1 to ensure stationarity (that is, a situation in which the mean and variance of the process do not change over time, see Hamilton, 1994, p. 53 and Chatfield, 2003, pg. 12). Furthermore, it is assumed that the innovations are independent and normally distributed with a mean equal to 0 and variance σ^2_ε.

The first order autoregressive (AR(1)) model as introduced by Hamilton (1994) and Chatfield (2003) can be extended to a multilevel model by allowing individuals to differ with respect to the model parameters μ and ϕ. This results in,

$$y_{it} = \mu_i + \zeta_{it}. \tag{5.4}$$

Where the temporal deviations can again be written as,

$$\zeta_{it} = \phi_i \zeta_{t-1} + \varepsilon_{it}, \tag{5.5}$$

with the model parameters being subject to the same asumption as mentioned above. Note that this multilevel extension of the AR(1) model is closely related to the simplex

model as introduced by Jöreskog (1971, 1979). However, whereas the simplex model assumes that the model parameters are fixed across individuals (i.e., $\mu_i = \mu, \forall i$, and $\phi_i = \phi, \forall i$), the multilevel AR(1) model does not. It is important to realize however, that reliably estimating (inter-individual differences in) individual AR-parameters likely requires around 20, and preferably 50, repeated measures for each participant (Jongerling et al., 2015). If fewer repeated measures are available, as is the case in the current study, a multilevel AR(1) model in which only the mean is random might be preferred, in which case the amount of autocorrelation should be fixed across individuals (i.e., $\phi_i = \phi$ in Equation 5.5). This is exactly what was done for the analyses in the current study. In addition, we also assume that the amount of autocorrelation is constant over time in this study, even though the AR-parameter could theoretically be allowed to vary over time (e.g., the amount of autocorrelation between time points 1 and 2 could me modeled as different from the amount of autocorrelation between time points 2 and 3).

The inertia parameter ϕ_i in Equation 5.5 reflects the degree to which previous scores or states carry over into current scores or states. Suppose we have a number of repeated measurements of NB for an individual. If the inertia parameter is close to zero, this implies that there is little or no carry over from the level of NB on the previous measurement occasion to the current level of NB. In contrast, when the inertia parameter is close to 1, this implies that an increased amount of NB at the previous measurement is likely to persist into the current measurement (and subsequent measurements), while decreased levels also tend to persist into subsequent measurements. This is where the interpretation of inertia comes from.

The innovation ϵ_{it} represents the part of the process that cannot be predicted based on previous scores or states. Thus, it can be thought of as the collection of all unobserved (or omitted) factors that influence the process under investigation.

In the AR(1) model, the growth curve for the average NB (i.e., the curve that shows the model predicted average NB at each of the measurement occasions) is simply given by

$$\mu_t = \mu \qquad \text{for} \quad t = 1, \ldots, T, \tag{5.6}$$

where T is the total number of time points, and μ is the average NB across all individuals.

In Chapter 5, R-code for this model is described in Section 4.2.1, while model files for the Bayesian analysis with the JAGS-program (Plummer, 2003, 2012) is given in Sections 10.0 through 10.4.

Latent Growth Model

The LGC model (Meredith & Tisak, 1990) is used to model linear change over time, and can be written as follows,

$$y_{it} = \alpha_i + \beta_i(t-1) + \varepsilon_{it}, \tag{5.7}$$

where α_i is an individual's intercept that represents his/her score at time point $t = 0$, β_i is the (linear) slope parameter which quantifies the amount of change between successive measurements, and ε_{it} is a random error term with zero mean and variance σ^2_ε. The residuals, ε_{it}, are assumed uncorrelated across individuals and time, and uncorrelated with the random intercept and random slope.

Note, that the slope parameter β_i is multiplied by the current measurement occasion minus 1 $(t - 1)$. This is done so the factor loading of the first measurement occasion is equal to 0, making the first measurement our reference point, and an individual's intercept (α_i) equal to his or her expected NB score at this first measurement occasion. A positive value for slope parameter β_i subsequently means that the scores get higher from each measurement occasion to the next, while a negative value implies that the NB decreases over time.

For this model, the growth curve for the average NB is,

$$\mu_t = \mu_\alpha + (t-1)\mu_\beta \text{ for } t = 1, \ldots, T. \tag{5.8}$$

where μ_α and μ_β represent the mean intercept and mean slope across individuals respectively. Note that this equation shows that multiplying the slope by $t - 1$ indeed makes the first measurement occasion equal to the intercept. After all, filling in Equation 5.8 for $t = 1$ results in,

$$\mu_1 = \mu_\alpha + 0 * \mu_\beta, \tag{5.9}$$

$$\mu_1 = \mu_\alpha. \tag{5.10}$$

In Chapter 5, R-code for this model is described in Section 4.2.2, while model files for the Bayesian analysis with the JAGS-program (Plummer, 2003, 2012) is given in Sections 10.5 through 10.8.

Quadratic Growth Model

The quadratic growth curve (QGC) model is very similar to the LGC model. The only difference is that the QGC model contains one additional parameter, the quadratic term (η_i), that allows for the modeling of quadratic growth over time. As a result, this type of model is better suited for modeling change that increases or decreases in rate over time. For example, if the increase (or decrease) in someone's NB is large at first, but then levels off, the QGC model can describe this type of change accurately, while the LGC, which assumes that the rate of change is the same across all measurement occasions can not. The QGC model can even model change that is increasing at first and decreasing later on (or vice versa).

The QGC model can be written as follows,

$$y_{it} = \alpha_i + \beta_i(t-1) + \eta_i(t-1)^2 + \varepsilon_{it}, \tag{5.11}$$

where α_i and β_i are again the individual intercept and linear slope, η_i is an individual's quadratic term, and ε_{it} is again a random error term with zero mean and variance

σ_ε^2, that is assumed to be uncorrelated across individuals and time, and uncorrelated with the other model parameters.

Note, that the slope parameter β_i is again multiplied by the current measurement occasion minus 1 $(t-1)$, and that the quadratic term is multiplied by $(t-1)^2$. Like with the LGC model, this is done to make the first measurement our reference point, and an individual's intercept (α_i) equal to his or her expected NB at this first measurement occasion.

The growth curve for average NB of the QGC is equal to,

$$\mu_t = \mu_\alpha + (t-1)\mu_\beta + (t-1)^2\mu_\eta \text{ for } t = 1, \ldots, T. \tag{5.12}$$

where μ_α and μ_β are again the mean intercept and mean slope over individuals, while μ_η is the mean quadratic effect.

In Chapter 5, R-code for this model is described in Section 4.2.3, while model files for the Bayesian analysis with the JAGS-program (Plummer, 2003, 2012) is given in Sections 10.9 through 10.16.

ALT model

The ALT model (Curran & Bollen, 2001;Bollen & Curran, 2004) is a combination of a LGC and an AR(1) model, and can be described as a LGC model with AR relations between the observed variables (Bollen & Curran, 2004). We included this model in the study because systematic change in NB over time is likely, but we also think that an individual's current amount of NB will depend on his/her previous amount. While the ALT model can model both these processes simultaneously, having both systematic change and autocorrelation between measurements introduces a difficulty.

This can be seen by looking at the expression for the ALT model, which can be written as,

$$y_{it} = \alpha_i + \beta_i(t-1) + \phi y_{i,t-1} + \varepsilon_{it}, \tag{5.13}$$

where α_i is a constant, β_i is a change parameter, ϕ is the AR parameter, and ε_{it} is a random error term that is subject to the same assumptions as under the AR(1) and LGC model. What Equation 5.13 shows is that (unless all $\phi = 0$) there is a recursion in the ALT model. That is, y_{it} is a function of α_i, β_i, and $y_{i,t-1}$, but this last term is a function of α_i, β_i and $y_{i,t-2}$, and so on. Due to this recursion, we cannot use the ALT model unless we find a satisfactory way to incorporate all previous (unobserved) observations into the model. If the AR-parameter is restricted to be the same across the entire range of measurement occasions and smaller than 1 in absolute value (i.e., $|\phi| < 1$), as it is in this study, then all previous measurement occasions can be incorporated into the ALT model by putting nonlinear constraints on α_i (i.e., $(1-\phi)^{-1}$) and β_i (i.e., $\phi(1-\phi)^{-2}$) at the first measurement occasion (Bollen & Curran, 2004) to get,

$$y_{i1} = \alpha_i(1-\phi)^{-1} - \beta_i\phi(1-\phi)^{-2} + z_{i1}, \text{ where} \tag{5.14}$$

$$\begin{aligned} z_{i1} &= \varepsilon_{i1} + \phi\varepsilon_{i0} + \phi^2\varepsilon_{i,-1} + \cdots + \\ &= \sum_{j=0}^{\infty} \phi^j \varepsilon_{i,1-j}, \end{aligned} \tag{5.15}$$

Assuming that the residual variation is equal across time, the term z_{i1} represents an infinite weighted sum of all the unobserved, previous residuals, which thus solves for the recursion in the ALT model. In addition, this term has a zero mean and variance σ_z^2 (which is a function of σ_ε^2 and ϕ)(Hamaker, 2005). After Equations 5.14 and 5.15 are used to model the data at the first measurement occasion, Equation 5.13 can be used for the subsequent measurement occasions ($t = 2, 3, ..., T$). Although these constraints for α_i and β_i at $t = 1$ solve the recursion problem they make the parameters of the ALT model hard to interpret. Specifically, with these constraints, the intercept and slope of the ALT model are equal to (Hamaker, 2005),

$$\delta_i = \alpha_i(1-\phi)^{-1} - \beta_i\phi(1-\phi)^{-2}, \text{ and} \tag{5.16a}$$

$$\gamma_i = \beta_i(1-\phi)^{-1}, \tag{5.16b}$$

and not simply to α_i and β_i. For this variation of the ALT model, the growth curve for the average NB can be written as,

$$\begin{aligned} \mu_t &= \mu_\delta + (t-1)\mu_\gamma \\ &= \mu_\alpha(1-\phi)^{-1} - \mu_\beta\phi(1-\phi)^{-2} + (t-1)\mu_\beta(1-\phi)^{-1} \quad \text{for } t = 1, \ldots, T. \end{aligned} \tag{5.17}$$

Because of this difficulty with the interpretation of the parameters of the ALT model after using the constraints, we also consider another solution to the recursion problem.

This second solution involves rewriting the ALT model as a LGC model with autocorrelated disturbances, such as proposed by Chi and Reinsel (1989). This can be done because the ALT model and a LGC model with autocorrelated errors are algebraically equivalent when the AR parameter is invariant over time (Hamaker, 2005), as it is in this study. The advantage of this second solution for the recursion problem, is that it separates the LGC, or trend, part of the model from the AR part, which keeps the interpretation of the model parameters straightforward. Specifically, a LGC model with autocorrelated errors can be written as,

$$y_{it} = \alpha_i + \beta_i(t-1) + \phi(y_{i,t-1} - (\alpha_i + \beta_i(t-2))) + \varepsilon_{it}. \tag{5.18}$$

where α_i and β_i are again the individual intercept and linear slope, ϕ is the autoregressive parameter which is subject to the same constraints as under the AR(1) model, and ε_{it} is again a random error. Note that β_i is again multiplied by $(t-1)$ to make

the first measurement occasion our reference point and an individual's intercept (α_i) equal to his/her expected NB at this first measurement occasion. For $t = 1$, the LGC model with autocorrelated errors can be written as,

$$y_{i1} = \alpha_i + \varepsilon_{i1}, \tag{5.19}$$

while for the subsequent occasions the model is,

$$y_{it} = \alpha_i + \beta_i(t-1) + \phi(y_{i,t-1} - (\alpha_i + \beta_i(t-2))) + \varepsilon_{it}. \tag{5.20}$$

Both error terms in these equations, ε_{i1} and ε_{it}, have a zero mean while the variance of ε_{i1}, $\sigma^2_{\varepsilon_{i1}}$, is equal to,

$$\sigma^2_{\varepsilon_{i1}} = \frac{\sigma^2_{\varepsilon_i}}{1-\phi^2}, \tag{5.21}$$

where $\sigma^2_{\varepsilon_i}$ is the variance of ε_{it}.

When writing the ALT model as an LGC model with autocorrelated disturbances, the intercept and the slope of the model are simply equal to α_i and β_i, while the parameter ϕ quantifies the amount of autocorrelation between successive measurement errors. The growth curve of average NB for this second variation is equal to,

$$\mu_t = \mu_\alpha + (t-1)\mu_\beta + \phi(\mu_{t-1} - (\mu_\alpha + (t-2)\mu_\beta)) \text{ for } t = 1, \ldots, T. \tag{5.22}$$

In Chapter 5, R-code for this model is described in Section 4.2.4, while modelfiles for the Bayesian analysis with the JAGS-program (Plummer, 2003, 2012) is given in Sections 10.17 through 10.24.

5.5.2 Second Level Model

On the second level inter-individual differences in the parameters of the AR(1), LGC, QGC, and ALT model are modeled (Jöreskog, 1971; Jöreskog, 1971; Meredith & Tisak, 1990; Bollen & Curran, 2004; Curran & Bollen, 2001). In this study we assume that all model parameters are normally distributed, making the second level expressions for models with only one random parameter equal to,

$$\mu_i \sim \mathcal{N}(\mu_\mu, \sigma^2_\mu), \tag{5.23}$$
$$\alpha_i \sim \mathcal{N}(\mu_\alpha, \sigma^2_\alpha), \tag{5.24}$$
$$\beta_i \sim \mathcal{N}(\mu_\beta, \sigma^2_\beta), \quad \text{and} \tag{5.25}$$
$$\eta_i \sim \mathcal{N}(\mu_\eta, \sigma^2_\eta). \tag{5.26}$$

When more than one parameter is random, the inter-individual differences in these parameters might be related. To account for this, the random parameter must be

allowed to correlate with each other by giving them a joint distribution. Since we assume that our parameters are normally distributed, we will model the inter-individual differences in models with more than one random parameter using a multivariate normal (MVN) distribution. For a model with three random parameters, θ_1, θ_2, and θ_3, the MVN distribution can be written as,

$$\begin{bmatrix}\theta_{1i}\\ \theta_{2i}\\ \theta_{3i}\end{bmatrix} \sim MVN\left(\begin{bmatrix}\mu_{\theta_1}\\ \mu_{\theta_2}\\ \mu_{\theta_3}\end{bmatrix}, \begin{bmatrix}\tau^2_{\theta_1} & & \\ \tau_{\theta_1\theta_2} & \tau^2_{\theta_2} & \\ \tau_{\theta_1\theta_3} & \tau_{\theta_2\theta_3} & \tau^2_{\theta_3}\end{bmatrix}\right), \tag{5.27}$$

where $\tau^2_{\theta_1}$, $\tau^2_{\theta_2}$, and $\tau^2_{\theta_3}$ are the inter-individual variances in the model parameters; $\tau_{\theta_1\theta_2}$, $\tau_{\theta_1\theta_3}$, and $\tau_{\theta_2\theta_3}$ are the covariances between the random parameters; and μ_{θ_1}, μ_{θ_2}, and μ_{θ_3} are the expected values of θ_{1i}, θ_{2i}, and θ_{3i} respectively. For a LGC model (Meredith & Tisak, 1990), with a random intercept α_i and a random slope β_i, the level two model would be,

$$\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix} \sim MVN\left(\begin{bmatrix}\mu_\alpha\\ \mu_\beta\end{bmatrix}, \begin{bmatrix}\tau^2_\alpha & \\ \tau_{\alpha\beta} & \tau^2_\beta\end{bmatrix}\right). \tag{5.28}$$

When we want to add predictors on the second level of our models this is simply done by writing the means of the normal distributions in Equations 5.23 through 5.26 as linear equations containing these predictors. For example, assume patients receive three different treatments (like they did in this study) and we want to add treatment type as a predictor for inter-individual differences in the slope parameter of a LGC model β. This merely involves substituting the mean slope parameter μ_β from Equation 5.25 with,

$$\mu_{\beta i} = \gamma_{00} + \gamma_{10} * D1_i + \gamma_{20} * D2_i, \tag{5.29}$$

where $D1$ and $D2$ are two dummy variables used to identify which treatment individual i received, γ_{00} is the mean slope for the reference group, γ_{10} is the regression coefficient for *D1* which quantifies the difference in mean slope between the reference group and the group belonging to a score of 1 on *D1*, and γ_{20} is the regression coefficient for *D2* which quantifies the difference in mean slope between the reference group and the group belonging to a score of 1 on *D2*. For the other parameters in Equations 5.23 through 5.26 a similar approach can be used.

5.5.3 Bayesian Estimation of the Multilevel Models

In this study we use the JAGS program (Plummer, 2003, 2012) for the Bayesian estimation of our multilevel models. JAGS is a free software package and can be used by itself or in combination with R (R Core Team, 2014) using the R2jags package (Yu-Sung & Masanao, 2014) to call JAGS from R. The actual steps and code necessary

to use JAGS to analyze data with the multilevel models discussed above are provided in Chapter 5 of this thesis.

Here, we will focus a bit more on Bayesian estimation itself. In Bayesian estimation, several steps are required before a model can be estimated by a program (e.g., JAGS). While a thorough discussion of Bayesian statistics is beyond the scope of this article (interested readers are referred to Gelman et al. (2004), Hamaker and Klugkist (2011), and Hoijtink (2009)), there is one feature of Bayesian analysis that needs to be discussed here (albeit briefly) – the prior distribution. In Bayesian statistics, researchers need to specify prior distributions for all model parameters, where these prior distributions represent a researcher's prior beliefs or knowledge about these parameters by assigning probabilities to their different possible values. These prior distributions are then combined with the distribution of the data using Bayes theorem in the following way,

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{f(y)}, \tag{5.30}$$

where y represents the sample data, and θ represents a model parameter we want to estimate (e.g., the AR-parameter ϕ of the AR(1) model, or the slope parameter β of the LGC model). In addition, $f(\theta|y)$ is the posterior distribution of parameter θ that represents the combined information about this parameter from both the prior and the data, $f(y|\theta)$ is the distribution of the data (y) conditional on parameter θ, $f(\theta)$ is the prior distribution for parameter θ, and $f(y)$ is the marginal distribution of the data. Posterior distributions of parameters of interest are subsequently used for model estimation. That is, the mean, median, or mode of a posterior distribution can be used as the point estimate of a parameter, while the standard deviation of the posterior distribution can be seen as a measure of the sample variability of this estimate (analogues to the standard error in standard maximum likelihood estimation). If one has little or no prior knowledge, *uninformative* priors can be used, which are characterized by assigning low and (approximately) equal probabilities to a very large range of possible values of a parameter. The results obtained with such priors depend almost exclusively on the data, and are therefore often close to ML estimates.

For *fixed* parameters, that is, parameters that are not variance terms and that do not vary across individuals, normal distributions with 0 mean and (very) large variances are often used as uninformative priors, because the large variances spread the normal distribution out over a very large range of values, with each value in the range getting a very small and approximately equal probability. In this study, fixed effects will be assigned normal priors with 0 means and variances equal to 10^{12}, that is,

$$\pi \sim \mathcal{N}(0, 10^{12}), \tag{5.31}$$

where π represents the fixed parameter(s) of a model. For example, in the QGC model the priors for the mean intercept, mean linear change parameter, and mean quadratic change parameter are given by,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}), \tag{5.32}$$

$$\mu_\beta \sim \mathcal{N}(0, 10^{12}), \tag{5.33}$$

and,

$$\mu_\eta \sim \mathcal{N}(0, 10^{12}). \tag{5.34}$$

For uncorrelated variance terms (e.g., the innovation variance σ_ε^2 of a AR(1) model), the inverse gamma distribution is often used in Bayesian statistics, because this distribution does not allow values smaller than 0. The inverse gamma (IG) distribution is made uninformative in this study by setting the two arguments of the distribution (the shape and the rate parameter) to the same value. In the current study we used inverse gamma distributions for residual variances and for the inter-individual variance of models with only one random parameter. In these priors the shape and rate parameter were equal to 0.000001, so that,

$$\psi \sim IG(0.000001, 0.000001), \tag{5.35}$$

where ψ represents all uncorrelated variance parameters of a model. For an AR(1) model with a random mean μ, but fixed AR-parameter ϕ, the priors for the variance terms are,

$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001), \tag{5.36}$$

and,

$$\tau_\mu^2 \sim IG(0.000001, 0.000001). \tag{5.37}$$

Finally, in models where more than one parameter is allowed to vary across individuals, it is possible that these random parameter are correlated with each other. To account for this, the variance terms in models with more than one random parameter are usually not assigned separate inverse gamma distributions, but are given a joint prior distribution, the Inverse Wishart (IW). This IW distribution is a prior for the entire covariance matrix of a model, and for a model with three random parameters θ_1, θ_2, and θ_3, it can be written as,

$$\begin{bmatrix} \tau_{\theta_1}^2 & & \\ \tau_{\theta_1\theta_2} & \tau_{\theta_2}^2 & \\ \tau_{\theta_1\theta_3} & \tau_{\theta_2\theta_3} & \tau_{\theta_3}^2 \end{bmatrix} \sim IW(R, df), \tag{5.38}$$

where the left side of Equation 5.38 represents the covariance matrix of the model, R is a scale matrix that positions the distribution in multivariate space, and df are the

degrees of freedom of the distribution which determine how informative it is. For an AR(1) model with both a random mean μ and a random AR-parameter ϕ, the IW prior would be,

$$\begin{bmatrix} \tau_\mu^2 & \\ \tau_{\mu\phi} & \tau_\phi^2 \end{bmatrix} \sim IW(R, df). \tag{5.39}$$

Usually, an identity matrix is used for scale matrix R, but depending on the range of scores in the data this may not always be appropriate. In this study we will use data-based variance estimates on the diagonal of scale matrix R. To make the IW an uninformative prior, the df's need to be set to the number of random effects in the model, which would be 3 in the first example, and 2 for the AR(1) model with random mean and AR-parameter.

In the following sections we will specify the priors used for the different multilevel models fitted to the empirical data in this paper.

Based on preliminary convergence checks, the number of iterations for the Bayesian estimation procedures was set to 10,000 with a burn-in of 5,000.

Priors of the AR(1) Model with Fixed Mean and AR-parameter

For the AR(1) model with fixed mean and AR-parameter, both the mean and the AR-parameter are fixed effects. For both these parameters, normal distributions with 0 means and variances equal to 10^{12} were therefore chosen as priors. That is,

$$\mu \sim \mathcal{N}(0, 10^{12}) \tag{5.40}$$

$$\phi \sim \mathcal{N}(0, 10^{12}). \tag{5.41}$$

For the residual variance we specified an inverse gamma distribution as a prior, to get,

$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001). \tag{5.42}$$

The R-code for this model is given in section 10.1 of Chapter 5.

Priors of the AR(1) Model with Random Mean and Fixed AR-parameter

In this variation of the AR(1) model, the (overall) mean and the AR parameter are fixed parameters, while the residual variance and the inter-individual variance in the mean are variance terms. Therefore, the prior distributions for the overall mean and AR-parameter, are equal to,

$$\mu_\mu \sim \mathcal{N}(0, 10^{12}) \tag{5.43}$$
$$\phi \sim \mathcal{N}(0, 10^{12}), \tag{5.44}$$

while the priors for the residual variance and the the inter-individual variance in the mean are,

$$\sigma^2_\varepsilon \sim IG(0.000001, 0.000001) \tag{5.45}$$
$$\sigma^2_\mu \sim IG(0.000001, 0.000001). \tag{5.46}$$

The R-code for this model is given in section 10.2 of Chapter 5.

Priors of the AR(1) Model with Fixed Mean and Random AR-parameter

For the AR(1) model with fixed mean and random AR-parameter, both the mean and the (overall) AR-parameter are fixed effects. The priors for these parameters are,

$$\mu \sim \mathcal{N}(0, 10^{12}) \tag{5.47}$$
$$\mu_\phi \sim \mathcal{N}(0, 10^{12}). \tag{5.48}$$

For the residual variance and the inter-individual variance in the AR-parameter we specified inverse gamma distributions as priors, to get,

$$\sigma^2_\varepsilon \sim IG(0.000001, 0.000001) \tag{5.49}$$
$$\tau^2_\phi \sim IG(0.000001, 0.000001). \tag{5.50}$$

The R-code for this model is given in section 10.3 of Chapter 5.

Priors of the AR(1) Model with Random Mean and AR-parameter

In this variation of the AR(1) model, both the (overall) mean and the (overall) AR-parameter are fixed parameters. Therefore, for the overall mean and AR-parameter, the prior distributions are equal to,

$$\mu_\mu \sim \mathcal{N}(0, 10^{12}) \tag{5.51}$$
$$\mu_\phi \sim \mathcal{N}(0, 10^{12}), \tag{5.52}$$

The priors for the residual variance and the (possibly) related inter-individual variance in the mean and AR-parameter are,

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.53}$$

$$\begin{bmatrix} \tau_{\mu}^2 & \\ \tau_{\mu\phi} & \tau_{\phi}^2 \end{bmatrix} \sim IW(R, df), \tag{5.54}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & .01 \end{bmatrix} \text{ and,} \tag{5.55}$$

$$df = 2. \tag{5.56}$$

The R-code for this model is given in section 10.4 of Chapter 5.

Priors of the LGC Model with Fixed Intercept and Slope

In this model, the mean and slope are fixed effects, while the residual variance is a variance term, making the priors equal to,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.57}$$

$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.58}$$

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.59}$$

The R-code for this model is given in section 10.5 of Chapter 5.

Priors of the LGC Model with Random Intercept and Fixed Slope

In this model, the overall intercept and slope are fixed effects, while the residual variance and the inter-individual variance in the intercept are variance terms, making the priors equal to,

$$\mu_{\alpha} \sim \mathcal{N}(0, 10^{12}) \tag{5.60}$$

$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.61}$$

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.62}$$

$$\sigma_{\alpha}^2 \sim IG(0.000001, 0.000001) \tag{5.63}$$

The R-code for this model is given in section 10.6 of Chapter 5.

Priors of the LGC Model with Fixed Intercept and Random Slope

In this model, the overall slope and intercept are fixed effects, while the residual variance and the inter-individual variance in the slope are variance terms, making the priors equal to,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.64}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.65}$$
$$\sigma^2_\varepsilon \sim IG(0.000001, 0.000001) \tag{5.66}$$
$$\sigma^2_\beta \sim IG(0.000001, 0.000001) \tag{5.67}$$

The R-code for this model is given in section 10.7 of Chapter 5.

Priors of the LGC Model with Random Intercept and Slope

In this model, the overall slope and overall intercept are fixed effects, while the residual variance, the inter-individual variance in the intercept, and the inter-individual variance in the slope are variance terms. In addition, the random intercept and slope might be correlated, which means that we should use an Inverse Wishart prior for the random effects. The priors for this model are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.68}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.69}$$
$$\sigma^2_\varepsilon \sim IG(0.000001, 0.000001) \tag{5.70}$$
$$\begin{bmatrix} \tau^2_\alpha & \\ \tau_{\alpha\beta} & \tau^2_\beta \end{bmatrix} \sim IW(R, df). \tag{5.71}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.72}$$
$$df = 2. \tag{5.73}$$

The R-code for this model is given in section 10.8 of Chapter 5.

Priors of the QGC Model with Fixed Intercept, Slope, and Quadratic Term

For this model the prior distributions are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.74}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.75}$$
$$\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.76}$$
$$\sigma^2_{\varepsilon} \sim IG(0.000001, 0.000001) \tag{5.77}$$

The R-code for this model is given in section 10.9 of Chapter 5.

Priors of the QGC Model with a Random Intercept and Fixed Slope and Quadratic Term

For this model the prior distributions are,

$$\mu_{\alpha} \sim \mathcal{N}(0, 10^{12}) \tag{5.78}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.79}$$
$$\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.80}$$
$$\sigma^2_{\varepsilon} \sim IG(0.000001, 0.000001) \tag{5.81}$$
$$\sigma^2_{\alpha} \sim IG(0.000001, 0.000001) \tag{5.82}$$

The R-code for this model is given in section 10.10 of Chapter 5.

Priors of the QGC Model with Random Slope and Fixed Intercept and Quadratic Term

For this model the prior distributions are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.83}$$
$$\mu_{\beta} \sim \mathcal{N}(0, 10^{12}) \tag{5.84}$$
$$\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.85}$$
$$\sigma^2_{\varepsilon} \sim IG(0.000001, 0.000001) \tag{5.86}$$
$$\sigma^2_{\beta} \sim IG(0.000001, 0.000001) \tag{5.87}$$

The R-code for this model is given in section 10.11 of Chapter 5.

Priors of the QGC Model with a Random Quadratic Term and Fixed Intercept and Slope

For this model the prior distributions are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.88}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.89}$$
$$\mu_\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.90}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.91}$$
$$\sigma_\eta^2 \sim IG(0.000001, 0.000001) \tag{5.92}$$

The R-code for this model is given in section 10.12 of Chapter 5.

Priors of the QGC Model with a Random Intercept and Slope, and a Fixed Quadratic Term

For this model, the priors are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.93}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.94}$$
$$\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.95}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.96}$$
$$\begin{bmatrix} \tau_\alpha^2 & \\ \tau_{\alpha\beta} & \tau_\beta^2 \end{bmatrix} \sim IW(R, df). \tag{5.97}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.98}$$
$$df = 2. \tag{5.99}$$

The R-code for this model is given in section 10.13 of Chapter 5.

Priors of the QGC Model with a Random Intercept and Quadratic Term, and a Fixed Slope

For this model, the priors are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.100}$$
$$\mu_\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.101}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.102}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.103}$$
$$\begin{bmatrix} \tau_\alpha^2 & \\ \tau_{\alpha\eta} & \tau_\eta^2 \end{bmatrix} \sim IW(R, df). \tag{5.104}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.105}$$

$$df = 2. \tag{5.106}$$

The R-code for this model is given in section 10.14 of Chapter 5.

Priors of the QGC Model with a Random Slope and Quadratic Term, and a Fixed Intercept

For this model, the priors are,

$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.107}$$

$$\mu_\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.108}$$

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.109}$$

$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.110}$$

$$\begin{bmatrix} \tau_\beta^2 & \\ \tau_{\beta\eta} & \tau_\eta^2 \end{bmatrix} \sim IW(R, df). \tag{5.111}$$

where,

$$R = \begin{bmatrix} 500 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.112}$$

$$df = 2. \tag{5.113}$$

The R-code for this model is given in section 10.15 of Chapter 5.

Priors of the QGC Model with Random Intercept, Slope, and Quadratic Term

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.114}$$

$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.115}$$

$$\mu_\eta \sim \mathcal{N}(0, 10^{12}) \tag{5.116}$$

$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.117}$$

$$\begin{bmatrix} \tau_\alpha^2 & & \\ \tau_{\alpha\beta} & \tau_\beta^2 & \\ \tau_{\alpha\eta} & \tau_{\beta\eta} & \tau_\eta^2 \end{bmatrix} \sim IW(R, df), \tag{5.118}$$

where,

$$R = \begin{bmatrix} 1000 & & \\ 0 & 500 & \\ 0 & 0 & 100 \end{bmatrix} \text{ and,} \tag{5.119}$$

$$df = 3. \tag{5.120}$$

The R-code for this model is given in section 10.16 of Chapter 5.

Priors of the ALT Model with Fixed Constant and Change parameter

For this model, the priors are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.121}$$

$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.122}$$

$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.123}$$

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.124}$$

Priors of the ALT Model with Random Constant and Fixed Change parameter

For this model, the priors are,

$$\mu_{\alpha} \sim \mathcal{N}(0, 10^{12}) \tag{5.125}$$

$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.126}$$

$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.127}$$

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.128}$$

$$\sigma_{\alpha}^2 \sim IG(0.000001, 0.000001) \tag{5.129}$$

Priors of the ALT Model with Fixed Constant and Random Change parameter

For this model, the priors are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.130}$$

$$\mu_{\beta} \sim \mathcal{N}(0, 10^{12}) \tag{5.131}$$

$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.132}$$

$$\sigma_{\varepsilon}^2 \sim IG(0.000001, 0.000001) \tag{5.133}$$

$$\sigma_{\beta}^2 \sim IG(0.000001, 0.000001) \tag{5.134}$$

Priors of the ALT Model with Random Constant and Change parameter

For this model, the priors are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.135}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.136}$$
$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.137}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.138}$$
$$\begin{bmatrix} \tau_\alpha^2 & \\ \tau_{\alpha\beta} & \tau_\beta^2 \end{bmatrix} \sim IW(R, df). \tag{5.139}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.140}$$
$$df = 2. \tag{5.141}$$

Priors of the ALT Model with Fixed Intercept and Slope

For this model, the priors are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.142}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.143}$$
$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.144}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.145}$$

Priors of the ALT Model with Random Intercept and Fixed Slope parameter

For this model, the priors are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.146}$$
$$\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.147}$$
$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.148}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.149}$$
$$\sigma_\alpha^2 \sim IG(0.000001, 0.000001) \tag{5.150}$$

Priors of the ALT Model with Fixed Intercept and Random Slope parameter

For this model, the priors are,

$$\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.151}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.152}$$
$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.153}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.154}$$
$$\sigma_\beta^2 \sim IG(0.000001, 0.000001) \tag{5.155}$$

Priors of the ALT Model with Random Intercept and Slope parameter

For this model, the priors are,

$$\mu_\alpha \sim \mathcal{N}(0, 10^{12}) \tag{5.156}$$
$$\mu_\beta \sim \mathcal{N}(0, 10^{12}) \tag{5.157}$$
$$\phi \sim \mathcal{N}(0, 10^{12}) \tag{5.158}$$
$$\sigma_\varepsilon^2 \sim IG(0.000001, 0.000001) \tag{5.159}$$
$$\begin{bmatrix} \tau_\alpha^2 & \\ \tau_{\alpha\beta} & \tau_\beta^2 \end{bmatrix} \sim IW(R, df). \tag{5.160}$$

where,

$$R = \begin{bmatrix} 1000 & \\ 0 & 500 \end{bmatrix} \quad \text{and,} \tag{5.161}$$
$$df = 2. \tag{5.162}$$

5.6 Appendix B: The DIC

Similar to other Information Criteria, like the AIC (Akaike, 1973) and BIC (Schwarz, 1978), the DIC (Spiegelhalter et al., 2002) can be seen as consisting of two parts; one part that measures model (mis)fit, and a second part that quantifies the dimensionality, or complexity of a model. Specifically, the DIC can be written as,

$$\text{DIC} = -2 \log f(y|\tilde{\theta}) + 2p_V. \tag{5.163}$$

where θ is a vector containing all model parameters (e.g., the AR-parameter ϕ, the mean parameter μ, and the residual variance σ_ε^2 of the AR(1) model with fixed mean

and AR-parameter), $\tilde{\theta}$ is a Bayesian estimate of this vector, $-2\log f(y|\tilde{\theta})$ is a measure of model misfit (called the deviance) that represents the difference in fit between the model under consideration and a hypothetical "true" model that would fit the data perfectly, and p_V is an estimate of the complexity of the model expressed as the number of effective model parameters. The term $f(y|\tilde{\theta})$ that is part of the deviance is simply the density of the sample data given the Bayesian estimate of the parameter vector. In the current study we assume that this density is normally distributed, but other distributions are also possible (e.g. a gamma distribution as in Thompson, Nixon, and Grieve (2006), or a log-normal distribution as in Stevens, O'Hagan, and Miller (2003)). The density of the data used with all the models discussed in the previous section are given in Chapter 5. Note that, since the deviance is a measure of misfit, lower values on the DIC imply a "better" model. In addition, Equation 5.163 shows that, like other model selection procedures, model selection based on the DIC is based on a trade-off between fit and complexity. If two models fit the data equally well (i.e., their deviances are equal), then the model with the lowest complexity will be selected.

Both the deviance and the number of effective model parameters are easily obtained through Gibbs-sampling (Gelman et al., 2004, p. 287-289). If we indicate the current iteration of the Gibbs-sampler with k (with $k = 1 \ldots K$) and the vector with the parameter estimates in iteration k as θ_k, the Bayesian estimate of the deviance can be written as,

$$-2\log f(y|\tilde{\theta}) = \frac{1}{K}\sum_{k=1}^{K}(-2\log f(y|\theta_k). \tag{5.164}$$

So the deviance can easily be obtained from the Gibbs-sampler by calculating the value of the deviance in each iteration of the Gibbs-sampler and subsequently calculating the mean of these values. Model complexity can be estimated in different ways using a Gibbs-sampler. Spiegelhalter originally estimated the parameter p_D as a measure for the number of effective parameters in the model (Spiegelhalter et al., 2002). This value can be written as,

$$p_D = -2\log f(y|\tilde{\theta}) - (-2\log(y|\bar{\theta})) \quad \text{where,} \tag{5.165}$$

$$\bar{\theta} = \frac{1}{K}\sum_{k=1}^{K}(\theta_k). \tag{5.166}$$

So, Spiegelhalter subtracted the value of the deviance calculated at the mean values of the model parameters ($\bar{\theta}$) from the mean value of the deviance to get an estimate of model complexity. This method for calculating model complexity usually works fine, but is not invariant to reparametrization of the model, and can result in negative estimates for the effective number of parameters in certain situations. An alternative

estimate for the effective number of parameters, suggested by Gelman et al. (2004) is estimate p_V,

$$p_V = \frac{1}{2K} \sum_{k=1}^{K} (-2 \log f(y|\theta_k) - (-2 \log f(y|\tilde{\theta})). \tag{5.167}$$

This estimate calculates the variance in the deviance values across the k iterations of the Gibbs-sampler and divides it by two to get the number of effective parameters. Estimate p_V solves the problems associated with p_D and is generally very robust and accurate. The only requisite is the use of weak or uninformative priors. Given the advantages of p_V, that is the estimate we used in this study.

6

Manual for the R-Code used in the article: Bayesian Net Benefit Regression on Longitudinal Cost-Effectiveness Data from a Multi-Center RCT

6.1 Introduction

In the study by Wetzelaer, Jongerling, Arntz, and Evers (2015) we used several Bayesian multilevel models for the analysis of longitudinal cost-effectiveness data from (multi-center) Randomized Controlled Trials (RCT). Specifically we fit a First-Order Autoregressive (AR(1)) Model, a Latent Growth Curve (LGC) Model, a Quadratic Growth Curve (QGC) Model, and a Autoregressive Latent Trajectory (ALT) Model to the Net Benefit (NB) of several treatments for Borderline Personality disorder, where the NB was allowed to be either normally, lognormally, or gamma distributed. In this short manual we illustrate how applied researchers can fit these models to their own data by describing 1) the required software, 2) the required format of the input files, 3) the actual model-code and where to find it, and 4) the output generated by the code. Note that the aim of this manual is merely to aid applied researcher in *running* the code on their own data. As such, this manual will not go into details on the theory and math behind the estimation. Bayesian Statistics, for example, will not be extensively discussed. For this, and for a more in-depth explanation of the different models and their components, the interested reader is referred to the article by Wetzelaer et al. (2015).

This chapter is available online as a manual for the analyses in Wetzelaer, P., Jongerling, J., Arntz, A., & Evers, S. (2015). Bayesian Net Benefit Regression on Longitudinal Cost-Effectiveness Data from a Multi-Center RCT.

J. Jongerling wrote the manual and all the provided computer code.

6.2 Required Software

To fit the Bayesian multilevel models, researchers need to install both the R statistical software (R Core Team, 2014) and the JAGS (Just Another Gibbs Sampler) program (Plummer, 2012). These programs are available from the websites `www.r-project.org/` and `mcmc-jags.sourceforge.net/` respectively. The R program is needed to load and (if needed) manipulate researchers' data, set up the Bayesian analyses, and store the output, while the JAGS program is needed for the actual computations of the analyses using Markov Chain Monte Carlo (MCMC) simulation. The JAGS program can be called from inside R however, using the R2jags package (Yu-Sung & Masanao, 2014), so users do not have to switch between the two programs to analyze their data. The R2jags package can be installed form within the R-program by running the command `install.packages(R2jags)` and subsequently running `library(R2jags)` to load the package.

6.3 Format of the Input Files

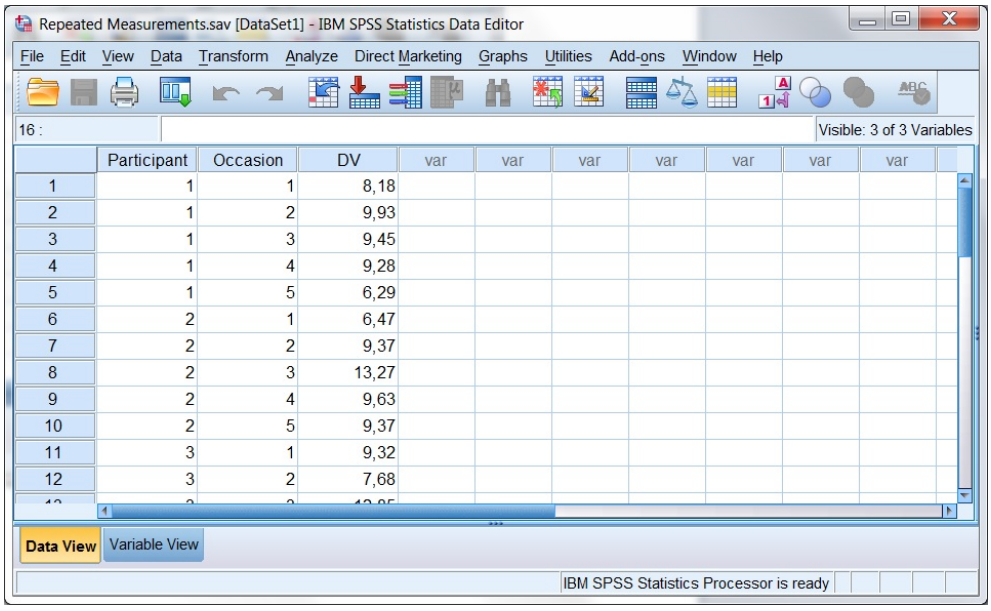
Researchers can provide their nested data in both the so called 'long' and 'wide' formats.In the 'long' format, the repeated observations of the dependent variable are all stored as scores on one variable, and each observation is entered on a separate line in the dataset. The individual and measurement occasion to which each score belong are subsequently indicated by two separate variables. Figure 6.1 shows part of an SPSS dataset that is structured in the 'long' format.

In these example data, the variable *Participant* is used to identify the different participants in the study, the variable *Occasion* identifies the different measurement occasions, and the variable *DV* contains the individual scores on the dependent variable. The first five entries on the *DV* variable can then be identified as the repeated measurements taken for participant 1 on occasion 1 through 5. Similarly, the scores on *DV* entered on line 6 through 10 are the repeated measurements belonging to participant number two.

In the 'wide' format the repeated observations of the dependent variable are stored as scores on *separate* variable, and all the information of an individual is entered on a *single* line in the dataset. Figure 6.2 shows the same data as Figure 6.1 but structured according to the 'wide' format.

In these example data, variables *DV1* through *DV5* represent the repeated measurements of the DV on occasion 1 through 5. The information of participant 1 is now all presented on the first row of the dataset, while all scores of participant 2 are on the second row. Once the data is structured according to one of the two appropriate formats, the dataset must be saved as a .txt file, and can then be read into R.

Fig. 6.1: SPSS data in 'long' format.



6.4 Model-Code

6.4.1 Loading the Data

A R-package containing an overall function to run all the code for the analyses is currently still in the making, but until then, applied researchers can run all the necessary analyses using the R-code available in Appendix A. In this section we present a short walk-through of this code.

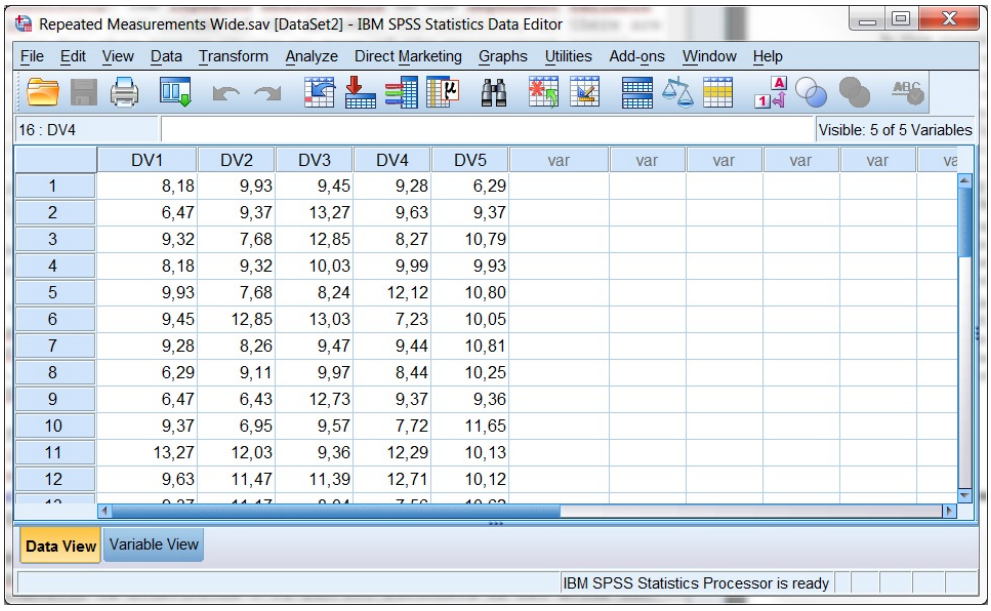
To fit the Bayesian Multilevel Models to their data, applied researchers first need read their datafile (structured using the instructions in the previous section) into R. This is done by the following lines of R-code.

```
Yrepeated <- read.csv("[Path To File]", sep = "", header = TRUE,
                      na.strings = "999999.00")
Yrepeated <- as.matrix(Yrepeated)
```

where the part that reads [Path To File] is replaced with the full pathname of the datafile (e.g., "C:/Study/Repeated/Datafile.txt"). Note that we are using 999999 as our missing values label, but this can be changed to any label that the researcher wants.

If the datafile was already in the 'wide' format (mentioned above and shown in Figure 6.2), the data is now correctly read in, and readers can skip the rest of this

Fig. 6.2: SPSS data in 'wide' format.



section and continue in section 4.2. However, if the data is in the 'long' format shown (mentioned above and shown in Figure 6.1), it needs to be restructured to 'wide' format before it can be analyzed with the code presented in this manual. This is done with the following lines of code. Here, we assume that, next to the repeated measurements (located in the fourth column of the dataset), there are three more variables in the dataset; two dummy variables ($D1$ and $D2$) located in the second an third column of the dataset that indicate group membership, and one variable (REC) located in the fifth column that indicated whether a participant recovered or not. However, extending the code to include more additional variables and/or variables located in different columns is straightforward.

```
YWidth <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=6)
D1 <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=1)
D2 <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=1)
Rec <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=6)
ind <- unique(Yrepeated[,1])

for (i in 1:(nrow(Yrepeated)/6)){
YWidth[i,] <- Yrepeated[Yrepeated[,1]==ind[i],4]
}
```

```
for (i in 1:(nrow(Yrepeated)/6)){
D1[i,] <- mean(Yrepeated[Yrepeated[,1]==ind[i],2])
}

for (i in 1:(nrow(Yrepeated)/6)){
D2[i,] <- mean(Yrepeated[Yrepeated[,1]==ind[i],3])
}

for (i in 1:(nrow(Ypim)/6)){
    REC[i,] <- Ypim[Ypim[,1]==ind[i],5]
}

Yrec <- rowMeans(REC)

Yrepeated <-  matrix(cbind(ind,D1,D2,YWidth,Yrec),nrow=
              length(ind))
```

The data is now ready to be sent to JAGS for analysis.

6.4.2 Analyzing the Data using JAGS

The JAGS program is specifically designed for Bayesian inference using MCMC simulation. Since JAGS uses Bayesian statistics it needs two additional pieces of information next to the data to be analyzed. First it needs a file describing the model to be estimated and the prior distributions for the model parameters, written in the BUGS language (Lunn et al., 2000). Second, it needs starting values for all the model parameters in order to start up the Gibbs-sampler (Gelman et al., 2004, p. 287-289). The model files for our models are available in Appendix B, while a more in-depth discussion of the models is available in Wetzelaer, Jongerling, Arntz, and Evers (2015). The starting values for the model parameters are specified in the call to the R2jags package that is used to run JAGS from within R. Before any of the model can fitted to the data we first need to load the R2jags package into R using the following code,

```
library(R2jags)
```

Next, the model specific code can be run.

AR(1) Model

One can fit four versions of the AR(1) model to the data; 1) an AR(1) model with a fixed mean and fixed AR-parameter, 2) an AR(1) model with a random mean and fixed AR-parameter, 3) an AR(1) model with fixed mean and random AR-parameter, and 4) an AR(1) model in which both parameters are random. The code for the AR model with the fixed mean is,

```
initsFixedMuAR <- function(){
  list(mu=rnorm(1), AR=runif(1,0,1))
  list(mu=rnorm(1), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

FixedMuAR <- jags(data = list("DV","T","N"), initsFixedMuAR,
             model.file="[Path To File]", parameters.to.save=
             c("mu","var.y", "AR"), n.chains =2, n.iter = Itt,
             n.burnin=Bi,n.thin=1, DIC=TRUE).

FixedMuAR <- autojags(FixedMuAR, n.iter = Itt, n.thin=1,
 n.update= 10, DIC=T)

FixedMuAR.mcmc <- as.mcmc(FixedMuAR)
traceplot(FixedMuAR.mcmc)
autocorr.plot(FixedMuAR.mcmc)
```

The first lines of code generates two lists with starting values for the model parameters by taking a random draw from a standard normal distribution for the mean, and by taking a draw from a uniform distribution over the interval (-1,1) for the AR-parameter. These two list are subsequently stored in the variable *initsFixedMuAR*.

The next three lines are used to specify the data that need to be send to JAGS, where the part between square brackets again needs to be changed by the researcher to match his data. The important thing here is that for the DV, the part between square brackets needs to specify those columns of the datamatrix created when the data was read into R that contain the repeated scores on the DV. The variables T and N simply contain the number of repeated measures and the number of people in the sample respectively.

The next line of code is the call to the JAGS program using the *jags* command form the R2jags package. The argument for this call are as follows. First the sample data is defined in a list containing the names of the three variables that were just created for 1) the repeated measures on the dependent variable, 2) the number of time points, and 3) the sample size. The second argument specifies the variable that contains the lists with starting values for the Gibbs-sampler (Gelman et al., 2004, p. 287-289). In the third argument, the model file to be used in the analysis is specified, and the part between the quotation marks should be replaced by the complete pathname for this file (e.g., C:/Study/Repeated/FixedARmodel-JAGS.txt). The specific model file needed for this model is given in section 10.1 through 10.4 of this manual. A zip-file containing

JAGS model files for all models discussed in this manual are available upon request. The fourth argument specifies for which model parameters the researcher wants to get estimates. Here we specify that we only want estimates for the mean ("mu"), the total variance of the time series ("var.y"), and the AR-parameter. One can ask for additional model parameter (if there are any) in this argument, as well as for the scores on the dependent variable ("DV"). Asking for these scores will results in the output of the JAGS program containing the original dataset with any missing values imputed. The fifth argument specifies the number of MCMC chains one want to run on the data. Note that, this number and the number of lists containing stating values should always match (so two chains need to be accompanied by two separate lists of starting values, as is the case here). We recommend specifying two or more separate MCMC chains so that the Gelman-Rubin statistic (Gelman et al., 2004, pg. 296-298), a statistic that gives some indication to whether the Gibbs-sampler converged properly, can be calculated by the JAGS program. The Gelman-Rubin statistic and model convergence will be mentioned in more detail in the section on the output of our model code. The next three arguments *n.iter*, *n.burnin*, and *n.thin* determine the behavior of the Gibbs-sampler. The first of these arguments specifies the total number of iterations of the Gibbs-sampler, while the second specifies the burn-in, that is, the number of iterations that will be used for the Gibbs-sampler to get to the appropriate posterior distribution of the saved model parameters. These burn-in iterations will be discarded and will therefore not influence the final parameter estimates. Based on our experiences with the models presented here, we advise a value of 10.000 for *n.iter* and a value of 5.000 for *n.burnin*. If the Gibbs-sampler does not converge (see section 8 of the model), these numbers can be increased The arguments n.thin, specifies the thining rate which indicated how many of the iterations of the Gibbs-sampler are kept for parameter estimation. Specifically, a thining rate of 1 indicates that all iterations are used, while a thining rate of 2 means that every other iterations is used. This argument is only of interest when the model does not converge properly and one wants to discard intermediate iterations of the Gibbs-sampler because the correlation between successive iterations is too high. Usually, this argument can be set to 1. The final argument specifies whether the researcher want the Deviance Information Criterion (DIC) (Spiegelhalter et al., 2002) value for the model, and can be equal to either "TRUE" or "FALSE". The DIC is a measure of model fit, and out of a set of models, the model with the lowest DIC value is the model that fits the sample data best.

The last three lines are again used to check for convergence of the Gibbs-sampler and to solve for any non-convergence issues. As mentioned above, this will be explained later in the *Output* section (section 8).

For the AR(1) model with the random mean and fixed AR-parameter the code is very similar,

```
initsRandomMuFixedAR <- function(){
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
```

```
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

RandomMuFixedAR <- jags(data = list("DV","T","N"),
   initsRandomMuFixedAR,
   model.file="[Path To File]",
   parameters.to.save= c("mean.mu","var.y",
   "AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)


RandomMuFixedAR <- autojags(RandomMuFixedAR, n.iter = Itt,
   n.thin=1, n.update=10,DIC=T)

RandomMuFixedAR.mcmc <- as.mcmc(RandomMuFixedAR)
traceplot(RandomMuFixedAR.mcmc)
autocorr.plot(RandomMuFixedAR.mcmc)
```

The only difference is that we now have separate means for each individual in the dataset, so instead of specifying starting values for "mu" we now specify starting values for the average value of this mean parameter across individuals "mean.mu". In addition, we also want an estimate for "mean.mu", so we have to include this parameter in the *parameters.to.save* argument. Note that adding the parameter "mu" to the *parameters.to.save* argument will now result in JAGS providing estimates of all the individual means.

For the AR(1) model with the fixed mean and random AR-parameter the code is
,

```
initsFixedMuRandomAR <- function(){
  list(mu=rnorm(1), mean.AR=runif(1,0,1))
  list(mu=rnorm(1), mean.AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

FixedMuRandomAR <- jags(data = list("DV","T","N"),
   initsFixedMuRandomAR,
   model.file="[Path To File]",
   parameters.to.save= c("mu","var.y",
```

```
   "mean.AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)


FixedMuRandomAR <- autojags(FixedMuRandomAR, n.iter = Itt,
   n.thin=1, n.update=10,DIC=T)

FixedMuRandomAR.mcmc <- as.mcmc(FixedMuRandomAR)
traceplot(FixedMuRandomAR.mcmc)
autocorr.plot(FixedMuRandomAR.mcmc)
```

The only difference with the code for the previous model is that now have a random AR-parameter instead of a random mean. This means that we now have separate AR-parameters for each individual in the dataset, so instead of specifying starting values for "AR" we now specify starting values for the average value of the AR-parameter across individuals "mean.AR". In addition, we also want an estimate for "mean.AR", so we have to include this parameter in the *parameters.to.save* argument. Note that adding the parameter "AR" to the *parameters.to.save* argument will now result in JAGS providing estimates of all the individual AR-parameters.

Finally the code for the AR(1) model with both a random mean and a random AR-parameter is given by,

```
initsRandomMuAR <- function(){
  list(mean.mu=rnorm(1), mean.AR=runif(1,0,1))
  list(mean.mu=rnorm(1), mean.AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

RandomMuAR <- jags(data = list("DV","T","N"), initsRandomMuAR,
              model.file="[Path To File]", parameters.to.save=
              c("mean.mu","var.y", "mean.AR"), n.chains =2,
              n.iter = Itt, n.burnin=Bi, n.thin=1, DIC=T)


RandomMuAR <- autojags(RandomMuAR, n.iter = Itt, n.thin=1,
  n.update= 10,DIC=T)

RandomMuAR.mcmc <- as.mcmc(RandomMuAR)
traceplot(RandomMuAR.mcmc)
autocorr.plot(RandomMuAR.mcmc)
```

As for the AR(1) model with a random mean we specify a starting value for the average value of this mean parameter across individuals, and like we did in the AR(1) model with a random AR-parameter we specify an initial value for the average value of the AR-parameter across individuals "mean.AR". In addition, we now want estimates for these two average parameter values, so we have to include both "mean.mu" and "mean.AR" in the *parameters.to.save* argument. Note that adding the parameter "mu" and/or "AR" to the *parameters.to.save* argument will now result in JAGS providing estimates of all the individual means and/or AR-parameters.

The code for the other models is very similar to the code for these two AR(1) models since that arguments for the call to JAGS will always be the same. Nevertheless we will shortly discuss the code for the LGC, QGC, and ALT model below, and we will highlight any differences between the codes of the different models.

LGC Model

For the LGC model we provide code for four different variations depending on which parameters are random. Specifically, we provide code for 1) a LGC model with a fixed intercept and a fixed slope, 2) a LGC model with a random intercept and a fixed slope, 3) a LGC model with a fixed intercept and a random slope, and 4) a LGC model with a random intercept and a random slope. The code for the first of the variations of the LGC, the LGC with a fixed intercept and a fixed slope, is as follows,

```
initsLGCFMS <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different
 Measurement Occasions]

LGCFMS <- jags(data = list("DV","T","N","Occasion"),
  initsLGCFMS, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta"),
  n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
  DIC=T)

LGCFMS <- autojags(LGCFMS, n.iter = Itt, n.thin=50,
```

```
  n.update= 10, DIC=T)

LGCFMS.mcmc <- as.mcmc(LGCFMS)
traceplot(LGCFMS.mcmc)
autocorr.plot(LGCFMS.mcmc)
```

The structure of this code is again similar to that of the two AR(1) models mentioned above. The first lines of code specify two list of starting values for the model parameters, where *alpha* represents the intercept of the LGC model and *beta* represents the slope, and stores these lists in a variable *initsLGCFMS*. The next four lines of code again specify the data that need to be send to JAGS, where the part between square brackets again needs to be changed by the researcher to match his data. In addition to the variables also created for the AR(1) model the LGC model also requires the specification of an indicator variable for the different measurement occasions *Occasion*. This *Occasion* variable is merely a list of integer values (starting at 0), with one value for each measurement occasion. So, if there are 7 measurement occasions, the *Occasion* variable is simply a vector containing the number 0 through 6. The next line of code is again the call to the JAGS program, which has the same arguments as were discussed with the AR(1) models.

When we allow either the intercept (*alpha*), the slope (*beta*), or both to be random across individuals, the changes to the code of the LGC model are similar to the changes made to the AR(1) code when we allow for intra-individual differences in the mean. Specifically, the parameter that is allowed to be random will be replaced by it average value across individuals in the list of starting values. In addition, parameter estimates for this average value across individuals will now be asked for in the *parameters.to.save* argument (although the individual estimates for the parameter can also be requested). So, for the LGC model with a random intercept, the code becomes,

```
initsLGCRMFS <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCRMFS <- jags(data = list("DV","T","N","Occasion"),
```

```
   initsLGCRMFS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y", "beta"),
   n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)


LGCRMFS <- autojags(LGCRMFS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

LGCRMFS.mcmc <- as.mcmc(LGCRMFS)
traceplot(LGCRMFS.mcmc)
autocorr.plot(LGCRMFS.mcmc)
```

For the LGC model with a random slope the code becomes,

```
initsLGCFMRS <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCFMRS <- jags(data = list("DV","T","N","Occasion"),
   initsLGCFMRS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "mean.beta"),
   n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)


LGCFMRS <- autojags(LGCFMRS, n.iter = Itt, n.thin=1, n.update= 10,
           DIC=T)

LGCFMRS.mcmc <- as.mcmc(LGCFMRS)
traceplot(LGCFMRS.mcmc)
autocorr.plot(LGCFMRS.mcmc)
```

And for the LGC model with a random intercept and a random slope the code becomes,

```
initsLGCRMRS <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
       mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
       mean=1000,sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCRMRS <- jags(data = list("DV","T","N","Occasion"),
           initsLGCRMRS, model.file="[Path To File]",
           parameters.to.save=c("mean.alpha","var.y",
           "mean.beta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
            n.thin=1, DIC=T)


LGCRMRS <- autojags(LGCRMRS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

LGCRMRS.mcmc <- as.mcmc(LGCRMRS)
traceplot(LGCRMRS.mcmc)
autocorr.plot(LGCRMRS.mcmc)
```

When both the intercept and the slope are allowed to vary across individuals, as is the case in this last model, this raises the possibility that these two model parameters are correlated with each other. An estimates for this correlation can be acquired by adding "cor" to the *parameters.to.save* argument.

QGC Model

The only difference between a LGC model and a QGC model is that the QGC model contains one additional parameter (*eta*) that describes the quadratic trend in the data. Therefore, the main difference between the code for this model and the codes of the LGC model discussed above, is that the lists of starting values now have to contain this additional (quadratic) parameter, and that we will also ask for estimates

of this parameter in the *parameters.to.save* argument. In addition, this additional parameter mean that the number of variations of the QGC model is larger than for the LGC. For the LGC model, we provide code for 1) a LGC model with a fixed intercept and a fixed slope, 2) a LGC model with a random intercept and a fixed slope, 3) a LGC model with a fixed intercept and a random slope, and 4) a LGC model with a random intercept and a random slope. For the QGC model we provide code for 1) a QGC model with a fixed intercept, slope, and quadratic term 2) a QGC model with a random intercept and a fixed slope and quadratic term, 3) a QGC model with a fixed intercept and quadratic term, and a random slope, 4) a QGC model with a fixed intercept and slope, and a random quadratic term, 5) a QGC with a random intercept and slope, and a fixed quadratic term, 6) a QGC with a random intercept and quadratic term, and a fixed slope, 7) a QGC with a fixed intercept and a random slope and quadratic term, and 8) a QGC model in which all three parameters are random.

The code for the QGC model with all three parameters fixed is,

```
initsQGCFMFSFSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
       mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
       mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMFSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMFSFSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta",
  "eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCFMFSFSQ <- autojags(QGCFMFSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMFSFSQ.mcmc <- as.mcmc(QGCFMFSFSQ)
traceplot(QGCFMFSFSQ.mcmc)
autocorr.plot(QGCFMFSFSQ.mcmc)
```

The code for the QGC model with a random intercept is,

```
initsQGCRMFSFSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMFSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMFSFSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y", "beta",
  "eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCRMFSFSQ <- autojags(QGCRMFSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMFSFSQ.mcmc <- as.mcmc(QGCRMFSFSQ)
traceplot(QGCRMFSFSQ.mcmc)
autocorr.plot(QGCRMFSFSQ.mcmc)
```

The code for the QGC model with a random slope is,

```
initsQGCFMRSFSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]
```

```
QGCFMRSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMRSFSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "mean.beta",
  "eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCFMRSFSQ <- autojags(QGCFMRSFSQ, n.iter = Itt, n.thin=1,
n.update= 10, DIC=T)

QGCFMRSFSQ.mcmc <- as.mcmc(QGCFMRSFSQ)
traceplot(QGCFMRSFSQ.mcmc)
autocorr.plot(QGCFMRSFSQ.mcmc)
```

The code for the QGC model with a random quadratic term is,

```
initsQGCFMFSRSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1, mean=1000,
  sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1, mean=1000,
  sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMFSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMFSRSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta",
  "mean.eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCFMFSRSQ <- autojags(QGCFMFSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMFSRSQ.mcmc <- as.mcmc(QGCFMFSRSQ)
traceplot(QGCFMFSRSQ.mcmc)
autocorr.plot(QGCFMFSRSQ.mcmc)
```

The code for the QGC model with a random intercept and slope is,

```
initsQGCRMRSFSQ <- function(){
```

```
  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMRSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMRSFSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "mean.beta", "eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSFSQ <- autojags(QGCRMRSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMRSFSQ.mcmc <- as.mcmc(QGCRMRSFSQ)
traceplot(QGCRMRSFSQ.mcmc)
autocorr.plot(QGCRMRSFSQ.mcmc)
```

The code for the QGC model with a random intercept and quadratic term is,

```
initsQGCRMFSRSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]
```

```
QGCRMFSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMFSRSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y", "beta",
  "mean.eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCRMFSRSQ <- autojags(QGCRMFSRSQ, n.iter = Itt, n.thin=1,
n.update= 10, DIC=T)

QGCRMFSRSQ.mcmc <- as.mcmc(QGCRMFSRSQ)
traceplot(QGCRMFSRSQ.mcmc)
autocorr.plot(QGCRMFSRSQ.mcmc)
```

The code for the QGC model with a random slope and quadratic term is,

```
initsQGCFMRSRSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMRSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMRSRSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "mean.beta",
  "mean.eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCFMRSRSQ <- autojags(QGCFMRSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMRSRSQ.mcmc <- as.mcmc(QGCFMRSRSQ)
traceplot(QGCFMRSRSQ.mcmc)
autocorr.plot(QGCFMRSRSQ.mcmc)
```

Finally, the code for the QGC model in which all parameters are random is,

```
initsQGCRMRSRSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMRSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMRSRSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "mean.beta", "mean.eta"), n.chains =2,
  n.iter = Itt, n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSRSQ <- autojags(QGCRMRSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMRSRSQ.mcmc <- as.mcmc(QGCRMRSRSQ)
traceplot(QGCRMRSRSQ.mcmc)
autocorr.plot(QGCRMRSRSQ.mcmc)
```

ALT Model

The final Bayesian multilevel model that can be fitted with our code is the ALT model. This model can be viewed as a combination of a LGC model and a AR(1) model and the code for this model therefore also is a combination of the code for these two models. Specifically, the only difference between a LGC model and an ALT model is that the ALT model also contains an AR-parameter. Therefore, the biggest difference between the code for this model and the codes of the LGC model discussed above, is that the lists of starting values now also have to contain the AR parameter, and that we will also ask for estimates of this parameter in the *parameters.to.save* argument. Additionally however, the intercept and the slope of the growth curves of the ALT model are a function of the parameters *alfa*, *beta*, and the AR parameter. This means that the separate parameters *alfa* and *beta* do not have a clear interpretation in this model (Jongerling & Hamaker, 2011) and that we somehow have to model a combination of these parameters to get the intercept and the slope of the ALT model. We provided two ways of doing this by writing code for two different parametrizations

of the ALT model. In the first parametrization we add code to the model file that expresses how the model parameters and the intercept and slope of the growth curve are related. In other words, in this parametrization we simply model the intercept and slope as a function of the *alfa*, *beta*, and AR parameter. In the second parametrization we rewrite the ALT model as a LGC model with autoregression between the measurement errors at successive time points. These 2 parametrizations are mathematically equivalent, but the second one separates the autoregressive part from the latent growth curve part (Hamaker, 2005) which mean that in this parametrization of the ALT model, the separate model parameters *alfa*, *beta*, and AR, can simple be interpreted as the intercept, slope and AR-parameter respectively.

Starting with the first parametrization of the ALT model, we provide code for 1) an ALT model with a fixed *alfa* and a fixed *beta* parameter, 2) an ALT model with a random *alfa* and a fixed *beta* parameter, 3) an ALT model with a fixed *alfa* and a random *beta* parameter, and 4) an ALT model with a random *alfa* and a random *beta* parameter. The AR parameter is always model as fixed because our data contained to few time points to reliably model intra-individual differences in this parameter.

The code for the ALT model with all three parameters fixed is,

```
initsALTFAFB <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=1000,
  sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=1000,
  sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFAFB <- jags(data = list("DV","T","N","Occasion"),
   initsALTFAFB, model.file="[Path To File]",
   parameters.to.save= c("intercept","var.y", "slope",
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTFAFB <- autojags(ALTFAFB, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTFAFB.mcmc <- as.mcmc(ALTFAFB)
autocorr.plot(ALTFAFB.mcmc)
```

```
traceplot(ALTFAFB.mcmc)
```

As mentioned above the biggest difference with the code for the LGC model discussed above is that the lists of starting values (specified on the first 7 lines of code) now also contain the AR parameter, and that we ask for estimates of this parameter in the *parameters.to.save* argument. In addition, note that although the intercept and the slope are a function of several model parameters, it is the estimates of these 'composite' parameters that we request in the *parameters.to.save* argument.

The code for the ALT model with a random *alpha* parameter is,

```
initsALTRAFB <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRAFB <- jags(data = list("DV","T","N","Occasion"),
   initsALTRAFB, model.file="[Path To File]",
   parameters.to.save= c("mean.intercept","var.y",
   "slope","AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTRAFB <- autojags(ALTRAFB, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTRAFB.mcmc <- as.mcmc(ALTRAFB)
autocorr.plot(ALTRAFB.mcmc)
traceplot(ALTRAFB.mcmc)
```

Note that allowing *alpha* to vary across individuals results in the intercept of the growth curve to also be random. We therefore ask for the average of this composite parameter across individuals (*mean.intercept*) in the *parameters.to.save* argument.

The code for a ALT model with a random *beta* parameter is,

```
initsALTFARB <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
```

```
  mean=1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFARB <- jags(data = list("DV","T","N","Occasion"),
   initsALTFARB, model.file="[Path To File]",
   parameters.to.save= c("mean.intercept","var.y",
   "mean.slope","AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTFARB <- autojags(ALTFARB, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTFARB.mcmc <- as.mcmc(ALTFARB)
autocorr.plot(ALTFARB.mcmc)
traceplot(ALTFARB.mcmc)
```

Note that allowing *beta* to vary across individuals results in both the intercept and the slope of the growth curve to also be random. We therefore ask for the averages of both these composite parameters (*mean.intercept* and *mean.slope*) in the *parameters.to.save* argument.

Finally, the code for the ALT model with both a random *alpha* and a random *beta* parameter is,

```
initsALTRARB <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100),
  mean.beta=rnorm(1,mean=1000, sd=100),
  AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100),
  mean.beta=rnorm(1,mean=1000, sd=100),
  AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
```

```
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRARB <- jags(data = list("DV","T","N","Occasion"),
   initsALTRARB, model.file="[Path To File]",
   parameters.to.save= c("mean.intercept","var.y",
   "mean.slope","AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTRARB <- autojags(ALTRARB, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTRARB.mcmc <- as.mcmc(ALTRARB)
autocorr.plot(ALTRARB.mcmc)
traceplot(ALTRARB.mcmc)
```

Like an ALT model with a random *beta* parameter, an ALT model with both a random *alpha* and a random *beta* parameter has both a random intercept and a random slope. We therefore ask for the averages of both these composite parameters (*mean.intercept* and *mean.slope*) in the *parameters.to.save* argument. In this variation of the ALT model however, the inta-individual variance in the slope is less constrained than in an ALT model with just a random *beta* parameter.

For the second parametrization of the ALT model, in which we rewrite the ALT model as a LGC model with autoregression between the measurement errors at successive time points, we provide code for 1) an ALT model with a fixed intercept and a fixed slope, 2)an ALT model with a random intercept and a fixed slope, 3) an ALT model with a fixed intercept and a random slope, and 4) an ALT model with a random intercept and a random slope. We again keep the AR parameter fixed in all variations of the ALT model because our data contained to few time points to reliably model intra-individual differences in this parameter.

The code for these 4 variations of the ALT model is almost identical to that of the 4 variations of the model under the first paramerization. However, in this parametrization the interpretation of the model parameters was straight forward, and the *alpha* and *beta* parameter can once again be interpreted as the intercept and slope of the growth curve. The only difference with the code for the 4 variations of the ALT model under the first specification is therefore that we directly ask for estimates of (the average values of) these two model parameters under the *parameters.to.save* argument, and not for the composite parameters we used above.

The code for the ALT model with a fixed intercept, a fixed slope, and fixed AR parameter then becomes,

```
initsALTFMFS <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=1000,
```

```
  sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=1000,
  sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFMFS <- jags(data = list("DV","T","N","Occasion"),
   initsALTFMFS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "beta", "AR"),
   n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)

ALTFMFS <- autojags(ALTFMFS, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTFMFS.mcmc <- as.mcmc(ALTFMFS)
autocorr.plot(ALTFMFS.mcmc)
traceplot(ALTFMFS.mcmc)
```

The code for the ALT model with a random intercept becomes is,

```
initsALTRMFS <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100),
  beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100),
  beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRMFS <- jags(data = list("DV","T","N","Occasion"),
   initsALTRMFS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y", "beta",
```

```
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)

ALTRMFS <- autojags(ALTRMFS, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTRMFS.mcmc <- as.mcmc(ALTRMFS)
autocorr.plot(ALTRMFS.mcmc)
traceplot(ALTRMFS.mcmc)
```

The code for a ALT model with a random slope is,

```
initsALTFMRS <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100),
  mean.beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100),
  mean.beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFMRS <- jags(data = list("DV","T","N","Occasion"),
   initsALTFMRS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "mean.beta",
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTFMRS <- autojags(ALTFMRS, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTFMRS.mcmc <- as.mcmc(ALTFMRS)
autocorr.plot(ALTFMRS.mcmc)
traceplot(ALTFMRS.mcmc)
```

Finally, the code for the ALT model with both a random intercept and a random slope is,

```
initsALTRMRS <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100),
```

```
  mean.beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100),
  mean.beta=rnorm(1,mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRMRS <- jags(data = list("DV","T","N","Occasion"),
   initsALTRMRS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y",
   "mean.beta","AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTRMRS <- autojags(ALTRMRS, n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


ALTRMRS.mcmc <- as.mcmc(ALTRMRS)
autocorr.plot(ALTRMRS.mcmc)
traceplot(ALTRMRS.mcmc)
```

6.5 QGC Model with Group as Predictor for the Random Parameters

In our applied example the QGC model with a random intercept and slope had the best model fit. A logical next step therefore was adding predictors for the inter-individual differences in the intercept, the slope, or both. As a predictor we used group membership as indicated by the two dummy variables created when reading the data into R. Adding Group as a predictor for the random intercept implies that we can write the intercept α_i as,

$$\alpha_i = \gamma_{00} + \gamma_{10} * D1_i + \gamma_{20} * D2_i + \varepsilon_{\alpha i}, \tag{6.1}$$

where γ_{00} is the intercept for the reference group, γ_{10} is the regression coefficient for *D1* which quantifies the difference in intercept between the reference group and the group belonging to a score of 1 on *D1*, and γ_{20} is the regression coefficient for *D2* which quantifies the difference in intercept between the reference group and the group belonging to a score of 1 on *D2*. The term $\varepsilon_{\alpha i}$ is a random error term. Since the

intercept is now a function of the three parameters γ_{00}, γ_{10}, and γ_{20} we ask for these parameters in the *parameters.to.save* argument instead of for α. This makes the code for the QGC model with Group as a predictor for the intercept equal to,

```
initsQGCRMRSFSQPredM <- function(){
  list(gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=1000,
  sd=100))

  list(gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=1000,
  sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]


QGCRMRSFSQPredM <- jags(data = list("DV","T","N","Occasion",
   "D1", "D2"), initsQGCRMRSFSQPredM,
   model.file="[Path To File]",
   parameters.to.save=c("gamma00","gamma10",
   "gamma20","var.y", "Cov", "mean.beta",
   "eta"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)


QGCRMRSFSQPredM <- autojags(QGCRMRSFSQPredM, n.iter = Itt, n.thin=1,
                    n.update= 10, DIC=T)


QGCRMRSFSQPredM.mcmc <- as.mcmc(QGCRMRSFSQPredM)
autocorr.plot(QGCRMRSFSQPredM.mcmc)
traceplot(QGCRMRSFSQPredM.mcmc)
```

When we add Group as a predictor for the random slope, the parameter β_i can be written as,

$$\beta_i = \gamma_{01} + \gamma_{11} * D1_i + \gamma_{21} * D2_i + \varepsilon_{\beta i}, \tag{6.2}$$

where γ_{01} is the slope for the reference group, γ_{11} is the regression coefficient for *D1* which quantifies the difference in slope between the reference group and the group belonging to a score of 1 on *D1*, and γ_{21} is the regression coefficient for *D2* which quantifies the difference in slope between the reference group and the group belonging to a score of 1 on *D2*. The term $\varepsilon_{\beta i}$ is a random error term. Since the slope is now a function of the three parameters γ_{01}, γ_{11}, and γ_{21} we again ask for these parameters in the *parameters.to.save* argument instead of for β. This makes the code for the QGC model with Group as a predictor for the slope equal to,

```
initsQGCRMRSFSQPredS <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  mean.alpha=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))


  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  mean.alpha=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))

}

QGCRMRSFSQPredS <- jags(data = list("DV","T","N","Occasion",
   "D1", "D2"), initsQGCRMRSFSQPredS,
   model.file="[Path To File]",
                  parameters.to.save=c("gamma01","gamma11",
                  "gamma21", "var.y", "Cov", "mean.alpha",
                  "eta"),n.chains =2, n.iter = Itt,
                  n.burnin=Bi, n.thin=1, DIC=T)


QGCRMRSFSQPredS <- autojags(QGCRMRSFSQPredS, n.iter = Itt, n.thin=1,
                  n.update= 10, DIC=T)


QGCRMRSFSQPredS.mcmc <- as.mcmc(QGCRMRSFSQPredS)
autocorr.plot(QGCRMRSFSQPredS.mcmc)
traceplot(QGCRMRSFSQPredS.mcmc)
```

Finally, the code for a QGC model in which Group is a predictor for both the intercept and the slope can be written as,

```
initsQGCRMRSFSQPredMS <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
```

```
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

}

QGCRMRSFSQPredMS <- jags(data = list("DV","T","N","Occasion",
"D1", "D2"), initsQGCRMRSFSQPredMS,
model.file="[Path To File]",
                    parameters.to.save=c("gamma00","gamma10",
                    "gamma20", "gamma01","gamma11", "gamma21",
                    "var.y", "Cov", "eta", "PredCon","PredST",
                    "PredCCT", "TotCon", "TotST", "TotCCT"),

                    n.chains =2, n.iter = Itt, n.burnin=Bi,
                    n.thin=1, DIC=T)


QGCRMRSFSQPredMS <- autojags(QGCRMRSFSQPredMS, n.iter = Itt, n.thin=1,
                    n.update= 10, DIC=T)

QGCRMRSFSQPredMS.mcmc <- as.mcmc(QGCRMRSFSQPredMS)
autocorr.plot(QGCRMRSFSQPredMS.mcmc)
traceplot(QGCRMRSFSQPredMS.mcmc)
```

Adding second level predictors to the (random) parameters of models other than the QGC model can be done in a similar manner. One just has to rewrite the mean parameter value across all individuals as a linear relation of the predictor(s) that one wants to add, and subsequently include the regression coefficients for these second level predictors in the *parameters.to.save* argument. Further instructions on adding second level predictors are given in Appendix B of the article by Wetzelaer, Jongerling, Arntz, and Evers (2015).

Finally, applied researcher might be interested in comparing different groups in terms of their total scores on the dependent variable *DV*. This can easily be done with our Bayesian models by adding a set of variables to the JAGS modelfiles that represent the sum scores of the *DV* across the different measurement occasions, for each individual group (see the section "*QGC Random Intercept and Slope, Fixed*

Quadratic Term: Predictor for Intercept and Slope" in Appendix B). In the code above for example, we added 6 additional statements to the *parameters.to.save* argument; *PredCon*, *PredST*, *PredCCT*, *TotCon*, *TotST*, and *TotCCT*. The first three of these statements (*PredCon*, *PredST*, and *PredCCT*) are vectors containing the predicted scores on the *DV* for the *Con*, *ST*, and *CCT* group, at each of the measurement occasions. The variables *TotCon*, *TotST*, and *TotCCT* subsequently, are the total scores on the dependent variable (across the different measurements) for the three different groups, obtained by adding the values in the *PredCon*, *PredST*, and *PredCCT* vector respectively. The posterior distributions of these total scores for the different groups can now be used to draw conclusions about relative performance.

Incidentally, the comparison of groups in terms of their total scores can be made even easier with the creation of (further) additional variables that directly represent the difference in total score between groups. Such variables are easily created by subtracting the total score of one group from that of another (e.g., subtracting the value of *TotCon* from that of *TotST*). The variables *STCon*, *STCCT*, and *CCTCon* in section "*QGC Random Intercept and Slope, Fixed Quadratic Term: Predictor for Intercept and Slope*" of Appendix B for instance, are examples of such difference score variables that can be used to directly evaluate relative performance.

6.6 Analyzing Data with Three Levels

Since clinical trial will often take place in several mental health centers, and these centers can be seen as an additional level in our hierarchical data (i.e., observations are nested within individuals, and these individuals are nested in the different centers), we also provide code for the analysis of a QGC model with a random intercept and a random slope, that includes such a third (center) level.

```
Center <- [Column of the Data Matrix That Contains the Center
           In Which The Individual Participated]
C <- [Number of Third Level Units]

initsQGCRMRSFSQL3 <- function(){
  list(mean.alpha=rnorm(C, mean=1000, sd=100),
  mean.beta=rnorm(C, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(C, mean=1000, sd=100),
  mean.beta=rnorm(C, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

}

QGCRMRSFSQL3 <- jags(data = list("DV","T","N","Occasion",
```

```
"Center","C"), initsQGCRMRSFSQL3,
model.file="[Path To File]",
parameters.to.save= c("mean.alpha","var.y",
                "Cov", "mean.beta", "eta", "MA","MB","VarA",
                "VarB"), n.chains =2, n.iter = Itt,
                n.burnin=Bi, n.thin=500, DIC=T)


QGCRMRSFSQL3 <- autojags(QGCRMRSFSQL3, n.iter = Itt, n.thin=500,
                n.update= 10, DIC=T)


QGCRMRSFSQL3.mcmc <- as.mcmc(QGCRMRSFSQL3)
autocorr.plot(QGCRMRSFSQL3.mcmc)
par(mfrow=c(3,3))
traceplot(QGCRMRSFSQL3.mcmc)
```

The only difference between the code for this three level QGC model and the two-level QGC model presented earlier is that we need to specify a variable that indicates to which third level unit an individual belongs *Center*, and a variable that indicates the number of third level units *C*. In addition, the *mean.alpha* and *mean.beta* term in the *parameters.to.save* argument will now give a separate intercept and linear slope value for each of the third level units. To also get the overall mean and the variance of the intercept and (linear) slopes across the third level units, we now have to ask for MA, MB, VarA, and VarB in *parameters.to.save*.

6.6.1 Three Level QGC Model With Group as Predictor for the Random Parameters

Like the two level QGC model, the three level QGC model can be extended to include predictors for the individual intercepts and slopes. As before, we use group membership (as indicated by the two dummy variables created when reading the data into R) as a predictor for the inter-individual differences in the intercepts and slopes, with the difference that there now is an additional level above the individual level on which we are modeling these inter-individual differences. When we added Group as a predictor for the random intercept and slope in the two level QGC model we showed α_i and β_i were then written as,

$$\alpha_i = \gamma_{00} + \gamma_{10} * D1_i + \gamma_{20} * D2_i + \varepsilon_{\alpha i}, \quad \text{and} \tag{6.3}$$

$$\beta_i = \gamma_{01} + \gamma_{11} * D1_i + \gamma_{21} * D2_i + \varepsilon_{\beta i}, \tag{6.4}$$

where γ_{00} and γ_{01} are the intercept and slope for the reference group, γ_{10} and γ_{11} are the regression coefficients for *D1* which respectively quantify 1) the difference in

the intercepts, and 2) the difference in the slopes between the reference group and the group belonging to a score of 1 on *D1*. Finally, γ_{20} and γ_{21} are the regression coefficients for *D2* which respectively quantify 1) the difference in the intercepts, and 2) the difference in the slopes between the reference group and the group belonging to a score of 1 on *D2*. The terms $\varepsilon_{\alpha i}$ and $\varepsilon_{\beta i}$ are random error terms.

When we add a third level to the model the parameters γ_{00}, γ_{10}, γ_{20}, γ_{01}, γ_{11}, and γ_{21} can vary across the third level units, and we need to take this random variance on the third level into account. This is simply done by drawing a separate value for each of these six parameters, for each of the third level units, from a overarching distribution. Assuming normal distributions this implies,

$$\gamma_{00} \sim \mathcal{N}(\mu_{\gamma_{00}}, \tau^2_{\gamma_{00}}) \tag{6.5}$$
$$\gamma_{10} \sim \mathcal{N}(\mu_{\gamma_{10}}, \tau^2_{\gamma_{10}}) \tag{6.6}$$
$$\gamma_{20} \sim \mathcal{N}(\mu_{\gamma_{20}}, \tau^2_{\gamma_{20}}) \tag{6.7}$$
$$\gamma_{01} \sim \mathcal{N}(\mu_{\gamma_{01}}, \tau^2_{\gamma_{01}}) \tag{6.8}$$
$$\gamma_{11} \sim \mathcal{N}(\mu_{\gamma_{11}}, \tau^2_{\gamma_{11}}) \tag{6.9}$$
$$\gamma_{21} \sim \mathcal{N}(\mu_{\gamma_{21}}, \tau^2_{\gamma_{21}}), \tag{6.10}$$

where γ_{00}, γ_{10}, γ_{20}, γ_{01}, γ_{11}, and γ_{21} are vectors containing the separate parameter values for each higher level unit c; $\mu_{\gamma_{00}}$, $\mu_{\gamma_{10}}$, $\mu_{\gamma_{20}}$, $\mu_{\gamma_{01}}$, $\mu_{\gamma_{11}}$, and $\mu_{\gamma_{21}}$ are the overall means of these parameters, and $\tau^2_{\gamma_{00}}$, $\tau^2_{\gamma_{10}}$, $\tau^2_{\gamma_{20}}$, $\tau^2_{\gamma_{01}}$, $\tau^2_{\gamma_{11}}$, and $\tau^2_{\gamma_{21}}$ are the random variances in these parameters across the third level units. Subsequently, α_i and β_i can be written as,

$$\alpha_i = \gamma_{00}[c] + \gamma_{10}[c] * D1_i + \gamma_{20}[c] * D2_i + \varepsilon_{\alpha i}, \quad \text{and} \tag{6.11}$$
$$\beta_i = \gamma_{01}[c] + \gamma_{11}[c] * D1_i + \gamma_{21}[c] * D2_i + \varepsilon_{\beta i}, \tag{6.12}$$

where c between square brackets implies that we are selecting that value from the parameter vector that belongs to the higher level unit (c) to which an individual belongs.

The code to run this three-level model is very similar to that of the two-level QGC model with Group as a predictor for the random variances. It can be written as,

```
initsQGCRMRSFSQL3P <- function(){
  list(Mugamma00=rnorm(1, mean=1000, sd=100),
  Mugamma10=rnorm(1, mean=1000, sd=100),
  Mugamma20=rnorm(1, mean=1000, sd=100),
  Mugamma01=rnorm(1, mean=1000, sd=100),
  Mugamma11=rnorm(1, mean=1000, sd=100),
  Mugamma21=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))
```

```
  list(Mugamma00=rnorm(1, mean=1000, sd=100),
  Mugamma10=rnorm(1, mean=1000, sd=100),
  Mugamma20=rnorm(1, mean=1000, sd=100),
  Mugamma01=rnorm(1, mean=1000, sd=100),
  Mugamma11=rnorm(1, mean=1000, sd=100),
  Mugamma21=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))
  }

QGCRMRSFSQL3P <- jags(data = list("DV","T","N","Occasion",
 "D1", "D2", "Center", "C"),
 initsQGCRMRSFSQL3P,
 model.file="[Path To File]",
                parameters.to.save=c("Mugamma00","Mugamma10",
                "Mugamma20","Mugamma01","Mugamma11","Mugamma21",
                "var.y", "Cov", "eta"), n.chains =2,
                n.iter = Itt, n.burnin=Bi, n.thin=500, DIC=T)


QGCRMRSFSQL3P <- autojags(QGCRMRSFSQL3P, n.iter = Itt, n.thin=500,
                n.update= 10, DIC=T)


QGCRMRSFSQL3P.mcmc <- as.mcmc(QGCRMRSFSQL3P)
autocorr.plot(QGCRMRSFSQL3P.mcmc)
par(mfrow=c(3,3))
traceplot(QGCRMRSFSQL3P.mcmc)
```

The only difference is that asking for γ_{00}, γ_{10}, γ_{20}, γ_{01}, γ_{11}, and γ_{21} in the *parameters.to.save* argument will now give use separate values for each of the higher level units. To get the average values for these parameters across the higher level units we have to ask for $\mu_{\gamma_{00}}$, $\mu_{\gamma_{10}}$, $\mu_{\gamma_{20}}$, $\mu_{\gamma_{01}}$, $\mu_{\gamma_{11}}$, and $\mu_{\gamma_{21}}$ instead.

6.7 Analyzing Lognormal or Gamma distributed Data

The R-code presented above assumes that the NB data is normally distributed, but applying our code to lognormal and gamma distributed data is straightforward. If we want to apply a QGC model to lognormal data for example, one merely uses the logarithm of the data as the dependent value. Our R-script already contains the following code to fit a QGC model with random intercept and slope to lognormal NB data,

```
DV <- ((-1*DV)+(lambda/6))
```

```
DV <- DV + .0000001
DV <- log(DV)


initsQGCRMRSFSQPredMSlog <- function(){
  list(gamma01=rnorm(1, mean=6.91, sd=4.61), gamma11=rnorm(1,
   mean=6.91, sd=4.61), gamma21=rnorm(1, mean=6.91, sd=4.61),
   gamma00=rnorm(1, mean=6.91, sd=4.61), gamma10=rnorm(1,
   mean=6.91, sd=4.61), gamma20=rnorm(1, mean=6.91, sd=4.61),
   eta=rnorm(1, mean=6.91, sd=4.61))

  list(gamma01=rnorm(1, mean=6.91, sd=4.61), gamma11=rnorm(1,
  mean=6.91, sd=4.61), gamma21=rnorm(1, mean=6.91, sd=4.61),
  gamma00=rnorm(1, mean=6.91, sd=4.61), gamma10=rnorm(1,
  mean=6.91, sd=4.61), gamma20=rnorm(1, mean=6.91, sd=4.61),
  eta=rnorm(1, mean=6.91, sd=4.61))
  }

QGCRMRSFSQPredMSlog <- jags(data = list("DV","T","N","Occasion",
   "D1", "D2"), initsQGCRMRSFSQPredMSlog,
   model.file="[Path To File]",
                       parameters.to.save=c("gamma00","gamma10",
                       "gamma20", "gamma01","gamma11", "gamma21",
                       "var.y", "Cov", "eta"), n.chains =2,
                       n.iter = Itt, n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSFSQPredMSlog <- autojags(QGCRMRSFSQPredMSlog, n.iter = Itt,
   n.thin=1, n.update= 10, DIC=T)


QGCRMRSFSQPredMSlog.mcmc <- as.mcmc(QGCRMRSFSQPredMSlog)
autocorr.plot(QGCRMRSFSQPredMSlog.mcmc)
traceplot(QGCRMRSFSQPredMSlog.mcmc)
```

Since a lognormal distribution can only model data in which every score is larger than 0, and since the NB of different treatments is usually negative, the first line of this code first multiplies the DV with -1 to remove the negative values. The second line subsequently makes sure that there are no NB values equal to 0 by adding a small constant to the data. After making sure that all scores on the DV are larger than 0, we simply take the logarithm of the variable *DV*. The rest of the code is similar to that of the code used for normally distributed data. Only the starting values are adjusted to the new range of scores on the DV.

Like a lognormal distribution, a gamma distribution can only be fitted to data that does not contain values of 0 or lower. One therefore needs to run the same first

two lines of code as presented above to remove all values smaller than 1 from DV. In addition, we create a variable that indicates in which treatment group an individual was placed, to allow the different groups to have different rate parameters (mentioned below). The rest of the R-code used for gamma distributed data is the same as for normally distributed data. One does have to change the specified distribution of the data in the modelfile used by the JAGS program however, such that,

$$y_{it} \sim \text{Gamma}(\kappa_{it}, \nu_i) \quad \text{and,} \tag{6.13}$$

$$\kappa_{it} = [\text{Mean Trend of Model}]\nu_i. \tag{6.14}$$

where κ and ν are the shape and rate parameter of the Gamma distribution, and Equation 6.14 makes sure that the mean trend of the gamma distributed data follows the mean trend of the fitted model (since the mean of a gamma distribution is equal to $\frac{\mu_\kappa}{\mu_\nu}$). For a QGC model in which all three model parameters are random Equation 6.14 is equal to,

$$\kappa_{it} = \left(\delta_i + \gamma_i(t-1) + \theta_i(t-1)^2\right)\theta_i, \tag{6.15}$$

for example.

The code to fit a QGC model with random intercept and slope to gamma distributed NB data is,

```
DV <- ((-1*DV)+(lambda/6))
DV <- DV + .0000001
Group <- (D1+(2*D2)) + 1


initsQGCRMRSFSQPredMSgamma <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100),r=rep(30,3))

  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100),r=rep(30,3))

}
```

```
QGCRMRSFSQPredMSgamma <- jags(data = list("DV","T","N","Occasion",
                         "D1", "D2", "Group"),
                         initsQGCRMRSFSQPredMSgamma,
                         model.file="[Path To File]",
                         parameters.to.save=c("gamma00","gamma10",
                         "gamma20", "gamma01","gamma11", "gamma21",
                         "Cov", "eta"), n.chains =2, n.iter = Itt,
                         n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSFSQPredMSgamma <- autojags(QGCRMRSFSQPredMSgamma,
 n.iter = Itt, n.thin=1, n.update= 10,
 DIC=T)


QGCRMRSFSQPredMSgamma.mcmc <- as.mcmc(QGCRMRSFSQPredMSgamma)
autocorr.plot(QGCRMRSFSQPredMSgamma.mcmc)
traceplot(QGCRMRSFSQPredMSgamma.mcmc)
```

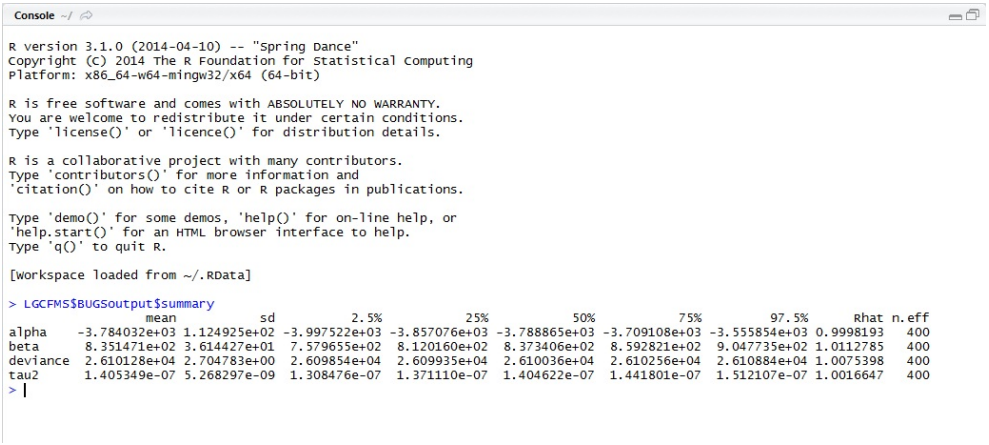
In Appendix B, a JAGS modelfile for the analysis of gamma distributed data with a QGC model with random intercept and slope, and a fixed quadratic term is given. For the other models, the modelfiles are easily adapted for gamma distributed data by changing the specified distribution of DV, and setting the shape parameter $\kappa_{i,t}$ equal to the appropriate mean trend multiplied by the rate parameter $\theta_{i,t}$.

6.8 Output

The code for each model provides the researcher with estimates for all the parameters specified in the *parameters.to.save* argument of the call to the JAGS program (Spiegelhalter et al., 2002). As an example, the output for the LGC-model with a fixed intercept and slope is given in Figure 6.3 As this figure shows, the JAGS program will return 1) the mean value, 2) the standard deviation, 3) the 2.5th, 25th, 50th, 75th, and 97.5th percentiles, 4) the Gelman-Rubin statistic (*Rhat*) (Gelman et al., 2004, pg. 296-298) for each of the parameters specified, and 5) the effective number of independent draws (Gelman, Carlin, Stern, & Rubin, 2004, pg. 298-299). Most of the values are self explanatory, but applied researchers are probably less familiar with the Gelman-Rubin statistic.

This statistic is a measure of model convergence, where convergence has to do with the fact that Bayesian inference is based on the posterior distribution of model parameter. Specifically, a Gibbs-sampler tries to construct an approximation of the posterior distribution of a parameter by repeatedly drawing values from the posterior distribution of a parameter, conditional on the value of all the other parameters in the model. All inference is subsequently based on this approximate posterior distribution. The mean or median of the distribution is used as a point estimate for example, while

Fig. 6.3: Example of Output form the JAGS Program.



the standard deviation of these distributions is can be as an equivalent of the standard error. It is therefore important that the Gibbs-sampler is indeed sampling form the correct distribution. Whether this is the case is indicated by the convergence of the Gibbs-sampler. If the sampler is indeed sampling from the (conditional) posterior distribution of a parameter we say the Gibbs-sampler converged, if it is not sampling from the correct posterior, we say the sampler did not converge. Although convergence can never be absolutely proven, there are a few checks that researchers can do to make sure that convergence was (at least) likely reached.

The first of these checks involves the Gelman-Rubin statistic (Gelman et al., 2004, pg. 296-298). If the value of this statistic is smaller than 1.2 for all estimated parameters, then that is an indication that the Gibbs-sampler probably converged. If it is larger than 1.2 convergence has not been reached, and a researcher should first try letting the Gibbs-sampler run for some additional iterations. This can be done by using the lines of code starting with the *autojags* command, which will make the Gibbs-sampler run for additional iterations until the statistics for every parameter is smaller than 1.2. So for the AR(1) model with fixed mean, the following line of code

```
FixedAR <- autojags(FixedAR, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T),
```

can be run to ask for additional iterations. Most of the arguments of this call we already saw in the original call to the JAGS program, with two exceptions. The first exception is the first argument, which points to the variable in which a researcher stored his or her original MCMC-simulation. The autojags command tells JAGS to run *n.iter* new MCMC-simulations, starting with the parameter values of the last original simulation, and to check if all Gelman-Rubin statistics are smaller than 1.2 after they are completed. If so, the MCMC-simulation stops and the new set of simulations is

stored in the variable in the first argument of the autojags command. If the *n.iter* new MCMC-iterations did not result in Gelman-Rubin statistics smaller than 1.2, the autojags commands runs another set of *n.iter* new MCMC-simulations, starting with the parameter values calculated in the last simulation (i.e., the last simulation of the first set of additional simulations), and keeps doing so until *n.update* new sets of simulations have been run. This *n.update* argument is the second new argument in the autojags command.

Researchers should always check the Gelman-Rubin statistics for all estimated parameter to make sure that it is lower that 1.2, but as we already mentioned, this is only the first check of (probable) convergence. The second is plotting the repeated draws of the Gibbs-sampler using a so called traceplot. In a traceplot, the value that the Gibbs-sampler drew for a specific parameter is on the y-axis, while the iteration of the Gibbs-sampler is on the x-axis. If the Gibbs-sampler converged, then this traceplot should not show any trend, and the repeated draws of a parameter value should randomly fluctuate around a constant mean value. An example of a traceplot for a 2-chained converged Gibbs-sampler is given in Figure 6.4.

For all the models discussed above, these plots are made with the lines of code that turn the variable containing the JAGS output into an MCMC object, and subsequently plot this object using the *traceplot* command. So to make these plots for the AR(1) model with fixed mean, we would run the lines,

```
FixedAR.mcmc <- as.mcmc(FixedAR)
traceplot(FixedAR.mcmc)
```

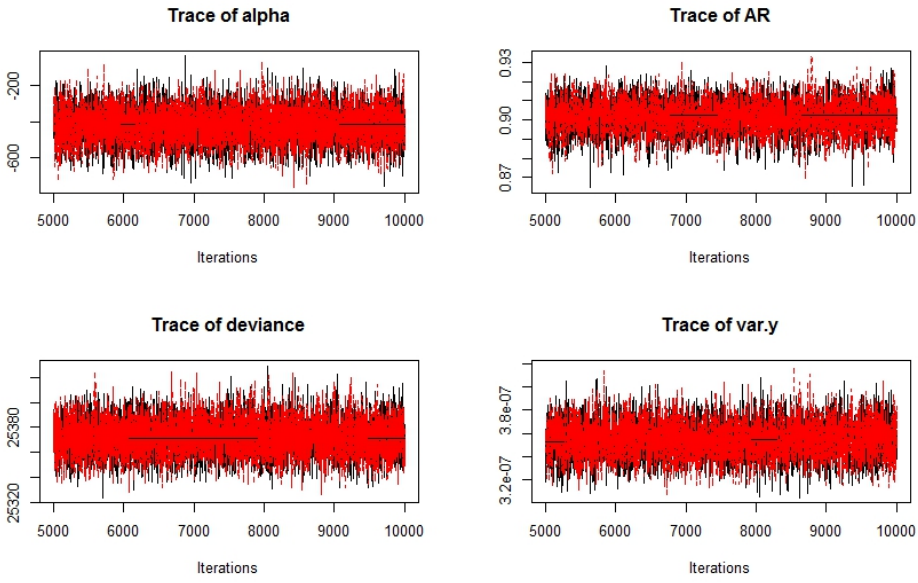
Only when the Gelman-Rubin statistic for every estimated parameter is below 1.2 and the traceplots for these parameter show the random fluctuations illustrated in Figure 6.4 can we be fairly certain that convergence was reached.

In cases where the Gibbs-sampler does not converge, indicated by either Gelman-Rubin statistics above 1.2, traceplots that show something other than random fluctuations around a steady mean, or both, this could be due to very high correlation between successive draws from the Gibbs-sampler. To check if this is the case applied researchers can use the *auto.corr* command on the MCMC object created for the traceplots. For the fixed mean AR(1) model this would involve running the line,

```
autocorr.plot(FixedAR.mcmc).
```

The outcome of this line of code is presented in Figure 6.5. The autocorrelation plot shows the the amount of autocorrelations between draws on the y-axis, while the number of iterations between those draws is shown on the x-axis. Ideally, the plot should show autocorrelation that drops to 0 quickly (as is the case in Figure 6.5). If this is not the case, then autocorrelation could be the cause of the non-convergence. It is then recommended that researchers first increase the value of the *n.thin* argument so that intermediate draws form the Gibbs-sampler are discarded and the correlations between the draws that are kept are smaller than before. If increasing the thining rate does not help, one should either use a reparametrization of the model (Gelman, Carlin, Stern, & Rubin, 2004, pg. 302-305), or use a different model altogether.

Fig. 6.4: Example of Traceplots for a Converged Model.

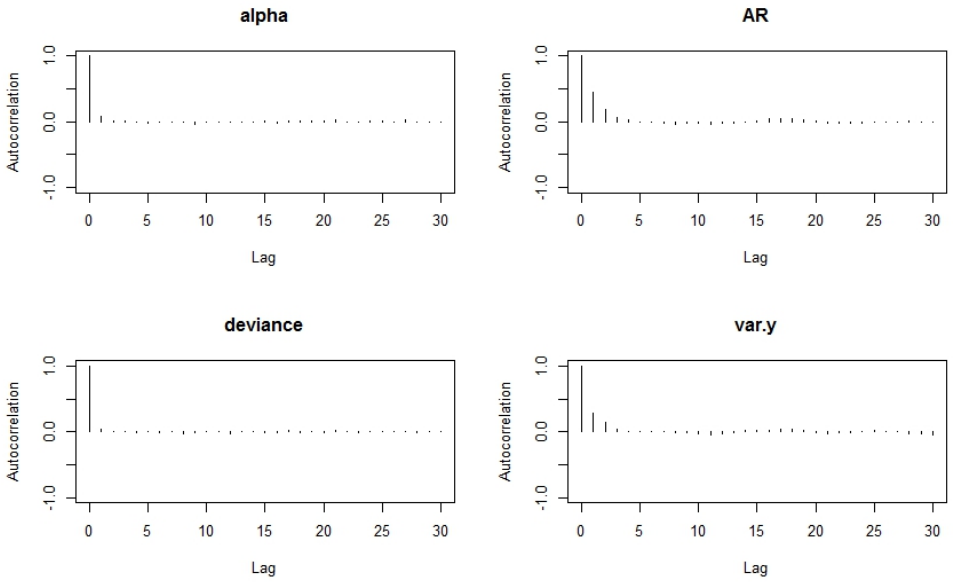


Finally, our approach will also provide the following output for the DIC (Spiegelhalter et al., 2002),

```
DIC info (using the rule, pD = var(deviance)/2)
pD = [pD Value for the Selected Model] and DIC = [DIC Value for the
  Selected Model]
DIC is an estimate of expected predictive error (lower deviance is
better).
```

Similar to other Information Criteria, like the AIC (Akaike, 1973) and BIC (Schwarz, 1978), the DIC (Spiegelhalter et al., 2002) is a measure of model (mis)fit based on a trade-off between model fit and model complexity. Here, model complexity is expressed as the number of effective model parameters by pD (Gelman, Carlin, Stern, & Rubin, 2004, pg. 182). Lower values on the DIC imply a 'better' model fit, and, as a rule of thumb, differences in DIC values larger than 5 are usually considered relevant.

Fig. 6.5: Example of Autocorrelation plots for a Converged Model.



6.9 Appendix A: R-Code

The following R-code can be run in it entirety. However, given the importance of carefully monitoring convergence of the different models, I recommend running the code for one model at a time.

```
################### Reading in and preparing Data ##############
Yrepeated <- read.csv("[Path To File]", sep = "", header = TRUE,
                      na.strings = "999999.00")
Yrepeated <- as.matrix(Yrepeated)


########## Optional Code in Case Data is in Long format #######
YWidth <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=6)
D1 <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=1)
D2 <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=1)
Rec <- matrix(NA,nrow=(nrow(Yrepeated))/6,ncol=6)
ind <- unique(Yrepeated[,1])

for (i in 1:(nrow(Yrepeated)/6)){
```

```
YWidth[i,] <- Yrepeated[Yrepeated[,1]==ind[i],4]
}

for (i in 1:(nrow(Yrepeated)/6)){
D1[i,] <- mean(Yrepeated[Yrepeated[,1]==ind[i],2])
}

for (i in 1:(nrow(Yrepeated)/6)){
D2[i,] <- mean(Yrepeated[Yrepeated[,1]==ind[i],3])
}

for (i in 1:(nrow(Ypim)/6)){
    REC[i,] <- Ypim[Ypim[,1]==ind[i],5]
}

Yrec <- rowMeans(REC)

Yrepeated <-  matrix(cbind(ind,D1,D2,YWidth,Yrec),nrow=
              length(ind))

#####################################################################

library(R2jags)

################ AR(1)Fixed Mean and AR ######################

initsFixedMuAR <- function(){
  list(alpha=rnorm(1), AR=runif(1,0,1))
  list(alpha=rnorm(1), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

FixedMuAR <- jags(data = list("DV","T","N"), initsFixedMuAR,
             model.file="[Path To File]", parameters.to.save=
             c("mu","var.y", "AR"), n.chains =2, n.iter = Itt,
             n.burnin=Bi,n.thin=1, DIC=TRUE).

FixedMuAR <- autojags(FixedMuAR, n.iter = Itt, n.thin=1,
 n.update= 10, DIC=T)

FixedMuAR.mcmc <- as.mcmc(FixedMuAR)
```

```
traceplot(FixedMuAR.mcmc)
autocorr.plot(FixedMuAR.mcmc)

################ AR(1)Random Mean and Fixed AR ###############
initsRandomMuFixedAR <- function(){
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

FixedMuRandomAR <- jags(data = list("DV","T","N"),
   initsRandomMuFixedAR,
                  model.file="[Path To File]",
                  parameters.to.save= c("mean.mu","var.y",
                  "AR"), n.chains =2, n.iter = Itt,
                  n.burnin=Bi, n.thin=1, DIC=T)


FixedMuRandomAR <- autojags(FixedMuRandomAR, n.iter = Itt,
       n.thin=1, n.update= 10,DIC=T)

FixedMuRandomAR.mcmc <- as.mcmc(FixedMuRandomAR)
traceplot(FixedMuRandomAR.mcmc)
autocorr.plot(FixedMuRandomAR.mcmc)

############# AR(1) Fixed Mean and Random AR #################
initsFixedMuRandomAR <- function(){
  list(alpha=rnorm(1), mean.AR=runif(1,0,1))
  list(alpha=rnorm(1), mean.AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

FixedMuRandomAR <- jags(data = list("DV","T","N"),
   initsFixedMuRandomAR,
                   model.file="[Path To File]",
                   parameters.to.save= c("mu","var.y",
                   "AR"), n.chains =2, n.iter = Itt,
                   n.burnin=Bi,n.thin=1, DIC=TRUE).
```

```
FixedMuRandomAR <- autojags(FixedAR, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

FixedMuRandomAR.mcmc <- as.mcmc(FixedMuRandomAR)
traceplot(FixedMuRandomAR.mcmc)
autocorr.plot(FixedMuRandomAR.mcmc)

######################### Random AR ############################
initsRandomAR <- function(){
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
  list(mean.mu=rnorm(1), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]

RandomAR <- jags(data = list("DV","T","N"), initsRandomAR,
            model.file="[Path To File]", parameters.to.save=
            c("mean.mu","var.y", "AR"), n.chains =2, n.iter =
            Itt, n.burnin=Bi, n.thin=1, DIC=T)


RandomAR <- autojags(RandomAR, n.iter = Itt, n.thin=1,
n.update= 10,DIC=T)

RandomAR.mcmc <- as.mcmc(RandomAR)
traceplot(RandomAR.mcmc)
autocorr.plot(RandomAR.mcmc)

############## LGC Fixed Intercept and Slope #################
initsLGCFMS <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1, mean=
  1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1, mean=
  1000,sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
```

```
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCFMS <- jags(data = list("DV","T","N","Occasion"),
  initsLGCFMS, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta"),
  n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
  DIC=T)

LGCFMS <- autojags(LGCFMS, n.iter = Itt, n.thin=50,
  n.update= 10, DIC=T)

LGCFMS.mcmc <- as.mcmc(LGCFMS)
traceplot(LGCFMS.mcmc)
autocorr.plot(LGCFMS.mcmc)

########## LGC Random Intercept, Fixed and Slope #############
initsLGCRMFS <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCRMFS <- jags(data = list("DV","T","N","Occasion"),
   initsLGCRMFS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y", "beta"),
   n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)


LGCRMFS <- autojags(LGCRMFS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

LGCRMFS.mcmc <- as.mcmc(LGCRMFS)
traceplot(LGCRMFS.mcmc)
```

```
autocorr.plot(LGCRMFS.mcmc)

########## LGC Fixed Intercept, Random and Slope #############
initsLGCFMRS <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCFMRS <- jags(data = list("DV","T","N","Occasion"),
   initsLGCFMRS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "mean.beta"),
   n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
   DIC=T)


LGCFMRS <- autojags(LGCFMRS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

LGCFMRS.mcmc <- as.mcmc(LGCFMRS)
traceplot(LGCFMRS.mcmc)
autocorr.plot(LGCFMRS.mcmc)

############## LGC Random Intercept and Slope ###############
initsLGCRMRS <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=
  rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), mean.beta=
  rnorm(1, mean=1000,sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
```

```
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

LGCRMRS <- jags(data = list("DV","T","N","Occasion"),
           initsLGCRMRS, model.file="[Path To File]",
           parameters.to.save=c("mean.alpha","var.y",
           "mean.beta"), n.chains =2, n.iter = Itt,
           n.burnin=Bi, n.thin=1, DIC=T)


LGCRMRS <- autojags(LGCRMRS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)

LGCRMRS.mcmc <- as.mcmc(LGCRMRS)
traceplot(LGCRMRS.mcmc)
autocorr.plot(LGCRMRS.mcmc)

####### QGC Fixed Intercept, Slope and Quadratic Term ########
initsQGCFMFSFSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
       mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
       mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
     Occasions]

QGCFMFSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMFSFSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta",
  "eta"), n.chains =2, n.iter = Itt, n.burnin=Bi,
  n.thin=1, DIC=T)

QGCFMFSFSQ <- autojags(QGCFMFSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMFSFSQ.mcmc <- as.mcmc(QGCFMFSFSQ)
```

```
traceplot(QGCFMFSFSQ.mcmc)
autocorr.plot(QGCFMFSFSQ.mcmc)

#### QGC Random Intercept, Fixed Slope and Quadratic Term ####
initsQGCRMFSFSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMFSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMFSFSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "beta", "eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCRMFSFSQ <- autojags(QGCRMFSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMFSFSQ.mcmc <- as.mcmc(QGCRMFSFSQ)
traceplot(QGCRMFSFSQ.mcmc)
autocorr.plot(QGCRMFSFSQ.mcmc)

#### QGC Random Slope, Fixed Intercept and Quadratic Term ####
initsQGCFMRSFSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
```

```
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMRSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMRSFSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y",
  "mean.beta", "eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCFMRSFSQ <- autojags(QGCFMRSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMRSFSQ.mcmc <- as.mcmc(QGCFMRSFSQ)
traceplot(QGCFMRSFSQ.mcmc)
autocorr.plot(QGCFMRSFSQ.mcmc)

#### QGC Random Quadratic Term, Fixed Slope and Intercept ####
initsQGCFMFSRSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMFSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMFSRSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y", "beta",
  "mean.eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCFMFSRSQ <- autojags(QGCFMFSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCFMFSRSQ.mcmc <- as.mcmc(QGCFMFSRSQ)
traceplot(QGCFMFSRSQ.mcmc)
```

```
autocorr.plot(QGCFMFSRSQ.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
initsQGCRMRSFSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMRSFSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMRSFSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "mean.beta", "eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSFSQ <- autojags(QGCRMRSFSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMRSFSQ.mcmc <- as.mcmc(QGCRMRSFSQ)
traceplot(QGCRMRSFSQ.mcmc)
autocorr.plot(QGCRMRSFSQ.mcmc)

#### QGC Random Intercept and Quadratic Term, Fixed Slope ####
initsQGCRMFSRSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
```

```
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMFSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMFSRSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "beta", "mean.eta"), n.chains =2, n.iter = Itt,
  n.burnin=Bi, n.thin=1, DIC=T)

QGCRMFSRSQ <- autojags(QGCRMFSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMFSRSQ.mcmc <- as.mcmc(QGCRMFSRSQ)
traceplot(QGCRMFSRSQ.mcmc)
autocorr.plot(QGCRMFSRSQ.mcmc)

#### QGC Random Slope and Quadratic Term, Fixed Intercept ####
initsQGCFMRSRSQ <- function(){
  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

  list(alpha=rnorm(1, mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), mean.eta=rnorm(1, mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCFMRSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCFMRSRSQ, model.file="[Path To File]",
  parameters.to.save= c("alpha","var.y",
  "mean.beta", "mean.eta"), n.chains =2,
  n.iter = Itt, n.burnin=Bi, n.thin=1,
              DIC=T)

QGCFMRSRSQ <- autojags(QGCFMRSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)
```

```
QGCFMRSRSQ.mcmc <- as.mcmc(QGCFMRSRSQ)
traceplot(QGCFMRSRSQ.mcmc)
autocorr.plot(QGCFMRSRSQ.mcmc)

####### QGC Random Intercept, Slope and Quadratic Term #######
initsQGCRMRSRSQ <- function(){
  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), mean.eta=rnorm(1,
  mean=1000, sd=100))

  list(mean.alpha=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), mean.eta=rnorm(1,
  mean=1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

QGCRMRSRSQ <- jags(data = list("DV","T","N","Occasion"),
  initsQGCRMRSRSQ, model.file="[Path To File]",
  parameters.to.save= c("mean.alpha","var.y",
  "mean.beta", "mean.eta"), n.chains =2,
  n.iter = Itt, n.burnin=Bi, n.thin=1,
              DIC=T)

QGCRMRSRSQ <- autojags(QGCRMRSRSQ, n.iter = Itt, n.thin=1,
  n.update= 10, DIC=T)

QGCRMRSRSQ.mcmc <- as.mcmc(QGCRMRSRSQ)
traceplot(QGCRMRSRSQ.mcmc)
autocorr.plot(QGCRMRSRSQ.mcmc)

################### ALT Fixed Alpha and Beta #################
initsALTFAFB <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}
```

```
DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFAFB <- jags(data = list("DV","T","N","Occasion"),
   initsALTFAFB, model.file="[Path To File]",
   parameters.to.save= c("intercept","var.y", "slope",
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTFAFB <- autojags(ALTFAFB, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTFAFB.mcmc <- as.mcmc(ALTFAFB)
autocorr.plot(ALTFAFB.mcmc)
traceplot(ALTFAFB.mcmc)

################## ALT Fixed Alpha, Random Beta ###############
initsALTRAFB <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRAFB <- jags(data = list("DV","T","N","Occasion"),
   initsALTRAFB, model.file="[Path To File]",
   parameters.to.save= c("mean.intercept","var.y",
   "slope","AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTRAFB <- autojags(ALTRAFB, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)
```

```
ALTRAFB.mcmc <- as.mcmc(ALTRAFB)
autocorr.plot(ALTRAFB.mcmc)
traceplot(ALTRAFB.mcmc)

################## ALT Random Alpha, Fixed Beta ###############
initsALTFARB <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFARB <- jags(data = list("DV","T","N","Occasion"),
   initsALTFARB, model.file="[Path To File]",
   parameters.to.save=c("mean.intercept","var.y",
   "mean.slope","AR"),n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTFARB <- autojags(ALTFARB, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTFARB.mcmc <- as.mcmc(ALTFARB)
autocorr.plot(ALTFARB.mcmc)
traceplot(ALTFARB.mcmc)

################### ALT Random Alpha and Beta ################
initsALTRARB <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1
  ,mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}
```

```
DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRARB <- jags(data = list("DV","T","N","Occasion"),
   initsALTRARB, model.file="[Path To File]",
   parameters.to.save= c("mean.intercept","var.y",
   "mean.slope","AR"), n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTRARB <- autojags(ALTRARB, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTRARB.mcmc <- as.mcmc(ALTRARB)
autocorr.plot(ALTRARB.mcmc)
traceplot(ALTRARB.mcmc)

################ ALT Fixed Intercept and Slope ################
initsALTFMFS <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=
  1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,mean=
  1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFMFS <- jags(data = list("DV","T","N","Occasion"),
   initsALTFMFS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "beta","AR"),
           n.chains =2, n.iter = Itt, n.burnin=Bi, n.thin=1,
           DIC=T)

ALTFMFS <- autojags(ALTFMFS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)
```

```
ALTFMFS.mcmc <- as.mcmc(ALTFMFS)
autocorr.plot(ALTFMFS.mcmc)
traceplot(ALTFMFS.mcmc)

########## ALT Random Intercept, Fixed and Slope #############
initsALTRMFS <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100), beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRMFS <- jags(data = list("DV","T","N","Occasion"),
   initsALTRMFS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y", "beta",
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTRMFS <- autojags(ALTRMFS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTRMFS.mcmc <- as.mcmc(ALTRMFS)
autocorr.plot(ALTRMFS.mcmc)
traceplot(ALTRMFS.mcmc)

########## ALT Fixed Intercept, Random and Slope #############
initsALTFMRS <- function(){
  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
```

```
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTFMRS <- jags(data = list("DV","T","N","Occasion"),
   initsALTFMRS, model.file="[Path To File]",
   parameters.to.save= c("alpha","var.y", "mean.beta",
   "AR"), n.chains =2, n.iter = Itt, n.burnin=Bi,
   n.thin=1, DIC=T)

ALTFMRS <- autojags(ALTFMRS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)


ALTFMRS.mcmc <- as.mcmc(ALTFMRS)
autocorr.plot(ALTFMRS.mcmc)
traceplot(ALTFMRS.mcmc)

############## ALT Random Intercept and Slope ################
initsALTRMRS <- function(){
  list(mean.alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))

  list(mean.alpha=rnorm(1,mean=1000, sd=100), mean.beta=rnorm(1,
  mean=1000, sd=100), AR=runif(1,0,1))
}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]

ALTRMRS <- jags(data = list("DV","T","N","Occasion"),
   initsALTRMRS, model.file="[Path To File]",
   parameters.to.save= c("mean.alpha","var.y",
   "mean.beta","AR"),n.chains =2, n.iter = Itt,
   n.burnin=Bi, n.thin=1, DIC=T)

ALTRMRS <- autojags(ALTRMRS, n.iter = Itt, n.thin=1,
   n.update= 10, DIC=T)
```

```
ALTRMRS.mcmc <- as.mcmc(ALTRMRS)
autocorr.plot(ALTRMRS.mcmc)
traceplot(ALTRMRS.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
############### Group as Predictor for Intercept ##############
initsQGCRMRSFSQPredM <- function(){
  list(gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=
  1000, sd=100))

  list(gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  mean.beta=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=
  1000, sd=100))

}

DV <- [Columns of the Data Matrix That Contain the DV]
T <- [Number of Repeated Measures]
N <- [Total Sample Size]
Occasion <- [Indicator Variable for the Different Measurement
 Occasions]


QGCRMRSFSQPredM <- jags(data = list("DV","T","N","Occasion",
   "D1", "D2"), initsQGCRMRSFSQPredM,
   model.file="[Path To File]",
                  parameters.to.save=c("gamma00","gamma10",
                  "gamma20", "var.y", "Cov", "mean.beta",
                  "eta"), n.chains =2, n.iter = Itt,
                  n.burnin=Bi, n.thin=1, DIC=T)


QGCRMRSFSQPredM <- autojags(QGCRMRSFSQPredM, n.iter = Itt,
   n.thin=1, n.update= 10, DIC=T)


QGCRMRSFSQPredM.mcmc <- as.mcmc(QGCRMRSFSQPredM)
autocorr.plot(QGCRMRSFSQPredM.mcmc)
traceplot(QGCRMRSFSQPredM.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
```

```
################## Group as Predictor for Slope ###############
initsQGCRMRSFSQPredS <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  mean.alpha=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=
  1000, sd=100))


  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  mean.alpha=rnorm(1, mean=1000, sd=100), eta=rnorm(1, mean=
  1000, sd=100))

}

QGCRMRSFSQPredS <- jags(data = list("DV","T","N","Occasion",
   "D1", "D2"), initsQGCRMRSFSQPredS,
   model.file="[Path To File]",
                  parameters.to.save=c("gamma01","gamma11",
                  "gamma21", "var.y", "Cov", "mean.alpha",
                  "eta"), n.chains =2, n.iter = Itt,
                  n.burnin=Bi, n.thin=1, DIC=T)


QGCRMRSFSQPredS <- autojags(QGCRMRSFSQPredS, n.iter = Itt,
   n.thin=1, n.update= 10, DIC=T)


QGCRMRSFSQPredS.mcmc <- as.mcmc(QGCRMRSFSQPredS)
autocorr.plot(QGCRMRSFSQPredS.mcmc)
traceplot(QGCRMRSFSQPredS.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
######### Group as Predictor for Intercept and Slope #########
initsQGCRMRSFSQPredMS <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1, mean=
  1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1, mean=
```

```
  1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100))

}

QGCRMRSFSQPredMS <- jags(data = list("DV","T","N","Occasion",
    "D1", "D2"), initsQGCRMRSFSQPredMS,
    model.file="[Path To File]",
                    parameters.to.save=c("gamma00","gamma10",
                    "gamma20", "gamma01","gamma11", "gamma21",
                    "var.y", "Cov", "eta", "PredCon","PredST",
                    "PredCCT"),n.chains =2, n.iter = Itt,
                    n.burnin=Bi, n.thin=1, DIC=T)


QGCRMRSFSQPredMS <- autojags(QGCRMRSFSQPredMS, n.iter = Itt,
    n.thin=1, n.update= 10, DIC=T)

QGCRMRSFSQPredMS.mcmc <- as.mcmc(QGCRMRSFSQPredMS)
autocorr.plot(QGCRMRSFSQPredMS.mcmc)
traceplot(QGCRMRSFSQPredMS.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
########################## Lognormal Data ######################
DV <- ((-1*DV)+(lambda/6))
DV <- DV + .0000001
DV <- log(DV)


initsQGCRMRSFSQPredMSlog <- function(){
  list(gamma01=rnorm(1, mean=6.91, sd=4.61), gamma11=rnorm(1,
   mean=6.91, sd=4.61), gamma21=rnorm(1, mean=6.91, sd=4.61),
   gamma00=rnorm(1, mean=6.91, sd=4.61), gamma10=rnorm(1,
   mean=6.91, sd=4.61), gamma20=rnorm(1, mean=6.91, sd=4.61),
   eta=rnorm(1, mean=6.91, sd=4.61))

  list(gamma01=rnorm(1, mean=6.91, sd=4.61), gamma11=rnorm(1,
  mean=6.91, sd=4.61), gamma21=rnorm(1, mean=6.91, sd=4.61),
  gamma00=rnorm(1, mean=6.91, sd=4.61), gamma10=rnorm(1,
  mean=6.91, sd=4.61), gamma20=rnorm(1, mean=6.91, sd=4.61),
  eta=rnorm(1, mean=6.91, sd=4.61))
  }

QGCRMRSFSQPredMSlog <- jags(data = list("DV","T","N","Occasion",
```

```
   "D1", "D2"), initsQGCRMRSFSQPredMSlog,
   model.file="[Path To File]",
                        parameters.to.save=c("gamma00","gamma10",
                        "gamma20", "gamma01","gamma11",
                        "gamma21", "var.y", "Cov", "eta"),
                        n.chains =2, n.iter = Itt, n.burnin=Bi,
                        n.thin=1, DIC=T)

QGCRMRSFSQPredMSlog <- autojags(QGCRMRSFSQPredMSlog,
   n.iter = Itt, n.thin=1, n.update= 10,
   DIC=T)


QGCRMRSFSQPredMSlog.mcmc <- as.mcmc(QGCRMRSFSQPredMSlog)
autocorr.plot(QGCRMRSFSQPredMSlog.mcmc)
traceplot(QGCRMRSFSQPredMSlog.mcmc)

#### QGC Random Intercept and Slope, Fixed Quadratic Term ####
################### Gamma Distributed Data ####################
DV <- ((-1*DV)+(lambda/6))
DV <- DV + .0000001
Group <- (D1+(2*D2)) + 1


initsQGCRMRSFSQPredMSgamma <- function(){
  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100),r=rep(30,3))

  list(gamma01=rnorm(1, mean=1000, sd=100), gamma11=rnorm(1,
  mean=1000, sd=100), gamma21=rnorm(1, mean=1000, sd=100),
  gamma00=rnorm(1, mean=1000, sd=100), gamma10=rnorm(1,
  mean=1000, sd=100), gamma20=rnorm(1, mean=1000, sd=100),
  eta=rnorm(1, mean=1000, sd=100),r=rep(30,3))

}


QGCRMRSFSQPredMSgamma <- jags(data = list("DV","T","N",
"Occasion", "D1", "D2", "Group"),
                        initsQGCRMRSFSQPredMSgamma,
                        model.file="[Path To File]",
```

```
                          parameters.to.save=c("gamma00",
                          "gamma10", "gamma20", "gamma01",
                          "gamma11", "gamma21", "Cov", "eta"),
                          n.chains =2, n.iter = Itt,
                          n.burnin=Bi, n.thin=1, DIC=T)

QGCRMRSFSQPredMSgamma <- autojags(QGCRMRSFSQPredMSgamma,
 n.iter = Itt, n.thin=1, n.update= 10,
 DIC=T)


QGCRMRSFSQPredMSgamma.mcmc <- as.mcmc(QGCRMRSFSQPredMSgamma)
autocorr.plot(QGCRMRSFSQPredMSgamma.mcmc)
traceplot(QGCRMRSFSQPredMSgamma.mcmc)
```

6.10 Appendix B: JAGS Models

6.10.1 AR(1) Fixed Mean and AR

```
model{

for (i in 1:N) {

DV[i,1] ~dnorm(mu, tau1)

for(t in 2:T){
  DV[i,t] ~dnorm(Mean[i,t] , tau2)
  Mean[i,t] <- cnst + AR*DV[i,t-1]
}
}

AR ~ dnorm(0, .000000000001)
mu ~ dnorm(0, .000000000001)
cnst <- mu*(1-AR)

tau2 ~ dgamma (0.000001, 0.000001)
tau1 <- (1-(pow(AR,2)))*tau2

var.y <- 1/tau2
}
```

6.10.2 AR(1) Random Mean and Fixed AR

```
model{

for (i in 1:N) {

mu[i] ~ dnorm(mean.mu, tau.mu)
cnst[i] <- mu[i]*(1-AR)
DV[i,1] ~ dnorm(mu[i], tau1)

for(t in 2:T){
  DV[i,t] ~dnorm(Mean[i,t] , tau2)
  Mean[i,t] <- cnst[i] + AR*DV[i,t-1]

}
}

AR ~ dnorm(0, .000000000001)
mean.mu ~ dnorm(0, .000000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau1 <- (1-(pow(AR,2)))*tau2
tau.mu ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
}
```

6.10.3 AR(1) Fixed Mean and Random AR

```
model{

for (i in 1:N) {

AR[i] ~ dnorm(mean.AR, tau.AR)
cnst[i] <- mu*(1-AR[i])
tau1[i] <- (1-(pow(AR[i],2)))*tau2

DV[i,1] ~dnorm(mu, tau1[i])

for(t in 2:T){
  DV[i,t] ~dnorm(Mean[i,t] , tau2)
  Mean[i,t] <- cnst[i] + AR[i]*DV[i,t-1]
}
```

```
}

mu ~ dnorm(0, .000000000001)
tau2 ~ dgamma (0.000001, 0.000001)

mean.AR ~ dnorm(0, .000000000001)
tau.AR ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
}
```

6.10.4 AR(1) Random Mean and Random AR

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
mu[i] <-LS[i,1]
AR[i] <- LS[i,2]

cnst[i] <- mu[i]*(1-AR[i])
tau1[i] <- (1-(pow(AR[i],2)))*tau2

DV[i,1] ~ dnorm(mu[i], tau1[i])

for(t in 2:T){
  DV[i,t] ~dnorm(Mean[i,t] , tau2)
  Mean[i,t] <- cnst[i] + AR[i]*DV[i,t-1]

}
}

mean.mu ~ dnorm(0, .0000000001)
mean.AR ~ dnorm(0, .0000000001)
MU[1]<- mean.mu
MU[2] <- mean.AR

tau2 ~ dgamma (0.000001, 0.000001)

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- .01
```

```
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])

var.y <- 1/tau2
}
```

6.10.5 LGC Fixed Intercept and Slope

```
model{
for (i in 1:N) {

for(t in 1:T){

DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha + beta*(Occasion[t]-1)

}
}

alpha ~ dnorm(0, .0000000001)
beta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.6 LGC Random Intercept, Fixed Slope

```
model{

for (i in 1:N) {

alpha[i] ~ dnorm(mean.alpha, tau.alpha)

for(t in 1:T){

DV[i,t] ~ dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta*(Occasion[t]-1)

}
}
```

```
beta ~ dnorm(0, .0000000001)
mean.alpha ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau.alpha ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.7 LGC Fixed Intercept, Random Slope

```
model{

for (i in 1:N) {

beta[i] ~ dnorm(mean.beta, tau.beta)

for(t in 1:T){

DV[i,t] ~ dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha + beta[i]*(Occasion[t]-1)

}
}

alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau.beta ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.8 LGC Random Intercept and Slope

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]
```

```
for(t in 1:T){

DV[i,t] ~ dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1)

}
}

mean.alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.beta

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)

R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])

var.y <- 1/tau2
}
```

6.10.9 QGC Fixed Intercept, Slope, and Quadratic Term

```
model{

for (i in 1:N) {

for(t in 1:T){

DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha + beta*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

alpha ~ dnorm(0, .0000000001)
```

```
beta ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2

}
```

6.10.10 QGC Random Intercept, Fixed Slope and Quadratic Term

```
model{

for (i in 1:N) {

alpha[i] ~ dnorm(mean.alpha, tau.alpha)

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

mean.alpha ~ dnorm(0, .0000000001)
beta ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau.alpha ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.11 QGC Random Slope, Fixed Intercept and Quadratic Term

```
model{
for (i in 1:N) {

beta[i] ~ dnorm(mean.beta, tau.beta)

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
```

```
mu[i,t] <- alpha + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau.beta ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.12 QGC Random Quadratic Term, Fixed Intercept and Slope

```
model{

for (i in 1:N) {

eta[i] ~ dnorm(mean.eta, tau.eta)

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha + beta*(Occasion[t]-1) +
   eta[i]*pow((Occasion[t]-1),2)

}
}

mean.eta ~ dnorm(0, .0000000001)
beta ~ dnorm(0, .0000000001)
alpha ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
tau.eta ~ dgamma (0.000001, 0.000001)
var.y <- 1/tau2
}
```

6.10.13 QGC Random Intercept and Slope, Fixed Quadratic Term

```
model{
```

```
for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

mean.alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.beta
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)

R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
}
```

6.10.14 QGC Random Intercept and Quadratic Term, Fixed Slope

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
eta[i] <-LS[i,2]
```

```
for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta*(Occasion[t]-1) +
   eta[i]*pow((Occasion[t]-1),2)

}
}

mean.alpha ~ dnorm(0, .0000000001)
mean.eta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.eta
beta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
}
```

6.10.15 QGC Random Slope and Quadratic Term, Fixed Intercept

```
model{

for (i in 1:N) {
LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
beta[i] <-LS[i,1]
eta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha + beta[i]*(Occasion[t]-1) +
   eta[i]*pow((Occasion[t]-1),2)

}
}

mean.beta ~ dnorm(0, .0000000001)
```

```
mean.eta ~ dnorm(0, .0000000001)
MU[1]<- mean.beta
MU[2] <- mean.eta
alpha ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
}
```

6.10.16 QGC Random Intercept, Slope, and Quadratic Term

```
model{

for (i in 1:N) {

LS[i,1:3]~dmnorm(MU[1:3], covmat[1:3,1:3])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]
eta[i] <- LS[i,3]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta[i]*pow((Occasion[t]-1),2)

}
}

mean.alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
mean.eta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.beta
MU[3] <- mean.eta

tau2 ~ dgamma (0.000001, 0.000001)
```

```
covmat[1:3,1:3] ~ dwish(R[,],3)
R[1,1] <- 1000
R[2,2] <- 500
R[3,3] <- 100 #500
R[1,2] <- 0
R[1,3] <- 0
R[2,3] <- 0
R[2,1]<- R[1,2]
R[3,1]<- R[1,3]
R[3,2]<- R[2,3]
Cov[1:3,1:3] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
rho2 <- Cov[1,3]/sqrt(Cov[1,1]*Cov[3,3])
rho3 <- Cov[2,3]/sqrt(Cov[2,2]*Cov[3,3])
var.y <- 1/tau2
}
```

6.10.17 ALT Fixed Alpha and Beta

```
model{

for (i in 1:N) {

DV[i,1] ~dnorm(mu[i,1] , tau1)
mu[i,1] <- (1/(1-AR))*alpha - (AR/pow((1-AR),2))*beta

for(t in 2:T){
  DV[i,t] ~dnorm(mu[i,t] , tau2)
  mu[i,t] <- alpha + beta*Occasion[t-1] + AR*DV[i,t-1]

}
}

AR ~ dnorm(0, .0000000001)
alpha ~ dnorm(0, .000000000001)
beta ~ dnorm(0, .000000000001)

tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1
```

```
intercept <- ((alpha)/(1-AR)) - ((beta*AR)/pow((1-AR),2))
slope <- beta/(1-AR)
}
```

6.10.18 ALT Random Alpha, Fixed Beta

```
model{

for (i in 1:N) {

alpha[i] ~ dnorm(mean.alpha, tau.alpha)
intercept[i] <- ((alpha[i])/(1-AR)) - ((beta*AR)/pow((1-AR),2))

DV[i,1] ~dnorm(mu[i,1] , tau1)
mu[i,1] <- (1/(1-AR))*alpha[i] - (AR/pow((1-AR),2))*beta

for(t in 2:T){
  DV[i,t] ~dnorm(mu[i,t] , tau2)
  mu[i,t] <- alpha[i] + beta*Occasion[t-1] + AR*DV[i,t-1]

}



AR ~ dnorm(0, .0000000001)
mean.alpha ~ dnorm(0, .000000000001)
beta ~ dnorm(0, .000000000001)

tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)
tau.alpha ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1

mean.intercept <- mean(intercept[])
slope <- beta/(1-AR)
}
```

6.10.19 ALT Fixed Alpha, Random Beta

```
model{
```

```
for (i in 1:N) {

beta[i] ~ dnorm(mean.beta, tau.beta)
intercept[i] <- ((alpha)/(1-AR)) - ((beta[i]*AR)/pow((1-AR),2))
slope[i] <- beta[i]/(1-AR)

DV[i,1] ~dnorm(mu[i,1] , tau1)
mu[i,1] <- (1/(1-AR))*alpha - (AR/pow((1-AR),2))*beta[i]

for(t in 2:T){
  DV[i,t] ~dnorm(mu[i,t] , tau2)
  mu[i,t] <- alpha + beta[i]*Occasion[t-1] + AR*DV[i,t-1]

}
}


AR ~ dnorm(0, .0000000001)
alpha ~ dnorm(0, .000000000001)
mean.beta ~ dnorm(0, .000000000001)

tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)
tau.beta ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1

mean.intercept <- mean(intercept[])
mean.slope <- mean(slope[])
}
```

6.10.20 ALT Random Alpha and Beta

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]
intercept[i] <- ((alpha[i])/(1-AR)) - ((beta[i]*AR)/pow((1-AR),2))
```

```
slope[i] <- beta[i]/(1-AR)

DV[i,1] ~dnorm(mu[i,1] , tau1)
mu[i,1] <- (1/(1-AR))*alpha[i] - (AR/pow((1-AR),2))*beta[i]

for(t in 2:T){
  DV[i,t] ~dnorm(mu[i,t] , tau2)
  mu[i,t] <- alpha[i] + beta[i]*Occasion[t-1] + AR*DV[i,t-1]

}
}


AR ~ dnorm(0, .0000000001)
mean.alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.beta


tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])

var.y <- 1/tau2
var.y1 <- 1/tau1

mean.intercept <- mean(intercept[])
mean.slope <- mean(slope[])
}
```

6.10.21 ALT Fixed Intercept and Slope

```
model{

for (i in 1:N) {
```

```
DV[i,1] ~ dnorm(trend[i,1] , tau1)
trend[i,1] <-  alpha

for(t in 2:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
trend[i,t] <- alpha + beta*Occasion[t-1]
mu[i,t] <- trend[i,t] + AR*(DV[i,t-1] - trend[i,t-1])
}
}

AR ~ dnorm(0, .0000000001)
alpha ~ dnorm(0, .000000000001)
beta ~ dnorm(0, .000000000001)
tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1
}
```

6.10.22 ALT Random Intercept, Fixed Slope

```
model{

for (i in 1:N) {

alpha[i] ~ dnorm(mean.alpha, tau.alpha)
DV[i,1] ~ dnorm(trend[i,1] , tau1)
trend[i,1] <-  alpha[i]

for(t in 2:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
trend[i,t] <- alpha[i] + beta*Occasion[t-1]
mu[i,t] <- trend[i,t] + AR*(DV[i,t-1] - trend[i,t-1])
}
}

AR ~ dnorm(0, .0000000001)
mean.alpha ~ dnorm(0, .000000000001)
beta ~ dnorm(0, .000000000001)

tau1 <-(1-pow(AR,2))*tau2
```

```
tau2 ~ dgamma (0.000001, 0.000001)
tau.alpha ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1
}
```

6.10.23 ALT Fixed Intercept, Random Slope

```
model{

for (i in 1:N) {

beta[i] ~ dnorm(mean.beta, tau.beta)
DV[i,1] ~ dnorm(trend[i,1] , tau1)
trend[i,1] <-  alpha

for(t in 2:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
trend[i,t] <- alpha + beta[i]*Occasion[t-1]
mu[i,t] <- trend[i,t] + AR*(DV[i,t-1] - trend[i,t-1])
}
}

AR ~ dnorm(0, .0000000001)
alpha ~ dnorm(0, .000000000001)
mean.beta ~ dnorm(0, .000000000001)

tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)
tau.beta ~ dgamma (0.000001, 0.000001)

var.y <- 1/tau2
var.y1 <- 1/tau1
}
```

6.10.24 ALT Random Intercept and Slope

```
model{

for (i in 1:N) {
```

```
LS[i,1:2]~dmnorm(MU[1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

DV[i,1] ~ dnorm(trend[i,1] , tau1)
trend[i,1] <-  alpha[i]

for(t in 2:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
trend[i,t] <- alpha[i] + beta[i]*Occasion[t-1]
mu[i,t] <- trend[i,t] + AR*(DV[i,t-1] - trend[i,t-1])
}
}

AR ~ dnorm(0, .0000000001)
mean.alpha ~ dnorm(0, .0000000001)
mean.beta ~ dnorm(0, .0000000001)
MU[1]<- mean.alpha
MU[2] <- mean.beta

tau1 <-(1-pow(AR,2))*tau2
tau2 ~ dgamma (0.000001, 0.000001)

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
var.y1 <- 1/tau1
}
```

6.10.25 QGC Random Intercept and Slope, Fixed Quadratic Term: Predictor for Intercept

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[i,1:2], covmat[1:2,1:2])
MU[i,1]<- gamma00 + gamma10*D1[i] + gamma20*D2[i]
```

```
MU[i,2] <- mean.beta
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

gamma00 ~ dnorm(0, .0000000001)
gamma10 ~ dnorm(0, .0000000001)
gamma20 ~ dnorm(0, .0000000001)

mean.beta ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])

var.y <- 1/tau2

PredCon[1] <- gamma00 + (mean.beta*(0)) + (eta*(0))
PredCon[2] <- gamma00 + (mean.beta*(1)) + (eta*(1))
PredCon[3] <- gamma00 + (mean.beta*(2)) + (eta*(4))
PredCon[4] <- gamma00 + (mean.beta*(3)) + (eta*(9))
PredCon[5] <- gamma00 + (mean.beta*(4)) + (eta*(16))
PredCon[6] <- gamma00 + (mean.beta*(5)) + (eta*(25))
PredCon[7] <- gamma00 + (mean.beta*(6)) + (eta*(36))

TotCon <- PredCon[1] + PredCon[2] + PredCon[3] +
  PredCon[4] + PredCon[5] + PredCon[6] +
  PredCon[7]

PredST[1] <- (gamma00+gamma10) + (mean.beta*(0)) +
 (eta*(0))
```

```
PredST[2] <- (gamma00+gamma10) + (mean.beta*(1)) +
 (eta*(1))
PredST[3] <- (gamma00+gamma10) + (mean.beta*(2)) +
 (eta*(4))
PredST[4] <- (gamma00+gamma10) + (mean.beta*(3)) +
 (eta*(9))
PredST[5] <- (gamma00+gamma10) + (mean.beta*(4)) +
 (eta*(16))
PredST[6] <- (gamma00+gamma10) + (mean.beta*(5)) +
 (eta*(25))
PredST[7] <- (gamma00+gamma10) + (mean.beta*(6)) +
 (eta*(36))

TotST <- PredST[1] + PredST[2] + PredST[3] + PredST[4] +
 PredST[5] + PredST[6] + PredST[7]

PredCCT[1] <- (gamma00+gamma20) + (mean.beta*(0)) +
          (eta*(0))
PredCCT[2] <- (gamma00+gamma20) + (mean.beta*(1)) +
  (eta*(1))
PredCCT[3] <- (gamma00+gamma20) + (mean.beta*(2)) +
  (eta*(4))
PredCCT[4] <- (gamma00+gamma20) + (mean.beta*(3)) +
  (eta*(9))
PredCCT[5] <- (gamma00+gamma20) + (mean.beta*(4)) +
  (eta*(16))
PredCCT[6] <- (gamma00+gamma20) + (mean.beta*(5)) +
  (eta*(25))
PredCCT[7] <- (gamma00+gamma20) + (mean.beta*(6)) +
  (eta*(36))

TotCCT <- PredCCT[1] + PredCCT[2] + PredCCT[3] +
  PredCCT[4] + PredCCT[5] + PredCCT[6] +
  PredCCT[7]

STCon <- TotST - TotCon
STCCT <- TotST - TotCCT
CCTCon <-  TotCCT - TotCon

}
```

6.10.26 QGC Random Intercept and Slope, Fixed Quadratic Term: Predictor for Slope

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[i,1:2], covmat[1:2,1:2])
MU[i,1] <- mean.alpha
MU[i,2] <- gamma01 + gamma11*D1[i] + gamma21*D2[i]
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

mean.alpha ~ dnorm(0, .0000000001)
gamma01 ~ dnorm(0, .0000000001)
gamma11 ~ dnorm(0, .0000000001)
gamma21 ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2

PredCon[1] <- mean.alpha + (gamma01*(0)) + (eta*(0))
PredCon[2] <- mean.alpha + (gamma01*(1)) + (eta*(1))
PredCon[3] <- mean.alpha + (gamma01*(2)) + (eta*(4))
PredCon[4] <- mean.alpha + (gamma01*(3)) + (eta*(9))
PredCon[5] <- mean.alpha + (gamma01*(4)) + (eta*(16))
PredCon[6] <- mean.alpha + (gamma01*(5)) + (eta*(25))
PredCon[7] <- mean.alpha + (gamma01*(6)) + (eta*(36))
```

```
TotCon <- PredCon[1] + PredCon[2] + PredCon[3] +
  PredCon[4] + PredCon[5] + PredCon[6] +
  PredCon[7]

PredST[1] <- mean.alpha + ((gamma01+gamma11)*(0)) +
 (eta*(0))
PredST[2] <- mean.alpha + ((gamma01+gamma11)*(1)) +
     (eta*(1))
PredST[3] <- mean.alpha + ((gamma01+gamma11)*(2)) +
     (eta*(4))
PredST[4] <- mean.alpha + ((gamma01+gamma11)*(3)) +
 (eta*(9))
PredST[5] <- mean.alpha + ((gamma01+gamma11)*(4)) +
 (eta*(16))
PredST[6] <- mean.alpha + ((gamma01+gamma11)*(5)) +
 (eta*(25))
PredST[7] <- mean.alpha + ((gamma01+gamma11)*(6)) +
 (eta*(36))

TotST <- PredST[1] + PredST[2] + PredST[3] + PredST[4] +
 PredST[5] + PredST[6] + PredST[7]

PredCCT[1] <- mean.alpha + ((gamma01+gamma21)*(0)) +
  (eta*(0))
PredCCT[2] <- mean.alpha + ((gamma01+gamma21)*(1)) +
  (eta*(1))
PredCCT[3] <- mean.alpha + ((gamma01+gamma21)*(2)) +
  (eta*(4))
PredCCT[4] <- mean.alpha + ((gamma01+gamma21)*(3)) +
  (eta*(9))
PredCCT[5] <- mean.alpha + ((gamma01+gamma21)*(4)) +
  (eta*(16))
PredCCT[6] <- mean.alpha + ((gamma01+gamma21)*(5)) +
  (eta*(25))
PredCCT[7] <- mean.alpha + ((gamma01+gamma21)*(6)) +
  (eta*(36))

TotCCT <- PredCCT[1] + PredCCT[2] + PredCCT[3] +
  PredCCT[4] + PredCCT[5] + PredCCT[6] +
  PredCCT[7]


STCon <- TotST - TotCon
```

```
STCCT <- TotST - TotCCT
CCTCon <-  TotCCT - TotCon

}
```

6.10.27 QGC Random Intercept and Slope, Fixed Quadratic Term: Predictor for Intercept and Slope

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[i,1:2], covmat[1:2,1:2])
MU[i,1] <- gamma00 + gamma10*D1[i] + gamma20*D2[i]
MU[i,2] <- gamma01 + gamma11*D1[i] + gamma21*D2[i]
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

gamma00 ~ dnorm(0, .0000000001)
gamma10 ~ dnorm(0, .0000000001)
gamma20 ~ dnorm(0, .0000000001)
gamma01 ~ dnorm(0, .0000000001)
gamma11 ~ dnorm(0, .0000000001)
gamma21 ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
```

```
PredCon[1] <- gamma00 + (gamma01*(0)) + (eta*(0))
PredCon[2] <- gamma00 + (gamma01*(1)) + (eta*(1))
PredCon[3] <- gamma00 + (gamma01*(2)) + (eta*(4))
PredCon[4] <- gamma00 + (gamma01*(3)) + (eta*(9))
PredCon[5] <- gamma00 + (gamma01*(4)) + (eta*(16))
PredCon[6] <- gamma00 + (gamma01*(5)) + (eta*(25))
PredCon[7] <- gamma00 + (gamma01*(6)) + (eta*(36))

TotCon <- PredCon[1] + PredCon[2] + PredCon[3] +
  PredCon[4] + PredCon[5] + PredCon[6] +
  PredCon[7]

PredST[1] <- (gamma00+gamma10) + ((gamma01+gamma11)*(0)) +
 (eta*(0))
PredST[2] <- (gamma00+gamma10) + ((gamma01+gamma11)*(1)) +
 (eta*(1))
PredST[3] <- (gamma00+gamma10) + ((gamma01+gamma11)*(2)) +
 (eta*(4))
PredST[4] <- (gamma00+gamma10) + ((gamma01+gamma11)*(3)) +
 (eta*(9))
PredST[5] <- (gamma00+gamma10) + ((gamma01+gamma11)*(4)) +
 (eta*(16))
PredST[6] <- (gamma00+gamma10) + ((gamma01+gamma11)*(5)) +
 (eta*(25))
PredST[7] <- (gamma00+gamma10) + ((gamma01+gamma11)*(6)) +
 (eta*(36))

TotST <- PredST[1] + PredST[2] + PredST[3] + PredST[4] +
 PredST[5] + PredST[6] + PredST[7]

PredCCT[1] <- (gamma00+gamma20) + ((gamma01+gamma21)*(0)) +
  (eta*(0))
PredCCT[2] <- (gamma00+gamma20) + ((gamma01+gamma21)*(1)) +
  (eta*(1))
PredCCT[3] <- (gamma00+gamma20) + ((gamma01+gamma21)*(2)) +
      (eta*(4))
PredCCT[4] <- (gamma00+gamma20) + ((gamma01+gamma21)*(3)) +
  (eta*(9))
PredCCT[5] <- (gamma00+gamma20) + ((gamma01+gamma21)*(4)) +
  (eta*(16))
PredCCT[6] <- (gamma00+gamma20) + ((gamma01+gamma21)*(5)) +
  (eta*(25))
PredCCT[7] <- (gamma00+gamma20) + ((gamma01+gamma21)*(6)) +
```

```
  (eta*(36))

TotCCT <- PredCCT[1] + PredCCT[2] + PredCCT[3] +
  PredCCT[4] + PredCCT[5] + PredCCT[6] +
  PredCCT[7]

STCon <- TotST - TotCon
STCCT <- TotST - TotCCT
CCTCon <-  TotCCT - TotCon

}
```

6.10.28 Three-level QGC Model with Random Intercept and Slope, Fixed Quadratic Term

```
model{

for (j in 1:C) {

mean.alpha[j] ~ dnorm(MA, TauA)
mean.beta[j] ~ dnorm(MB, TauB)

}


for (i in 1:N) {

MU[i,1] <- mean.alpha[Center[i]]
MU[i,2] <- mean.beta[Center[i]]

LS[i,1:2]~dmnorm(MU[i,1:2], covmat[1:2,1:2])
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~dnorm(mu[i,t] , tau2)
 mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
      eta*pow((Occasion[t]-1),2)

}
}

MA ~ dnorm(0, .0000000001)
```

```
MB ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
TauA ~ dgamma (0.000001, 0.000001)
TauB ~ dgamma (0.000001, 0.000001)

VarA <- 1/TauA
VarB <- 1/TauB

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]

Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
}
```

6.10.29 Three-level QGC Model with Random Intercept and Slope, Fixed Quadratic Term: Predictor for Intercept and Slope

```
model{



for (j in 1:C) {
# These lines draw separate intercepts and regression
# coefficients for each center in the study. Thus they
# allow for different treatment means in each center.

# If one want the initial mean NB of the treatments to be
# the same across centers, the following three lines should
# be removed, and each individual should get a common gamma00,
# gamma10, and gamma20

gamma00[j] ~ dnorm(Mugamma00, Taugamma00)
gamma10[j] ~ dnorm(Mugamma10, Taugamma10)
gamma20[j] ~ dnorm(Mugamma20, Taugamma20)

# If one want the change in NB of the treatments to be the
```

```
# same across centers, the following three lines should be
# removed, and each individual should get a common gamma10,
# gamma11, and gamma21

gamma01[j] ~ dnorm(Mugamma01, Taugamma01)
gamma11[j] ~ dnorm(Mugamma11, Taugamma11)
gamma21[j] ~ dnorm(Mugamma21, Taugamma21)
}


for (i in 1:N) {

LS[i,1:2]~ dmnorm(MU[i,1:2], covmat[1:2,1:2])

MU[i,1] <- gamma00[Center[i]] + gamma10[Center[i]]*D1[i] +
           gamma20[Center[i]]*D2[i]

MU[i,2] <- gamma01[Center[i]] + gamma11[Center[i]]*D1[i] +
           gamma21[Center[i]]*D2[i]

alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]


for(t in 1:T){

DV[i,t] ~ dnorm(mu[i,t] , tau2)
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)

}
}

Mugamma00 ~ dnorm(0, .0000000001)
Mugamma10 ~ dnorm(0, .0000000001)
Mugamma20 ~ dnorm(0, .0000000001)

Mugamma01 ~ dnorm(0, .0000000001)
Mugamma11 ~ dnorm(0, .0000000001)
Mugamma21 ~ dnorm(0, .0000000001)

eta ~ dnorm(0, .0000000001)

tau2 ~ dgamma (0.000001, 0.000001)
```

```
Taugamma00 ~ dgamma (0.000001, 0.000001)
Taugamma10 ~ dgamma (0.000001, 0.000001)
Taugamma20 ~ dgamma (0.000001, 0.000001)

Taugamma01 ~ dgamma (0.000001, 0.000001)
Taugamma11 ~ dgamma (0.000001, 0.000001)
Taugamma21 ~ dgamma (0.000001, 0.000001)


Vargamma00 <- 1/Taugamma00
Vargamma10 <- 1/Taugamma10
Vargamma20 <- 1/Taugamma20

Vargamma01 <- 1/Taugamma01
Vargamma11 <- 1/Taugamma11
Vargamma21 <- 1/Taugamma21


# Note that we are using the same covariance matrix for
# people form each center in the study. If differences in
# variance between the centers is likely, we should draw
# separate R matrices the same way we draw separate gamma's
# above.

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]

Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])
var.y <- 1/tau2
}
```

6.10.30 QGC Random Intercept and Slope, Fixed Quadratic Term: Predictor for Intercept and Slope, Gamma Distributed Data

```
model{

for (i in 1:N) {

LS[i,1:2]~dmnorm(MU[i,1:2], covmat[1:2,1:2])
```

```
MU[i,1] <- gamma00 + gamma10*D1[i] + gamma20*D2[i]
MU[i,2] <- gamma01 + gamma11*D1[i] + gamma21*D2[i]
alpha[i] <-LS[i,1]
beta[i] <- LS[i,2]

for(t in 1:T){
DV[i,t] ~ dgamma(shape[i,t], r[Group[i]])
mu[i,t] <- alpha[i] + beta[i]*(Occasion[t]-1) +
   eta*pow((Occasion[t]-1),2)
shape[i,t]<- mu[i,t]*r[Group[i]]

}
}

for(k in 1:3){r[k] ~ dnorm(0, .0000000001) }

gamma00 ~ dnorm(0, .0000000001)
gamma10 ~ dnorm(0, .0000000001)
gamma20 ~ dnorm(0, .0000000001)
gamma01 ~ dnorm(0, .0000000001)
gamma11 ~ dnorm(0, .0000000001)
gamma21 ~ dnorm(0, .0000000001)
eta ~ dnorm(0, .0000000001)

covmat[1:2,1:2] ~ dwish(R[,],2)
R[1,1] <- 1000
R[2,2] <- 500
R[1,2] <- 0
R[2,1]<- R[1,2]
Cov[1:2,1:2] <- inverse(covmat[,])
rho1 <- Cov[1,2]/sqrt(Cov[1,1]*Cov[2,2])


PredCon[1] <- gamma00 + (gamma01*(0)) + (eta*(0))
PredCon[2] <- gamma00 + (gamma01*(1)) + (eta*(1))
PredCon[3] <- gamma00 + (gamma01*(2)) + (eta*(4))
PredCon[4] <- gamma00 + (gamma01*(3)) + (eta*(9))
PredCon[5] <- gamma00 + (gamma01*(4)) + (eta*(16))
PredCon[6] <- gamma00 + (gamma01*(5)) + (eta*(25))
PredCon[7] <- gamma00 + (gamma01*(6)) + (eta*(36))

PredST[1] <- (gamma00+gamma10) + ((gamma01+gamma11)*(0)) +
 (eta*(0))
PredST[2] <- (gamma00+gamma10) + ((gamma01+gamma11)*(1)) +
```

```
 (eta*(1))
PredST[3] <- (gamma00+gamma10) + ((gamma01+gamma11)*(2)) +
 (eta*(4))
PredST[4] <- (gamma00+gamma10) + ((gamma01+gamma11)*(3)) +
 (eta*(9))
PredST[5] <- (gamma00+gamma10) + ((gamma01+gamma11)*(4)) +
 (eta*(16))
PredST[6] <- (gamma00+gamma10) + ((gamma01+gamma11)*(5)) +
 (eta*(25))
PredST[7] <- (gamma00+gamma10) + ((gamma01+gamma11)*(6)) +
 (eta*(36))

PredCCT[1] <- (gamma00+gamma20) + ((gamma01+gamma21)*(0)) +
  (eta*(0))
PredCCT[2] <- (gamma00+gamma20) + ((gamma01+gamma21)*(1)) +
  (eta*(1))
PredCCT[3] <- (gamma00+gamma20) + ((gamma01+gamma21)*(2)) +
  (eta*(4))
PredCCT[4] <- (gamma00+gamma20) + ((gamma01+gamma21)*(3)) +
  (eta*(9))
PredCCT[5] <- (gamma00+gamma20) + ((gamma01+gamma21)*(4)) +
  (eta*(16))
PredCCT[6] <- (gamma00+gamma20) + ((gamma01+gamma21)*(5)) +
  (eta*(25))
PredCCT[7] <- (gamma00+gamma20) + ((gamma01+gamma21)*(6)) +
  (eta*(36))
}
```

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov & F. Caski (Eds.), *Proceedings of the second international symposium on information theory.*

Aroian, L. A. (1947). The probability function of the product of two normally distributed variables. *The Annals of Mathematical Statistics*, *18*, 265 - 271.

Aron, E., & Aron, A. (1997). Sensory-processing sensitivity and its relation to introversion and emotionality. *Journal of Personality and Social Psychology*, *73*, 345-368.

Bamelis, L., Arntz, A., Wetzelaer, P., Verdooren, R., & Evers, S. (2013). Economic evaulation of schema therapy and clarification-oriented psychotherapy for personality disorder: a multicentre random trial. *Journal of Mental Health Policy and Economics*, *16*.

Bates, D., Maechler, M., & Bolker, B. (2011). lme4: Linear mixed-effects models using s4 classes [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=lme4` (R package version 0.999375-39)

Bohrnstedt, G., & Goldberger, A. (1969). On the exact covariance of products of random variables. *Journal of the American Statistical Association*, *64*, 1439-1442.

Bollen, K., & Curran, P. (2004). Autoregressive latent trajectories (ALT) model: A synthesis of two traditions. *Sociological Methods and Research*, *32*, 336-383.

Bollen, K., & Curran, P. (2006). *Latent curve models: A structural equation perspective.* New York, NY: Wiley.

Browne, M., & Du Toit, S. H. C. (1991). Models for learning data. In L. Collins & J. L. Horn (Eds.), *Best methods for the analysis of change: Recent advances, unanswered questions, future directions* (p. 47-68). Washington, DC: American Psychological Association.

Bryk, A., & Raudenbush, S. (1992). *Hierarchical linear models.* Newbury Park, CA: Sage.

Chatfield, C. (2003). *The analysis of time series: An introduction* (6th, Ed.). Washington, DC: Chapman & Hall/CRC.

Chi, E., & Reinsel, G. (1989). Models for longitudinal data with random effects and AR(1) errors. *Journal of the American Statistical Association*, *84*, 452-459.

Curran, P. J., & Bollen, K. A. (2001). The best of both worlds: Combining autoregressive and latent curve models. In L. M. Collins & A. G. Sayer (Eds.), *New methods for the analysis of change* (p. 105-136). Washington, DC: American Psychological Association.

De Haan-Rietdijk, S., Gottman, J., Bergeman, C., & Hamaker, E. L. (2014). Get over it! a multilevel threshold autoregressive model for state-dependent affect regulation. *Psychometrika*.

Enders, C., & Tofighi, D. (2007). Centering predictor variables in cross-sectional multilevel models: A new look at an old issue. *Psychological Methods*, *12*, 121-138.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2004). *Bayesian data analysis*. New York, NY: Chapman & Hall.

Gelman, A., Su, Y.-S., Yajima, M., Hill, J., Pittau, M. G., Kerman, J., et al. (2011). arm: Data analysis using regression and multilevel/hierarchical models [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=arm` (R package version 1.4-13)

Hamaker, E. L. (2005). Conditions for the equivalence of the autoregressive latent trajectory model and a latent growth curve model with autoregressive disturbances. *Sociological Methods and Research*, *33*, 404-416.

Hamaker, E. L. (2012). Why researchers should think within-person: A paradigmatic rationale. In M. R. Mehl & T. S. Conner (Eds.), *Handbook of research methods for studying daily life*. New York, NY: Guilford Publications.

Hamaker, E. L., & Grasman, R. P. P. P. (2015). To center or not to center? investigating inertia with a multilevel autoregressive model. *Frontiers in Psychology*.

Hamaker, E. L., & Klugkist, I. (2011). Bayesian estimation of multilevel models. In K. Roberts & J. Hox (Eds.), *Handbook of advanced multilevel analysis* (p. 137-161). New York: Taylor and Francis.

Hamilton, J. (1994). *Time series analysis*. Princeton, NJ: Princeton University Press.

Hoch, J., Briggs, A., & Willan, A. (2002). Something old, something new, something borrowed, something blue: a framework for the marriage of health econometrics and cost-effectiveness analysis. *Health Economics*.

Hoijtink, H. (2009). Bayesian data analysis. In A. Millsap R.E. & Maydeu-Olivares (Ed.), *The sage handbook of quantitative methods in psychology* (p. 423-443). London: SAGE.

Hox, J. (2010). *Multilevel analysis: Techniques and applications* (2nd ed.; G. Marcoulides, Ed.). New York, NY: Routledge.

Jongerling, J., & Hamaker, E. L. (2011). On the trajectories of the predetermined ALT model: What are we really modeling? *Structural Equation Modeling: A Multidisciplinary Journal*, *18*, 370-382.

Jongerling, J., Laurenceau, J. P., & Hamaker, E. L. (2015). A multilevel AR(1) model: Allowing for inter-individual differences in trait-scores, inertia, and innovation variance. *Multivariate Behavioral Research*, *50*(3), 334-349.

Jöreskog, K. G. (1971). Estimation and testing of simplex models. *British Journal of Mathematical and Statistical Psychology*, *23*, 121-145.

Jöreskog, K. G. (1979). Statistical estimation of structural models in longitudinal-developmental investigations. In J. Nesselroade & P. Baltes (Eds.), *Longitudinal research in the study of behavior and development* (p. 303-352). New York: Academic Press.

Koval, P., & Kuppens, P. (2012). Changing emotion dynamics: individual differences in the effect of anticipatory social stress on emotional inertia. *Emotion*, *12*, 256-267.

Koval, P., Kuppens, P., Allen, N. B., & Sheeber, L. B. (2012). Getting stuck in depression: The roles of rumination and emotional inertia. *Cognition & Emotion*, *26*, 1412-1427.

Kreft, I., Leeuw, J. de, & Aiken, L. (1995). The effect of different forms of centering in hierarchical linear models. *Multivariate and Behavioral Methods*, *30*, 1-21.

Kuppens, P., Allen, N., & Sheeber, L. (2010). Emotional inertia and psychological maladjustment. *Psychological Science*, *21*, 984-991.

Kuppens, P., Sheeber, L. B., Yap, M. B., Whittle, S., Simmons, J. G., & Allen, N. B. (2012). Emotional inertia prospectively predicts the onset of depressive disorder in adolescence. *Emotion*, *12*, 283-289.

Laurenceau, J. P., Feldman Barrett, L., & Rovine, M. J. (2005). The interpersonal process model of intimacy in marriage: A daily-diary and multilevel modeling approach. *Journal of Family Psychology*, *19*, 314-323.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). Winbugs – a Bayesian modeling framework: concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325-337.

Mandys, F., Dolan, C., & Molenaar, P. (1994). Two aspects of the simplex model: Goodness of fit to linear growth curve structures and the analysis of mean trends. *Journal of Educational and Behavioural Statistics*, *19*, 201-215.

Meredith, W., & Tisak, J. (1990). Latent curve analysis. *Psychometrika*, *55*, 107-122.

Mood, A. M., Graybill, F. A., & Boes, D. C. (1985). *Introduction to the theory of statistics*. London: McGraw-Hill.

Nesselroade, J. R. (1991). Interindividual differences in intraindividual change. In L. M. Collins & J. L. Horn (Eds.), *Best methods for the analysis of change: Recent advances, unanswered questions, future directions.* Washington, D.C.: American Psychological Association.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society. Series B (Methodological)*, *57*, 99-138.

Ployhart, R. E., & Hakel, M. D. (1998). The substansive nature of performance variability: Predicting interindividual differences in intraindividual performance. *Personnel Psychology*, *51*, 859-901.

Plummer, M. (2003). Jags: A program for analysis of Bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing (dsc 2003).*

Plummer, M. (2012). Jags version 3.3.0 user manual [Computer software manual].

R Core Team. (2014). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.

Rasbash, J., Charlton, C., Browne, W., Healy, M., & Cameron, B. (2009). *MLwiN Version 2.1.* Centre for Multilevel Modelling, University of Bristol.

Rottenberg, J. (2005). Mood and emotion in major depression. *Current Directions in Psychological Science*, *14*, 167-170.

Rovine, M. J., & Walls, T. A. (2006). A multilevel autoregressive model to describe interindividual differences in the stability of a process. In J. L. Schafer & T. A. Walls (Eds.), *Models for intensive longitudinal data.* New York, NY: Oxford University Press.

Schmittmann, V. D., Dolan, C. V., Maas, H. L. J. Van der, & Neale, M. C. (2005). Discrete latent Markov models for normally distributed response data. *Multivariate Behavioral Research*, *40*, 461-488.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461-464.

Snijders, T., & Bosker, R. (1994). Modeled variance in two-level models. *Sociological Methods and Research*, 342-363.

Spiegelhalter, D., Best, N., Carlin, B., & Linde, A. v. d. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *64(4)*, 583-639.

Stevens, J., O'Hagan, A., & Miller, P. (2003). Case study in the Bayesian analysis of a cost-effectiveness trial in the evaluation of health care technologies: Depression. *Pharmaceutical Statistics*.

Sturtz, S., Ligges, U., & Gelman, A. (2005). R2WinBUGS: A package for running WinBUGS from R. *Journal of Statistical Software*, *12*, 1-16. Retrieved from `http://www.jstatsoft.org`

Suls, J., Green, P., & Hillis, S. (1998). Emotional reactivity to everyday problems, affective inertia, and neuroticism. *Personality and Social Psychology Bulletin*, *24*, 127-136.

Thompson, S., Nixon, R., & Grieve, R. (2006). Addressing the issues that arise in analysing multicentre cost data, with application to a multinational study. *Journal of Health Economics*.

Wang, L. P., Hamaker, E. L., & Bergeman, C. S. (2012). Investigating inter-individual differences in short-term intra-individual variability. *Psychological Methods*, *17*, 567-581.

Watson, D., & Clark, L. A. (1999). The PANAS-X: Manual for the positive and negative affect schedule - expanded form [Computer software manual].

Wetzelaer, P., Jongerling, J., Arntz, A., & Evers, S. (2015). Bayesian net benefit regression on longitudinal cost-effectiveness data from a multicentre rct. (not yet submitted)

Wichers, M., Barge-Schaapman, D. Q. C. M., Nicolson, N. A., Peeters, F., Vries, M. de, Mengelers, R., et al. (2009). Reduced stress-sensitivity or increased reward experience: The psychological mechanism of response to antidepressant medication. *Neuropsychopharmacology*, *34*, 923-931.

Wichers, M., Lothmann, C., Simons, C. J. P., Nicolson, N. A., & Peeters, F. (2012). The dynamic interplay between negative and positive emotions in daily life predicts response to treatment in depression: A momentary assessment study. *British Journal of Clinical Psychology*, *51*, 206-222.

Wichers, M., Peeters, F., Geschwind, N., Jacobs, N., Simons, C. J. P., Derom, C., et al. (2010). Unveiling patterns of affective responses in daily life may improve outcome prediction in depression: A momentary assessment study. *Journal of Affective Disorders*, *124*, 191-195.

Wichers, M., Peeters, F., Rutten, B. P. F., Jacobs, N., Derom, C., Thiery, E., et al. (2012). A time-lagged momentary assessment study on daily life physical activity and affect. *Health Psychology*, *31*, 135-144.

Yu-Sung, S., & Masanao, Y. (2014). R2jags: A package for running jags from r [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=R2jags` (R package version 0.04-01)

Zyphur, M., Chaturverdi, S., & Arvey, R. (2008). Job performance over time is a function of latent trajectories and previous performance. *Journal of Applied Psychology*, *93*, 217-224.

List of Figures

List of Tables

Summary

This thesis focused on modeling inter-individual differences in both stable- and developmental processes, where stable processes are characterized by within-person reversible variability over time in the absence of a gross underlying trend (Nesselroade, 1991), and developmental processes are characterized by structural change over time, with intra-individual variability occurring around an (individual's) mean trend or growth curve.

For the inter-individual differences in stable processes, intensively sampled data, that is, data sampled over a large number of repeated measurements (that are usually close together in time), is required and we focused on a *dynamic multilevel model* (DMM) that was based on modeling the repeated measures of an individual at level 1 using a time series model, while allowing for individual differences in the model parameters at level 2.

If intensively sampled data is not available, models focused more on systematic change trajectories can be used to gain valuable insight in the longitudinal process under study. For this non-intensively sampled data, special attention was given to the the autoregressive latent trajectory (ALT) model that was introduced by Curran and Bollen (2001) (see also Bollen & Curran, 2004), and which combines a latent growth curve (LGC) model (Meredith & Tisak, 1990; Curran & Bollen, 2001; Bollen & Curran, 2004) with an autoregressive model (Jöreskog, 1971, 1979).

Part 1: Modeling Inter-Individual differences with Intensively Sampled Data

Chapter 2 focused on the substantive interpretation of (inter-individual differences in) the parameters of a two level AR(1) model. Specifically, it was argued that the mean parameter μ_i, represents an individual's trait score (i.e., his/her long-run tendency, equilibrium, or long-term preferred state), while the AR-parameter ϕ_i can be viewed as an inertia parameter that reflects the degree to which previous scores or states carry over into current scores or states. In addition, the random error term ϵ_{it} represents

the part of the process that cannot be predicted based on previous scores or states. Thus, it can be thought of as the collection of all unobserved (or omitted) factors that influence the process under investigation. Since, these effects are absorbed into the innovation term, they influence its variance σ^2.

With respect to individual differences in the model parameters, the need to allow for individual differences in means is obvious: Different individuals have different trait levels or preferred states, and this can be captured by individual differences in μ_i. In addition, the importance of allowing for individual differences in the AR-parameter has been the focus of a small number of studies (Rovine & Walls, 2006; Suls et al., 1998; Wang et al., 2012; Kuppens et al., 2010; Koval et al., 2012), which have shown that this measure of inertia can be meaningfully related to other person characteristics. In addition, it has been shown that inertia can prospectively predict the onset of depression in adolescence (Kuppens et al., 2012) and health outcomes (Wang et al., 2012).

The possibility of individual differences in the innovation variance has been largely ignored in the literature however. Yet, in Chapter 2 it was argued that, from a substantive point of view, the existence of individual differences in innovation variances is to be expected for two reasons. First, there are probably individual differences in the range of fluctuation of the unobserved or omitted factors that influence the process under investigation, and this can be reflected by individual differences in the innovation variance. Second, individuals are likely to differ from each other with respect to their responsiveness to such factors. In other words, it is argued in Chapter 2 that individual differences in the innovation variance are indicative of (very likely) differential sensitivity and/or exposure to unmodelled factors, and relating these individual differences in innovation variance to other individual differences is likely to help researchers obtain more insight in the process under investigation. Furthermore, there is also a statistical motivation for including the innovation variance as a random effect. While ignoring random effects typically does not bias the estimates of the fixed effects in multilevel models (cf. Hox, 2010), Chapter 2 described an extensive simulation study that showed that when the innovation variance was erroneously modeled as fixed in a two level AR(1) model, that this lead to bias in the estimation of the AR-parameter (where the direction of the bias depended on the correlation between the innovation variance and the AR-parameter). This can be explained by the fact that the variance of an AR(1) process is a function of both the innovation variance and the AR-parameter, and when one of these is fixed across individuals, the other is the only random source that can account for individual differences in observed variance. The impact, or cost, of this bias in the AR-parameter depends on the amount of autocorrelation. The simulation study showed that under realistic circumstances the maximum bias is likely around -.12. So if the true value of the AR-parameter is far away from 0, the consequences of the bias are probably not that severe. If the true value is close to 0 however, the bias could change the estimate from positive to negative. The latter possibility is a more severe problem, as a negative AR-parameter describes a qualitatively different process than a positive one.

Finally, Chapter 2 also showed that Bayesian estimation of the two level AR(1) model is preferred over maximum likelihood (ML) estimation, because ML estimation of the model either lead to considerable bias in the estimate of the AR-parameter, or to the inability to model individual means. In fact, the only criterion on which ML estimation outperformed Bayesian estimation, was with respect to reproducing the rank order in the individual AR-parameters ϕ_i. Bayesian estimation on the other hand could model individual means with minimal bias in model parameters. Bias in the fixed effects did increases when either N or T decreased, while the bias for the random effects, which was always positive, seemed more strongly affected by N than by T, but results indicated that Bayesian estimation of the two level AR(1) model can be effectively used for making inferences even with small sample sizes such as $N = 20$ and $T = 10$. If the rank order in the individual innovation variances and the individual AR-parameters is also important, $T \geq 50$ is advised.

Chapter 3, took a more detailed look at the variance structure of the two level AR(1) model, and determined how (inter-individual differences in) the model parameters influence the total variance. Specifically, an expression for the total variance of a two level AR(1) model was derived that can be written as,

$$\sigma_y^2 \approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2,$$

where σ_y^2 denotes the total variance of the time series; μ_{σ^2} and μ_ϕ are the mean innovation (or error) variance and AR-parameter across individuals; τ_ϕ^2 and τ_μ^2 are the inter-individual variances in respectively the AR-parameter and the mean; and $\tau_{\phi\sigma^2}$ is the covariance between the AR-parameter and the innovation variance. Subsequently, this expression was used to find equations for the proportion of explained variance on level 1, the proportion of explained variance on level 2, the total proportion of explained variance, the proportion of variance explained by autocorrelation, and the Intra-Class Correllation (ICC) in a two level AR(1) model. Since the expression above contains all the model parameters of a two level AR(1) model, applied researchers can calculate these different proportions of explained variance and the ICC using only parameter estimates. This was demonstrated with an example concerning structured diary data about the positive affect of 96 married woman from the study by Laurenceau, Feldman Barrett, and Rovine (2005).

Unexpectedly, the expression for the total variance of a two level AR(1) model also revealed that random variance in the innovation variance $\tau_{\sigma^2}^2$ does not directly influence the total variance of a two level AR(1) model. Instead, inter-individual variance in the innovation variance only influences the total variance through its correlation with the AR-parameter $\tau_{\phi\sigma^2}$. This result has important implications. In Chapter 2 it was argued that individual differences in τ_σ^2 are indicative of differential sensitivity and/or exposure to unmodeled factors, and that ignoring these individual differences can lead to biased parameter estimates. Taken together with the results found here, it could now be concluded that such differential sensitivity and/or exposure does not necessarily show up in the total variance of an AR(1) model. Specifically, any inter-

individual differences in these areas that are independent of the AR-parameter will go unnoticed when only looking at the total variance of the time series. This shows that the separate detection and modeling of individual differences in innovation variances is very important.

Part 2: Modeling Inter-Individual differences with Non-Intensively Sampled Data.

Chapter 4 of the thesis examined the two "start-up" methods suggested by Curran and Bollen (2001) to deal with the recursion inherent to the ALT model, and specifically focused the effect of these two methods on the model parameters. It turned out that with both start-up methods the substantive interpretation of the parameters of the ALT model is not straightforward, and that this was especially true after using the start-up method that results in the predetermined ALT model, for which no expression can be derived for the intercept and the slope. This form of the ALT model turned out to be very flexible, and Chapter 4 showed that it could lead to unexpected trajectories. For example, a predetermined ALT model which contains no change parameter but only a constant (i.e., $\beta_i = 0, \forall i$), can generate all the growth curves in Figure 1, while a predetermined ALT model with negative mean and individual change parameters can still generate growth curves that are increasing.

Furthermore, Chapter 4 gave two examples that showed that the predetermined ALT model can adequately fit to data generated by models with model structures that are rather different from that of the ALT model itself. This was shown by generating a population covariance matrix and population means for 5 waves of data using a quadratic growth curve (QGC) model, and subsequently fitting the predetermined ALT model to these data.

Finally, Chapter 4 also gave an empirical illustration that showed that these issues with the interpretation of model parameters and model fit can occur with real data as well.

In this chapter we therefore advice researchers to either check their model predicted mean scores (by entering the appropriate parameter values into the equations of the ALT model, or by checking a plot of the model predicted scores) in order to determine the shape of the trajectory that is being modeled, or to apply the alternative "start-up" method for the ALT model that puts constraints on the model parameters at the first-measurement occasion. With this second "start-up" method the interpretation of the model parameters still is not straight forward, but at least expression for the intercept and the slope of the growth curve can be derived.

Chapters 5 and 6 of this thesis, were aimed at enabling applied researchers to analyze their own data with either the ALT model, or a number of other models suitable for the study of both stable and developmental processes.

In this context, Chapter 5 described a multi-center randomized trial cost-effectiveness study for the treatment of personality disorders, that served as an applied example

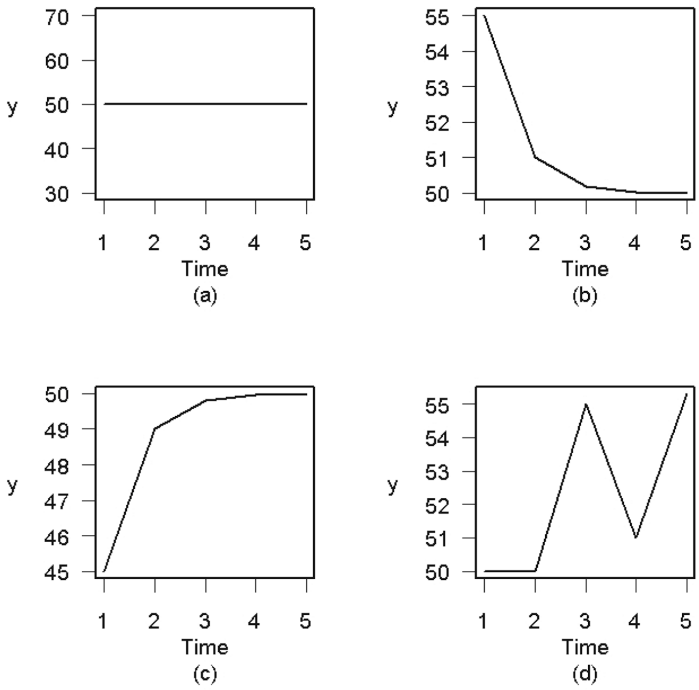


Fig. 1: Actual Growth Trajectories in an ALT model with only a latent intercept. Panel (a) shows actual (model predicted) growth trajectory for $\mu = 50$, $\mu_\alpha = 40$, and $\phi_i = \phi = 0.2$. Panel (b) shows the growth trajectory for $\mu = 55$, $\mu_\alpha = 40$, and $\phi_i = \phi = 0.2$. Panel (c) shows the growth trajectory for $\mu = 45$, $\mu_\alpha = 40$, and $\phi_i = \phi =$. Panel (d) shows the growth trajectory for $\mu = 50$, $\mu_\alpha = 40$, and $\phi_i = \phi = 0.2$ at $t = 2, 4$ and $\phi_i = \phi = 0.3$ at $t = 3, 5$.

for researchers. This chapter outlined how the best fitting model can be selected from among the set of models provided using the Deviance Information Criterion (DIC) (Spiegelhalter et al., 2002), and how the different model should be interpreted. Additionally, it proved that the new longitudinal random effects models presented in this chapter can be a useful extension to the Net-Benefit Regression (NBR) framework (Hoch et al., 2002). After determining the best fitting model through a systematic strategy, this model could subsequently be used to directly produce meaningful Cost-Effectiveness Acceptability Curves (CEAC's, for example see (Bamelis et al., 2013)) without first having to resort to additional procedures for dealing with missing data.

Finally, Chapter 6 provided applied researcher with all the materials needed for the analysis of their data with the random effect models from Chapter 5. Specifically, it provided code for the Bayesian estimation of multilevel AR(1), LGC, QGC, and

(constrained) ALT models. Based on the results of Chapter 4, the predetermined ALT model was not included, due to the fact that its parameters can not be interpreted directly. In addition, Chapter 6 provided a detailed manual that gave step-by-step instruction on how to prepare data for analysis with one of the provided models, how to run the code for the actual analysis, and how to interpret the output.

Samenvatting

Deze thesis was gericht op het modelleren van inter-individuele verschillen in zowel stabiele- als groeiprocessen, waarbij stabiele processen worden gekenmerkt door omkeerbare veranderingen binnen personen in de afwezigheid van een onderliggende trend (Nesselroade, 1991), en groeiprocessen worden gekarakteriseerd door structurele veranderingen over de tijd met omkeerbare veranderingen rondom iemand zijn/haar gemiddelde trend of groeicurve.

Voor het modelleren van inter-individuele verschillen in stabiele processen is intensief gesampelde data - oftewel data die is verzameld over een groot aantal herhaalde metingen (die veelal dicht op elkaar zitten)- nodig en hebben we ons gericht op een *dynamisch multilevel model* dat de herhaalde metingen van personen op level 1 modelleerde door middel van een tijdreeks model, en op level 2 individuele verschillen in de parameters van dit model toeliet.

Wanneer er geen intensief gesampelde data voorhanden is, kunnen modellen die meer gericht zijn op systematische veranderings-trajecten worden gebruikt om waardevolle inzichten te verkrijgen in het te bestuderen longitudinale proces. Voor het modelleren van deze niet-intensief gesampelde data werd er vooral gekeken naar het autoregressieve latente traject (ALT) model dat werd geïntroduceerd door Curran and Bollen (2001) (zie ook Bollen & Curran, 2004), en waarin een latent groeicurve (LGC) model (Meredith & Tisak, 1990; Curran & Bollen, 2001; Bollen & Curran, 2004) wordt gecombineerd met een autoregressief model (Jöreskog, 1971, 1979).

Deel 1: Inter-Individuele Verschillen Modeleren met Intensief Gesampelde Data

In Hoofdstuk 2 werd een inhoudelijke interpretaties gegeven voor (inter-individuele verschillen in) de parameters van een twee level AR(1) model. Specifiek werd beargumenteerd dat de gemiddelde parameter μ_i, een persoon zijn/haar karakteristieke score voorstelt (i.c., zijn/haar lange-termijn tendens, evenwicht, of lange-termijn staat), ter-

wijl de AR-parameter ϕ_i gezien kan worden als een inertia parameter die weergeeft in welke mate eerdere scores of staten doorspelen in de huidige scores of staten.

Daarnaast, stelt de random meetfout ϵ_{it} dat gedeelte van het proces voor dat niet kan worden voorspelt op basis van voorgaande scores. Daardoor kan het worden opgevat als de collectie van alle niet-geobserveerde (of weggelaten) factoren die het bestudeerde proces beïnvloeden.

Aangezien deze effecten onderdeel worden van de innovatie term, beïnvloeden zij de variantie hiervan σ^2.

Met betrekking tot individuele verschillen in de model parameters, is het nut van individuele verschillen in het gemiddelde toelaten duidelijk: Verschillende personen hebben verschillende karakteristieke scores of lange termijn tendensen, en dit kan worden weergegeven door middel van individuele verschillen in μ_i. Daarnaast, staat het belang van het toestaan van individuele verschillen in de AR-parameter centraal in een klein aantal studies (Rovine & Walls, 2006; Suls et al., 1998; Wang et al., 2012; Kuppens et al., 2010; Koval et al., 2012), die hebben aangetoond dat deze maat voor inertie gerelateerd kan worden aan andere persoons kenmerken. Bovendien is aangetoond dat inertie toekomstige depressiviteit in adolescenten (Kuppens et al., 2012) en gezondheidsmaten (Wang et al., 2012) kan voorspellen.

De mogelijkheid van individuele verschillen in de innovation variantie is tot nu toe echter grotendeels genegeerd in de literatuur. Hoofdstuk 2 stelde daarentegen dat, vanuit inhoudelijk perspectief, individuele verschillen in de innovatie variantie zouden moeten worden verwacht vanwege twee redenen. Ten eerste zijn er waarschijnlijk individuele verschillen in de grote van de fluctuaties in niet-geobserveerde of weggelaten factoren die het bestudeerde proces beïnvloeden, en dit kan tot uiting komen in individuele verschillen in de innovatie variantie. Ten tweede verschillen personen waarschijnlijk met betrekking tot hun gevoeligheid voor deze factoren. Met andere woorden, in Hoofdstuk 2 wordt beargumenteerd dat individuele verschillen in de innovatie variantie indicatief kunnen zijn van (zeer waarschijnlijke) verschillen in gevoeligheid voor- en/of reactiviteit op niet-gemodelleerde factoren, en het relateren van deze individuele verschillen in innovatie variantie aan andere individuele verschillen kan onderzoekers helpen om meer inzicht te krijgen in de bestudeerde processen. Daarnaast is er ook nog een statistische motivatie voor het opnemen van een random innovatie variantie. Terwijl het weglaten van random effecten normaal gesproken niet tot bias in de schattingen van fixed effecten leidt in multilevel modellen (cf. Hox, 2010), werd in Hoofdstuk 2 een uitgebreide simulatie studie beschreven, die liet zien dat wanneer de innovatie variantie onterecht als fixed wordt gemodelleerd in een twee level AR(1) model, dit bias in de schatting van de AR-parameter tot gevolg heeft (waarbij de richting van de bias afhing van de correlatie tussen de innovatie variantie en de AR-parameter). Dit kan worden verklaard door het feit dat de variantie van een AR(1) proces een functie is van zowel de innovatie variantie als de AR-parameter, zodat als één van beide wordt gemodelleerd al fixed over personen, de ander de enige bron van random variantie is die verschillen tussen personen kan verklaren. De ernst van deze bias in de AR-parameter hing af van de hoeveelheid autocorrelatie. De simulatie studie liet zien dat onder realistische omstandigheden, de maximale bias waarschijnlijk

rond de -.12 ligt. Dus als de ware waarde van de AR-parameter ver van 0 af ligt, zullen de consequenties van de bias waarschijnlijk meevallen. Als de ware waarde daarentegen dicht bij 0 ligt, kan de bias de schatting omzetten van positief naar negatief. Dit laatste is een groter probleem, aangezien een negatieve AR-parameter een wezenlijk ander proces beschrijft dan een positieve.

Tot slot werd in Hoofdstuk 2 ook aangetoond dat het Bayesiaans schatten van het twee level AR(1) model de voorkeur heeft over maximum likelihood (ML) schatting, omdat ML schatting van het model ofwel leidde tot aanzienlijke bias in de schatting van de AR-parameter, ofwel het modelleren van individuele gemiddelden onmogelijk maakte. Het enige criterium waarop ML schatting wel beter presteerde dan Bayesiaanse schatting, was op het reproduceren van de rangorde in de individuele AR-parameters ϕ_i. Bayesiaanse schatting daarentegen kon individuele gemiddelden moduleren, met minimale bias in de model parameters. Bias in de fixed effecten nam wel toe naarmate N of T afnam, terwijl bias in de random effecten, die altijd positief was, sterker van N afhing dan van T, maar resultaten lieten zien dan Bayesiaanse schatting van het twee level AR(1) model zinvol kan worden toegepast voor parameter bepaling met steekproeven zo klein als $N = 20$ en $T = 10$. Als de rangorde van de individuele innovatie varianties en individuele AR-parameters ook van belang is, wordt $T \geq 50$ geadviseerd.

In Hoofdstuk 3, werd de variantie structuur van het twee level AR(1) model verder onder de loep genomen, en werd bepaald hoe (inter-individuele verschillen in) de model parameters de totale variantie beïnvloeden. Specifiek, werd de volgende expressie afgeleidt voor de totale variantie van een twee level AR(1) model,

$$\sigma_y^2 \approx \frac{\mu_{\sigma^2}}{1-(\mu_\phi^2+\tau_\phi^2)} + \frac{2\mu_\phi\tau_{\phi\sigma^2}}{(1-(\mu_\phi^2+\tau_\phi^2))^2} + \frac{\mu_{\sigma^2}[(4(\frac{\mu_\phi}{\tau_\phi})^2+2)\tau_\phi^4]}{(1-(\mu_\phi^2+\tau_\phi^2))^3} + \tau_\mu^2,$$

waar σ_y^2 staat voor de totale variantie van de tijdreeks; μ_{σ^2} en μ_ϕ respectievelijk de gemiddelde innovatie (of error) variantie en AR-parameter zijn; τ_ϕ^2 en τ_μ^2 de inter-individuele varianties zijn in respectievelijk de AR-parameter en het gemiddelde; en $\tau_{\phi\sigma^2}$ de covariantie is tussen de AR-parameter en de innovatie variantie. Vervolgens werd deze expressie gebruikt om vergelijkingen op te stellen voor de proportie verklaarde variantie op level 1, de proportie verklaarde variantie op level 2, de totale prortie verklaarde variantie, de proportie van de variantie die wordt verklaard door autocorrelatie, en de Intra-Klasse Correlatie (ICC) van een twee level AR(1) model. Aangezien de hierboven gegeven expressie al de parameters van een twee level AR(1) model bevat, kunnen toegepaste onderzoekers deze verschillende proporties en de ICC uitreken met slechts de parameter schattingen. Dit werd gedemonstreerd aan de hand van een voorbeeld waarin data van een gestructureerde dagboek-studie naar het positieve affect van 96 getrouwde vrouwen werd gebruikt (Laurenceau, Feldman Barrett, & Rovine, 2005).

Verrassend genoeg liet de expressie voor de totale variantie van een twee level AR(1) model tevens zien dat random variantie in de innovatie variantie $\tau_{\sigma^2}^2$ geen rechtstreekse invloed heeft op de totale variantie van een twee level AR(1) model. In

plaats daarvan beïnvloeden inter-individuele verschillen in de innovatie variantie de totale variantie alleen maar via hun correlatie met de AR-parameter $\tau_{\phi\sigma^2}$. Dit resultaat heeft belangrijke implicaties. In Hoofdstuk 2 werd beargumenteerd dat individuele verschillen in τ_σ^2 indicatief zijn voor differentiële sensitiviteit en/of blootstelling aan niet-gemodelleerde factoren, en dat het negeren van deze individuele verschillen kan leiden tot bias in de parameter schattingen. Samengenomen met de resultaten die in dit hoofdstuk werden gevonden, kan nu worden geconcludeerd dat deze differentiële sensitiviteit en/of blootstelling niet per sé tot uiting komt in de totale variantie van een AR(1) model. Specifiek is het zo dat inter-individuele verschillen op deze gebieden die onafhankelijk zijn van de AR-parameter, niet zullen worden opgemerkt als alleen naar de totale variantie van een tijdreeks wordt gekeken. Dit benadrukt nog maar eens dat het apart detecteren en modelleren van individuele verschillen in de innovatie variantie van groot belang is.

Deel 2: Inter-Individuele Verschillen Modelleren met Niet-Intensief Gesampelde Data

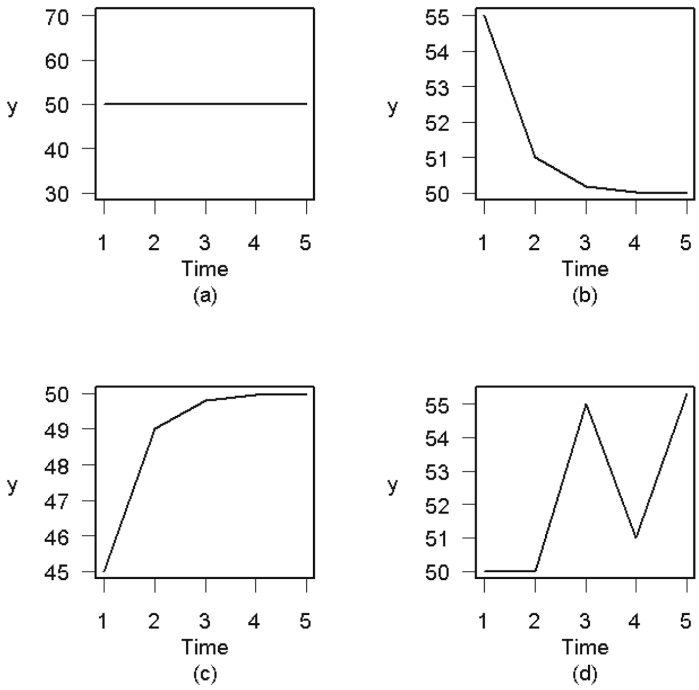
Hoofdstuk 4 van de thesis onderzocht twee "opstart" methoden die waren voorgesteld door Curran and Bollen (2001) om om te gaan met de recursie in het ALT model, en keek met name naar het effect van deze opstart methoden op de model parameters.

Het bleek dat met beide opstart methoden de inhoudelijke interpretatie van de parameters van het ALT model niet eenvoudig is. Dit geldt met name voor de opstart methode die resulteert in het predetermined ALT model, aangezien voor deze vorm van het ALT model geen vergelijkingen voor de intercept en de helling kunnen worden afgeleidt. Het predetermined ALT model bleek erg flexibel te zijn, en Hoofdtuk 3 liet zien dat het kon leiden tot onverwachte trajecten. Een predetermined ALT model met slechts een constante (i.c., $\beta_i = 0$, $\forall i$) en geen veranderings-parameter, kan bijvoorbeeld alle groeicurves in Figuur 2 genereren, terwijl een predetermined ALT model waarin μ_β en β_i negatief zijn nog steeds stijgende groeicurves kan hebben.

Hoofdstuk 4 gaf ook twee voorbeelden die aantoonde dat het predetermined ALT model goed kan passen op data die werd gegenereerd door modellen met een heel ander modelstructuur dan het ALT model zelf. Dit werd aangetoond door een populatie covariantie matrix en populatie gemiddelden te genereren voor 5 herhaalde datametingen met behulp van een kwadratisch groeicurve (QGC) model, en vervolgens het predetermined ALT model op deze data te fitten.

Tot slot werd in Hoofdstuk 4 ook een empirische illustratie gegeven die aantoonde dat deze problemen met de interpretatie van de model parameters en model fit ook kan voorkomen met echte data.

In dit hoofdstuk adviseren we onderzoekers dan ook om de door het model voorspelde gemiddelde scores te bekijken (door de juiste parameter waarden in te vullen in de vergelijkingen van het ALT model, of door een plot te maken van door het model voorspelde waarden) om de vorm te bepalen van het traject dat wordt gemodelleerd,



Figuur 2: Daadwerkelijke Groeicurves in een ALT model met slechts een latent intercept. Panel (a) bevat de daadwerkelijke (door het model voorspelde) groeicurve voor $\mu = 50$, $\mu_\alpha = 40$, en $\phi_i = \phi = 0.2$. Panel (b) bevat de groeicurve voor $\mu = 55$, $\mu_\alpha = 40$, en $\phi_i = \phi = 0.2$. Panel (c) bevat de groeicurve voor $\mu = 45$, $\mu_\alpha = 40$, en $\phi_i = \phi =$. Panel (d) bevat de groeicurve voor $\mu = 50$, $\mu_\alpha = 40$, en $\phi_i = \phi = 0.2$ voor $t = 2, 4$ en $\phi_i = \phi = 0.3$ en $t = 3, 5$.

of om de alternatieve opstart methode toe te passen die resulteert in het constrained ALT model. Met deze tweede methode is de interpretatie van de modelparameters nog steeds niet eenvoudig, maar er kunnen op zijn minst vergelijkingen worden afgeleidt voor de intercept en helling van de groeicurves.

Hoofdstuk 5 en 6 van de thesis hadden als doel om toegepaste onderzoekers instaat te stellen om hun eigen data te analyseren met dan wel het ALT model, dan wel een ander model uit een set van andere modellen die geschikt zijn voor het bestuderen van zowel stabiele als ontwikkelings-processen.

In deze context, beschreef Hoofdstuk 5 een multi-centrum gerandomiseerde trial kosteneffectiviteits-studie voor de behandeling van persoonlijkheidsstoornissen, die diende als toegepast voorbeeld voor onderzoekers. Dit hoofdstuk beschreef hoe het best passende model kon worden geselecteerd vanuit een set van modellen op basis

van het Deviance Information Criterium (DIC) (Spiegelhalter et al., 2002), en hoe dit best passende model vervolgens moest worden geïnterpreteerd.

Daarbij liet het hoofdstuk zien dat de in dit hoofdstuk besproken longitudinale random effect modellen een nuttige toevoeging kunnen zijn aan het Net-Benefit Regression (NBR) framework (Hoch et al., 2002). Nadat het best passende model is bepaald aan de hand van een systematische strategie, kon dit model vervolgens direct worden gebruikt om betekenisvolle Kosten-Effectiviteits Acceptabiliteits-curves (CEAC's), voor een voorbeeld zie (Bamelis et al., 2013)) te bepalen, zonder dat er eerst nog extra handelingen nodig waren om met missende waarden om te gaan.

Tot slot gaf Hoofdstuk 6 toegepaste onderzoekers alle informatie die nodig was om zelf hun eigen data te analyseren met de in Hoofdstuk 5 besproken random effect modellen. Specifiek werd code gegeven voor de Bayesiaanse schatting van een multilevel AR(1), LGC, QGC, en (constrained) ALT model. Op basis van de resultaten van Hoofdstuk 4, werd er geen code gegeven voor het predetermined ALT model, aangezien de parameters van dit model moeilijk zijn te interpreteren. Daarnaast bevatte Hoofdstuk 6 een gedetailleerde handleiding die stap-voor-stap instructies gaf over hoe de data moest worden voorbereidt voor analyse met één van de beschreven modellen, hoe de code voor de daadwerkelijke analyses moest worden gedraaid, en hoe de output moest worden geïnterpreteerd.

Curriculum Vitae

Personal Information

Name: Joran Jongerling
E-mail address: jongerling@fsw.eur.nl
Date of birth: 25-07-1982

Job Experience

- Erasmus University Rotterdam, Assistant Professor (.8 fte: 01/04/2015 - present).
  Assistant Professor in Methods & Statistics performing research in the field of dynamic models, with a specific focus on Bayesian multilevel time series modeling; programming analysis methods for (new elaborations of) statistical models, study and evaluation of analysis methods, reporting in scientific journals, providing statistical consultation to students and fellow researchers.
  Developing and teaching of Methods & Statistics courses in both Dutch and English to students of the faculty of Social Sciences, University College students, and PhD-students. Co-Supervisor for Master theses.

- Utrecht University, PhD (1.0 fte: 01/09/2009 - 01/01/2015).
  Performing research in the field of dynamic models, with a specific focus on Bayesian multilevel time series modeling; programming analysis methods for (new elaborations of) statistical models, study and evaluation of analysis methods, reporting in scientific journals, providing statistical consultation to students and fellow researchers.
  Teaching of Methods & Statistics courses in both Dutch and English to students of the faculty of Social Sciences, University College students, Research Master students, and PhD-students; development and teaching of courses taught as part of the matching activities of Psychology, Social Sciences and Sociology, coordination of courses, development and teaching of (instructional) lectures, development and coordination of examination procedures.

Publications

- Jongerling, J., Laurenceau, J.P. & Hamaker, E.L. (2015). *A Multilevel AR(1) Model: Allowing for Inter-Individual Differences in Trait-Scores, Inertia, and Innovation Variance.* Multivariate Behavioral Research, 50, 334-349.

- Kromhout, M. A., Jongerling, J., Achterberg W.P. (2014). *Relation between caffeine and behavioral symptoms in elderly patients with dementia: An observational study.* The Journal of Nutrition, Health & Aging, 18(4), 407-410.

- Jongerling, J. & Hamaker, E.L. (2011). *On the Trajectories of the Predetermined ALT Model: What Are We Really Modeling?* Structural Equation Modeling: A Multidisciplinary Journal, 18, 370-382.

- Brussé, I., Duvekot, J., Jongerling, J., Steegers, E., De Koning, I. (2008). *Impaired maternal cognitive functioning after pregnancies complicated by severe preeclampsia: a pilot case-control study.* Acta Obstetricia et Gynecologica Scandinavica, 87, 1-5.