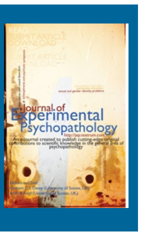
Check for updates

TEXTRUM

Journal of
Experimental Psychopathology

JEP Volume 2 (2011), Issue 2, 271–293
ISSN 2043-8087 / DOI: 10.5127/jep.013711



A Primer on Using Multilevel Models in Clinical and Experimental Psychopathology Research

Andy P. Field[a], **Daniel B. Wright**[b]

[a] *School of Psychology, University of Sussex*

[b] *Psychology Department, Florida International University*

Abstract

A Multilevel model is a statistical tool for analysing data that has a hierarchical data structure (in other words, data are nested within contexts). This paper describes what a multilevel model is, how it is described mathematically, the advantages of using this data analysis technique, and some practical issues to consider. We then move on to describe the application of multilevel models using two scenarios pertinent to researchers interested in clinical trial analysis and experimental psychopathology research. We describe how to use the software R to run these analyses.

© Copyright 2011 Textrum Ltd. All rights reserved.

Keywords: Multilevel models, clinical data, psychopathology

Correspondence to either:

Prof. Andy P. Field, Child Anxiety Theory and Treatment Laboratory (CATTLab), School of Psychology, University of Sussex, Falmer, Brighton, East Sussex, BN1 9QH. Email: andy@statisticshell.com;

Prof. Daniel B. Wright, Department of Psychology, Florida International University, 11200 S.W. 8th Street, Miami, FL 33199, USA. Email: dwright@fiu.edu

Received 17-Dec-2010; received in revised form 21-Jan-2011; accepted 21-Jan-2011

Table of Contents

A Primer on Using Multilevel Models in Clinical and Experimental Psychopathology Research

Clinical and experimental psychopathology data are often hierarchical: variables are clustered or nested within other variables. In clinical practice, one obvious example of a hierarchical data structure is a multicentre randomized control trial (e.g., Kendall, et al., 2009; Riddle, et al., 2001; Walkup, Compton, & Kendall, 2009) in which patients will be treated not only by different therapists but within different clinics (perhaps even in different geographic locations). Figure 1 illustrates this scenario: patients represent level 1 of the hierarchy, but are treated by different therapists (level 2 of the hierarchy), some of who work at different clinics (level 3 of the hierarchy). The levels of the hierarchy (i.e. therapist and clinic) are known as *contextual variables*, because they provide a context within which the effects in which we are interested happen. This hierarchy is important because both the therapist by whom a client is treated and the clinic at which they are treated might reasonably affect the success of the intervention. Even if all therapists use standardized protocols, there will be individual differences in a given therapist's skills, training, selection process and success in apply these protocols (Beidas & Kendall, 2010). In addition, the therapist-client relationship is a well-established predictor of the success of therapy that explains variance in outcomes above and beyond specific aspects of the protocol used (e.g., Hughes & Kendall, 2007). Similarly there can be subtle differences in the demographic, family and diagnosis variables of clients reporting to different clinics, especially when those clinics are geographically distant (Southam-Gerow, Silverman, & Kendall, 2006).

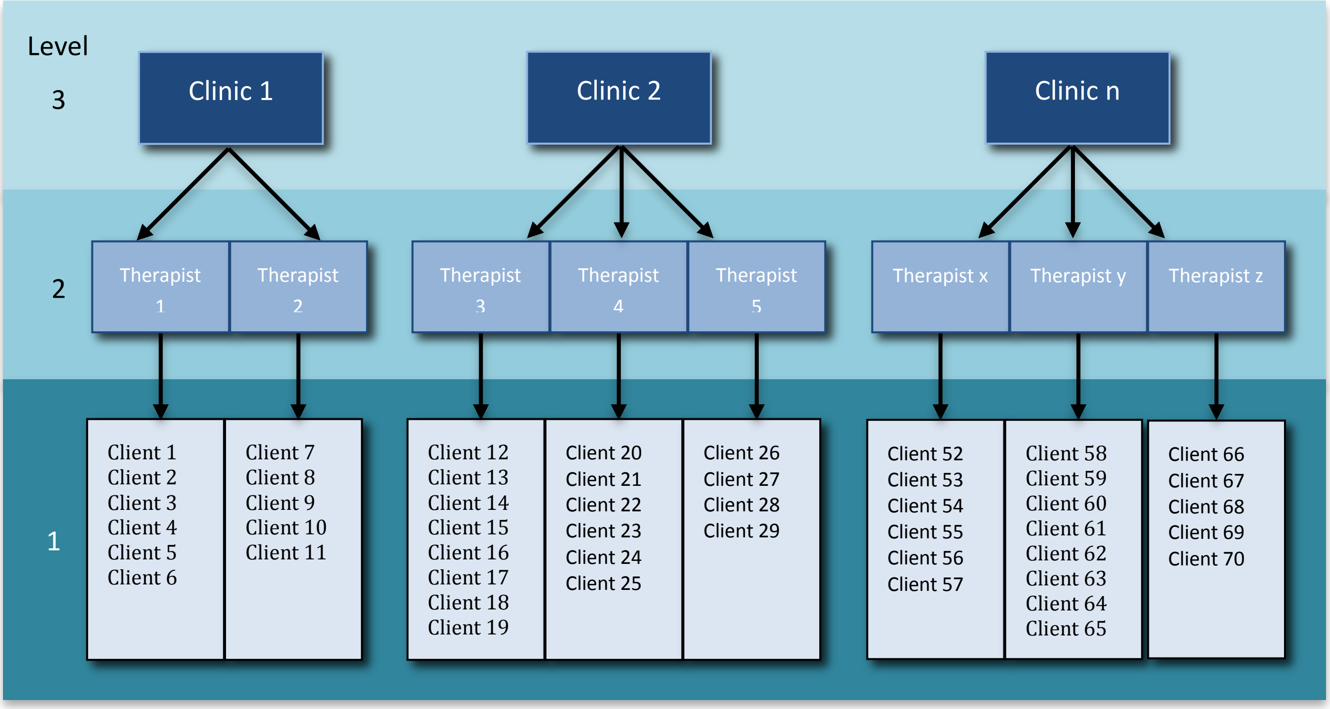

Level
3
2
1
Clinic 1
Clinic 2
Clinic n
Therapist 1
Therapist 2
Therapist 3
Therapist 4
Therapist 5
Therapist x
Therapist y
Therapist z
Client 1 Client 2 Client 3 Client 4 Client 5 Client 6
Client 7 Client 8 Client 9 Client 10 Client 11
Client 12 Client 13 Client 14 Client 15 Client 16 Client 17 Client 18 Client 19
Client 20 Client 21 Client 22 Client 23 Client 24 Client 25
Client 26 Client 27 Client 28 Client 29
Client 52 Client 53 Client 54 Client 55 Client 56 Client 57
Client 58 Client 59 Client 60 Client 61 Client 62 Client 63 Client 64 Client 65
Client 66 Client 67 Client 68 Client 69 Client 70

Figure 1: An example of a three-level hierarchical data structure. Patients (level 1) are treated by different therapists (level 2) at different clinics (level 3).

Experimental psychopathology research also often uses hierarchical data structures. Many experimental paradigms have been devised to measure the cognitive and physiological processes that cause and maintain mental health problems. In anxiety, for example, tasks such as the dot-probe task, implicit association task, affective priming, and visual search have been used to quantify attentional biases to threat and to measure indirectly fear-related beliefs (see Hadwin & Field, 2010, for a review); similarly, event-related potentials (ERP), heart rate and skin conductance responses to stimuli have been used to quantify physiological reactions to threat stimuli and situations (Dennis, Malone, & Chen, 2009; Field & Schorah, 2007; Öhman, Esteves, & Soares, 1995) . In all of these procedures participants are faced with multiple trials in which some kind of stimulus is presented and a measure is taken (a physiological reaction or a reaction time to make a decision). This scenario is similar to the one above in that just as patients treated at the same clinic should be more similar than patients treated at different clinics, responses from a particular person should be more related to each other than responses from different people. Put another way, responses within a person are not independent of each other (for example, if a person is really scared of dogs then their fear responses to different pictures of dogs should all be quite high but other people's responses might be quite low because they are not afraid of dogs). As with the clinical trial in Figure 1 it is important to take account of this lack of independence in responses.

In the clinical trial example, participants were at the bottom of the hierarchy (Figure 1); however, if we consider responses as being nested within a person, the individual is now at the top of the hierarchy (level 2) with responses below them (level 1). Figure 2 shows this situation for a hypothetical experiment in which reaction times (RT) on 8 trials are measured for each individual. These reaction times on different trials are nested within participants in a two-level hierarchy. As can be seen from Figure 2 it is possible that a given reaction time might be missing and this is common in psychopathology research. For example, in reaction time tasks, trials are often excluded if the response given was an error, there are often technical failures that result in some physiological responses not being recorded, and with ERP and skin conductance measures trials are often excluded if the noise to signal ratio is too high. As we shall see, these missing data points pose fewer problems when multilevel models are used than with traditional procedures for repeated measures data.

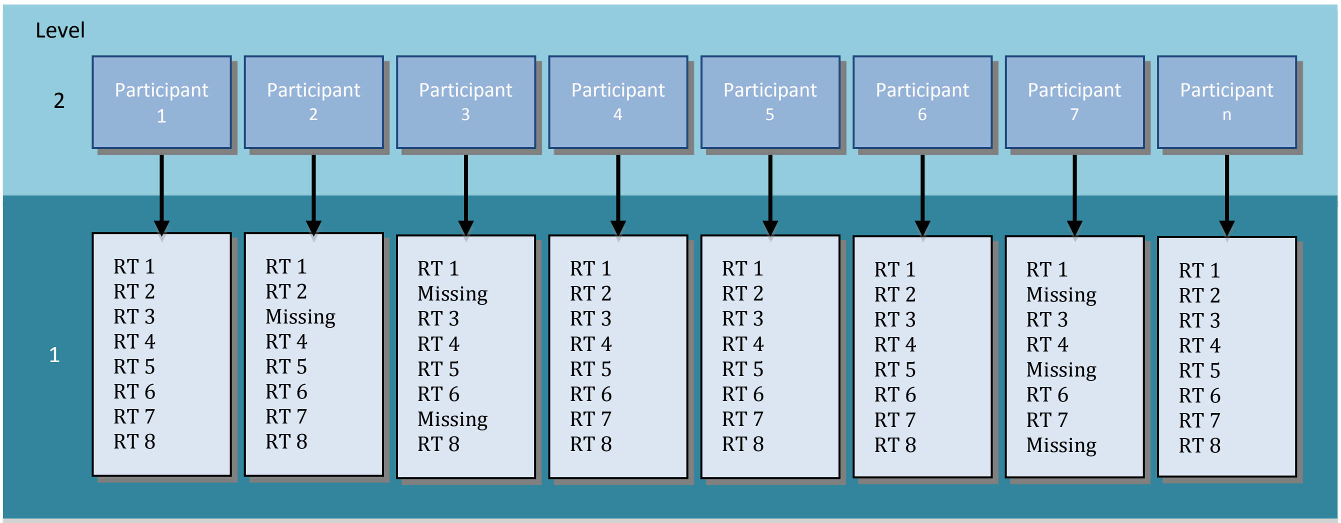

Level
2
Participant 1
Participant 2
Participant 3
Participant 4
Participant 5
Participant 6
Participant 7
Participant n
1
RT 1 RT 2 RT 3 RT 4 RT 5 RT 6 RT 7 RT 8
RT 1 RT 2 Missing RT 4 RT 5 RT 6 RT 7 RT 8
RT 1 Missing RT 3 RT 4 RT 5 RT 6 Missing RT 8
RT 1 RT 2 RT 3 RT 4 RT 5 RT 6 RT 7 RT 8
RT 1 RT 2 RT 3 RT 4 RT 5 RT 6 RT 7 RT 8
RT 1 RT 2 RT 3 RT 4 RT 5 RT 6 RT 7 RT 8
RT 1 Missing RT 3 RT 4 Missing RT 6 RT 7 Missing
RT 1 RT 2 RT 3 RT 4 RT 5 RT 6 RT 7 RT 8

Figure 2: An example of a two-level hierarchical data structure in which reaction time responses (level 1) are nested within participants (level 2).

Of course it is possible to have more complex hierarchies. For example it is not difficult to imagine a situation that combines Figure 1 and 2 and, therefore, reaction time measures are used in patients who are treated by different therapists at different clinics. This would give rise to a four level hierarchy: Clinic (level 4), therapist (level 3), participant (level 2), and reaction time (level 1). Multilevel models can also handle situations in which, for example, a therapist works at two different clinics and uses similar intervention techniques at both clinics. It should also be apparent that Figure 2 could equally represent time series data in which responses at different time points are nested within individuals. This is known as a growth model (Field, 2009). (In fact, the example of reaction times measured for different trials in an experiment is effectively a time series design in which the time intervals between measures are seconds rather than days, weeks or months.)

A multilevel linear model (also known as hierarchical linear models, hierarchical regression and mixed model[1]) is a versatile tool for analysing hierarchical data structures. Most important, it offers a way of explicitly modelling the relatedness of observations. To do this it uses the Intra Class Correlation (ICC). Imagine a two-level example of patients (level 1) being treated by different therapists (level 2). The ICC represents the proportion of the total variability in symptom severity that is attributable to the therapist. If the variability within a therapist's patients is small (they all have high symptom severity), but the variability between her patients and those of a different therapist is maximised (i.e. some therapists see clients who have relatively mild symptoms), then the ICC is relatively large. Conversely, if symptom severity is relatively equal across all therapists then symptom severity will vary a lot within patients

[1] The term 'Mixed model' is often erroneously used to mean 'mixed design', but these are not the same thing. A mixed model is one that contains both fixed and random effects (as we will explain); in contrast, a mixed design contains only fixed factors one or more of which have been measured between subjects whereas others have been measured within subjects.

assigned to the same therapist, and the differences in symptom severity between patients assigned to different therapists will be relatively small. This would give rise to a relatively small ICC. Therefore, the ICC is a gauge of whether a contextual variable has an effect on the outcome; it reflects the variability between levels of a contextual variable (comparing therapists) relative to the variability within levels of a contextual variable (in this case the therapist to which a patient is assigned).

This paper offers a brief tutorial, based heavily on Field (2009), on using multilevel models but with examples pertinent to researchers engaging in clinical psychology or experimental psychopathology research. We begin by describing the benefits of using multilevel models, and then provide a brief overview of what a multilevel model is. We progress to look at some practical issues before describing two examples using the R package *nlme* (Pinheiro, Bates, DebRoy, Sarkar, & R Development Core Team, 2010).

I Already Know How to Analyse My Data, Why Should I Learn Something New?

There are many benefits to using multilevel models that overcome problems that are common in clinical and experimental data. In this section we briefly outline some of these problems.

Overcoming the Independence of Errors Assumption

Many of the statistical tests commonly used by clinical psychology and experimental psychopathology researchers (e.g. general linear models such as regression, ANOVA, *t*-tests) assume that the errors in the models (known as *residuals*) are independent. That is, there is no systematic relationship between any of the residuals. Violation of this assumption, at best, limits the generalizability of the model, and at worst can make test statistics, notably the *F*-ratio, behave very erratically (see Field, 2009, for a non-technical review of these issues). When people are sampled from similar contexts, the independence assumed by the tests with which we typically analyze data is unlikely to be true. By modelling the variation due to contextual variables we overcome this problem of non-independent observations. This also opens up the possibility of conducting regression analyses on data for which individual participants contribute multiple data points (i.e. time series designs). Imagine a clinical trial in which depression scores are taken at each of 10 sessions. If we wanted to look at variables that predict depression (as we would in an OLS regression analysis) a typical solution might be to compute the change in depression pre- to post-treatment for each client so that they contribute only a single depression score to the analysis (and, therefore, residuals will most likely be independent). In computing this change score we have lost a lot of data (80% of the data will not be used). Using a multilevel model enables us to look at predictors of depression but retain the data from all time points.

Homogeneity of Regression Slopes

Analysis of covariance assumes that the relationship between the covariate and the outcome is the same across the different groups that make up the predictor variable. When this assumption is met the resulting *F* statistic can be assumed to have the corresponding *F*-distribution; however, when the assumption is not met the resulting *F* statistic can not be assumed to have the corresponding *F*-distribution meaning that the resulting test statistic is being evaluated against a distribution different to the one that it actually has. Consequently, the Type I error rate of the test is inflated and the power to detect effects is not maximised (Hollingsworth, 1980). This is especially true when group sizes are unequal (Hamilton, 1977) and when the standardized regressions slopes differ by more than .4 (Wu, 1984). Using a multilevel model we can model the variability in regression slopes.

Missing Data

Many statistical procedures (e.g., Regression, ANOVA, ANCOVA) cannot cope well with missing data, or test statistics (such as *F*) behave oddly when designs are unbalanced. At best, unequal numbers of observations compromise the robustness of the test statistic when other assumptions are violated: for example, the *F* ratio in ANOVA and ANCOVA is considerably less robust in the context of unequal group sizes (see Field, 2009, for a non-technical review).

Missing data are a particular problem within clinical trials because it is common to attempt to collect follow-up data, often many months after treatment has ended when patients might be difficult to track down. Of course, there are ways to correct for and impute missing data, but these techniques are often quite complicated (Yang, Li, & Shoptaw, 2008), therefore, often when using repeated measures designs if a single time point is missing the whole case usually needs to be deleted; missing data leads to more data being deleted. Multilevel models do not require complete data sets and so when data are missing for one time point they do not need to be imputed, nor does the whole case need to be deleted. Instead parameters can be estimated successfully with the available data, which offers a relatively easy solution to dealing with missing data.

It is important to stress that no statistical procedure can overcome data that are missing; Good methods, designs and research execution should be used to minimize missing values and reasons for missing values should always be explored. It is just that when using traditional statistical procedures for repeated measures data additional procedures to account for missing data are usually necessary and can be problematic.

What is a Multilevel Model?

We will explain the basics of multilevel models using an example. The data file is the same as the cosmetic surgery example used in Chapter 19 of Field (2009), but with the variables changed to represent a clinical trial. Using this file has the advantage that, although we will explain how to run the analysis using R, readers can consult this text to see how the analysis is run in SPSS or SAS (Field & Miles, 2010), or indeed find out more about running the analysis in R (Field & Miles, In Press). In the current context, the example looks at the effects of an intervention (Cognitive Behaviour Therapy, CBT, vs. a waiting list control) on anxiety scores. The data set contains the following variables: (1) our outcome variable, *Post_Anxiety*, a measure of anxiety after the intervention; (2) *Base_Anxiety*, the level of anxiety when entering the trial (i.e. at the start of treatment); (3) Intervention, a dummy variable that specifies whether the person was in the CBT group, 1, or on the waiting list, 0, which acts as our control group; and (4) *Clinic*, which specifies which of 10 clinics the person attended to have treatment.

Fixed and Random Coefficients

General linear models are characterized by two things: the intercept of the model, b_0, and the regression coefficients for each predictor, b_1, b_2, b_3 etc. In a general form, then, the general linear model is:

$$\begin{aligned}\text{Outcome}_i = b_0 + b_1\text{Predictor 1}_i + b_2\text{Predictor 2}_i + \cdots \\ b_n\text{Predictor }n_i + \varepsilon_i\end{aligned} \tag{1}$$

Normally when you conduct a general linear model (e.g., OLS regression, ANOVA, ANCOVA etc.) you, whether you know it or not, treat these parameters as fixed. However, in multilevel models there is a distinction between fixed parameters and random parameters.

By assuming that regression parameters are fixed, we assume that the resulting model holds true across the entire sample and that for every case of data in the sample we can predict a score using the same values of the *b*s. However, we can also conceptualize these parameters as being random. In other words, we can assume that they are different in different people (so, for one person we assume certain values for the *b*s but for a different person we assume a different set of *b*s). Usually, we are assuming that these *b*s vary across contexts (for example, clinics). Given that the *b*s quantify the size of the effect that we are studying, this assumption of random parameters implies that effects vary across contexts. For our clinical trial example, this means that rather than assume that the effect of CBT (compared to a waiting list) is equal in different clinics (regression parameters are fixed) we instead assume that treatment effects might vary across different clinics (regression parameters are random). There are two ways in which we can introduce random effects: the intercept, and the slope.

The random intercept model

The first way to introduce random parameters into the model is to assume that the intercepts vary across contexts (or groups). For our clinical trial, if we assume that the slopes are fixed and the intercepts are random then we assume that the difference between the waiting list and CBT group is the same in each clinic (i.e., the slopes are the same), but that the overall levels of anxiety differ in each clinic (i.e., the intercepts are different)[2].

The random slope model

We can also assume that the slopes vary across contexts. For our clinical trial, if we assume that the intercepts are fixed but the slopes are random then this is like assuming that the difference between the waiting list and CBT group is different in each clinic (i.e., the slopes are different), but that the overall levels of anxiety are the same in each clinic (i.e., the intercepts are the same). This is one situation when we violate the assumption of homogeneity of regression slopes in ANCOVA[3]. Therefore, we can use a multilevel model to explicitly estimate this variability in slopes across contexts/groups. The most realistic situation is to assume that both intercepts and slopes vary around the estimates of the centre of the distributions for the intercepts and slope (in other words both the difference in anxiety between the treatment and control group, and the overall levels of anxiety vary across clinics).

Building a multilevel model

We will now look at how to represent these models. Imagine that we wanted to predict someone's Anxiety after treatment (Post_Anx) from the intervention group to which they belonged (CBT or waiting list). We can represent this as a linear model as follows:

$$\text{Post\_Anx}_i = b_0 + b_1\text{Intervention}_i + \varepsilon_i \qquad (2)$$

The *i* represents the level 1 variable, in this case the people we tested. In this example, we had a contextual variable, which was the clinic in which the anxiety treatment was conducted. As we have seen, we might expect the effect of treatment on anxiety to vary as a function of which clinic the treatment was conducted at because clinicians will differ in their skill and technique (random slopes), but also overall levels of anxiety to vary across clinics (random intercepts).

[2] The situation of having fixed slopes but random intercepts (and vice versa) is just one situation that we have picked because the interpretation is fairly straightforward.

[3] Heterogeneity of regression slopes can arise when intercepts are random, but we describe a situation in which intercepts are fixed to keep the discussion as simple as possible. This situation of random slopes and fixed intercepts is not particularly realistic because differences in slope will tend to create differences in the overall level.

To include a random intercept we add a component to the intercept that measures the variability in intercepts, u_{0j}. Therefore, the intercept changes from b_0 to become $(b_0 + u_{0j})$. This term estimates the intercept of the overall model fitted to the data, b_0, and the variability of intercepts around that overall model, u_{0j}. The overall model becomes:

$$\text{Post\_Anx}_{ij} = (b_0 + u_{0j}) + b_1 X_{ij} + \varepsilon_{ij} \tag{3}$$

The *j*s in the equation reflect levels of the variable over which the intercept varies (in this case the clinic) — the level 2 variable. This equation is usually written to look like an ordinary regression equation except that the intercept has changed from a fixed, b_0, to a random one, b_{0j}, which is defined in a separate equation:

$$\begin{aligned} Y_{ij} &= b_{0j} + b_1 X_{ij} + \varepsilon_{\text{ij}} \\ b_{0j} &= b_0 + u_{0j} \end{aligned} \tag{4}$$

If we want to include random slopes for the effect of the intervention on post treatment anxiety, then we add a component to the slope of the overall model that measures the variability in slopes, u_{1j}. Therefore, the slope changes from b_1 to become$(b_1 + u_{1j})$. This term estimates the slope of the overall model fitted to the data, b_1, and the variability of slopes, u_{1j}, in different contexts around that overall model. Again this is typically written taking the error terms out into a separate equation to make the link to a familiar linear model clear. It now looks like an ordinary regression equation except that the slope has changed from a fixed, b_1, to a random one, b_{1j}, which is defined in a separate equation:

$$\begin{aligned} Y_{ij} &= b_{0i} + b_{1j} X_{ij} + \varepsilon_{ij} \\ b_{1j} &= b_1 + u_{1j} \end{aligned} \tag{5}$$

If we want to model a situation with random slopes *and* intercepts, then we combine the two models above. We still estimate the intercept and slope of the overall model (b_0 and b_1) but we also include the two terms that estimate the variability (across clinics) in intercepts, u_{0j}, and slopes, u_{1j}. We write this model as a basic linear model, but replace the fixed intercept and slope (b_0 and b_1) with their random counterparts (b_{0j} and b_{1j}):

$$\begin{aligned} Y_{ij} &= b_{0j} + b_{1j} X_{ij} + \varepsilon_{ij} \\ b_{0j} &= b_0 + u_{0j} \\ b_{1j} &= b_1 + u_{1j} \end{aligned} \tag{6}$$

If we were conducting an ordinary regression and we wanted to extend the model to include another predictor, for example anxiety before surgery, you should be familiar with the idea that we simply add this variable to the model with an associated beta:

$$\text{Post\_Anx}_i = b_0 + b_1 \text{Intervention}_{\text{i}} + b_2 \text{Pre\_Anx}_i + \varepsilon_i \tag{7}$$

If we want to allow the intercept of the effect of treatment on anxiety after treatment to vary across clinics then we replace b_0 with b_{0j}. Similarly, if we want to allow the slope of the effect of treatment on anxiety after treatment to vary across clinics then we replace b_1 with b_{1j}. So, even with a random intercept and slope, our model stays much the same:

$$\begin{aligned}&\text{Post\_Anx}_{ij} = b_{0j} + b_{1j}\text{Intervention}_{ij} + b_2\text{Pre\_Anx}_{ij} + \varepsilon_{ij}\\&b_{0j} = b_0 + u_{0j}\\&b_{1j} = b_1 + u_{1j}\end{aligned} \tag{8}$$

Remember that the *j* in the equation relates to the level 2 contextual variable (clinic in this case). To sum up, when conducting a multilevel model you are simply doing a regression in which intercepts, slopes, or both, can vary across different contexts. All that really changes is that for every parameter that we allow to be random, we get an estimate of the variability of that parameter as well as the parameter itself.

Some Practical Issues

Assessing the Fit and Comparing Models

The overall fit of the model is tested using a chi-square likelihood ratio test; the smaller the value of this statistic, the better. There are different versions of this statistic, which are corrected for various things. Two of the most common ones are: (1) Akaike's information criterion (AIC), a goodness-of-fit measure that is corrected for model complexity; and (2) Schwarz's Bayesian criterion (BIC), a more conservative version of the AIC, which is appropriate when sample sizes are large and the number of parameters is small.

The AIC and BIC are the most commonly used. They are useful as a way of comparing models and are not interpretable in their own right. Typically, multilevel models are built up from a baseline model and then fit indices compared to see whether additions to the model have improved the fit. For example, you might start with a basic model, add in a random intercept and see if the fit improves, then add in random slopes and again see if this change improves the fit. Provided that full maximum likelihood (ML) estimation is used (and not restricted maximal likelihood, REML) and the new model contains all of the effects of the older model it is simple to compare models: You simply calculate the difference between the log-likelihoods of the models, which gives us a chi-square statistic with degrees of freedom equal to the difference in the number of parameters between the two models. There is nothing inherently wrong with using REML, but if you want to compare models in the way just described then you need to use full ML.

Types of covariance structures

If you include random effects or have repeated measurements (as in the reaction time example in Figure 2 or a time series design) then you should model the covariance structure of your data. If you have both random effects *and* repeated measures, it is possible to specify different covariance structures for each. The reason is that a model covariance matrix that is a good fit of the data will *usually* increase the power to detect fixed effects.

The covariance structure specifies the form of a matrix in which the diagonal elements are variances and the off-diagonal elements are covariances, known as the variance-covariance matrix. There are various forms that this matrix could take and most of the time the best you can do is take an educated guess at the form that best fits your data. It is also useful sometimes to run the model with different covariance structures and compare them use the goodness-of-fit indices mentioned above to see whether some structures fit better than others.

The covariance structure is the starting point to estimate the fixed effects model parameters. As such, you will get different results depending on which covariance structure you choose. The default covariance structure for random effects usually assumes that all random effects are independent (the covariances in the variance-covariance matrix are 0). Variances of random effects are assumed to be the same (hence they are 1 in the variance-covariance matrix). This structure is sometimes called the *independence model* or *variance components*. This extreme model is often not appropriate (Pinheiro, et al., 2010).

A similar structure is called *diagonal* and is the same as the independence model except that variances are assumed to be heterogeneous. Another very common covariance structure (especially for repeated measures and time series data) is a first-order autoregressive structure, usually referred to as AR(1). This covariance structure assumes that the covariances between repeated measurements is highest at adjacent time points. For example, the correlation between time points 1 and 2 is ρ; but the correlation between time point 1 and 3 is ρ^2. At time 4, the correlation with time 1 goes down again to ρ^3. So, the correlations between adjacent time points (t1 and t2, t2 and t3, t3 and t4) are assumed to be ρ, scores two intervals apart (t1 and t3, t2 and t4) are assumed to have correlations of ρ^2, and scores three intervals apart (t1 and t4) are assumed to have correlations of ρ^3. So the correlation between scores gets smaller over longer time intervals. In AR(1) variances are assumed to be homogeneous but there is a version of this covariance structure where variance can be heterogeneous. This structure is often used for repeated-measures data (especially when measurements are taken over time such as in growth models). Finally, you can use an *unstructured* covariance structure in which covariances are assumed not to conform to a systematic pattern.

Assumptions

Multilevel models merely extend the general linear model and so all of the assumptions for regression apply to multilevel models (Field, 2009, as well as many other textbooks, review these assumptions). In addition, random coefficients are assumed to be normally distributed around the overall model (e.g., random intercepts in the different contexts are assumed to be normally distributed around the overall model). There are also multilevel models designed for outcomes that are dichotomous, a logistic multilevel model (Hox, 2010, has a good introductory chapter on this topic), and as mentioned earlier the assumptions of independent errors can sometimes be solved by a multilevel model because certain covariance structures (e.g., AR(1)) explicitly model the co-dependence between data points.

Sample Size and Power

The situation with power and sample size is very complex indeed because potentially we need to answer several quite different questions: (1) what power is there to detect both fixed and random effects? (2) What is the optimal ‘sample’ of units for a contextual variable (to take our clinics example, how many clinics should we include in the study)? and (3) what is the optimal ‘sample’ of units *within* contexts (how many clients should we have within each clinic?)? Twisk (2006) points out that typically to determine the number of individuals needed in a multilevel study, a standard sample-size calculation is made that is then corrected for the hierarchical data structure. However, there are two corrections that are typically applied and they can yield wildly different estimates.

Kreft and de Leeuw (1998) and Twisk (2006) agree that (1) the number of contexts relative to individuals within those contexts is important; and (2) models with more levels will have more parameters to be estimated and therefore require larger the sample sizes. Kreft and de Leeuw (1998) suggest that if you are looking for cross-level interactions then you should aim to have more than 20 contexts (groups) in the higher-level variable, and that group sizes ‘should not be too small’. It is also noteworthy that they argue that it is impossible to produce any meaningful rules of thumb because there are so many factors involved in multilevel analysis. Twisk (2006) also believes that even if you can calculate the optimal combination of context levels and participants within those levels, in practice it will be hard to achieve those optimal values. He believes that sample size calculations are rather limited and recommends using them with caution.

If Twisk's words of warning do not deter you then Field (2009) notes that there are computer programs that will attempt the necessary calculations for you. HLM (http://www.ssicentral.com/hlm/index.html) will do power calculations for multilevel models, and for two-level models Tom Snijders' PinT program (http://stat.gamma.rug.nl/multilevel.htm) works well.

Centring Variables

Centring is the process of transforming a variable into deviations around a fixed point, typically the grand mean. There are two forms of centring that are typically used in multilevel modelling: grand mean centring (GMC) and centring within clusters (CWC). GMC means that for a given variable we take each score and subtract from it the mean of all scores (for that variable). CWC that for a given variable we take each score and subtract from it the mean of the scores (for that variable) within the group to which the data point belongs.

Let's look at an example. Imagine we had clients with social anxiety (level 1) treated within clinics (level 2), our outcome variable was social anxiety symptom severity and we wanted to predict this from the number of secondary diagnoses pre-treatment (*comorbidity*). We could centre comorbidity either on its mean (so express comorbidity as the deviation from the average number of secondary diagnoses of all clients), or we could centre it on the mean comorbidity within a particular clinic (so express a client's comorbidity as a deviation from the average comorbidity of clients at the same clinic).

The issues around centring in multilevel models are very complex, and we have space only to highlight some key issues, but Enders and Tofighi (2007) and Kreft and colleagues (Kreft & de Leeuw, 1998; Kreft, de Leeuw, & Aiken, 1995) have written excellent reviews, which we recommend that you read. If you fit a multilevel model using uncentred predictors and then fit the same model but with GMC predictors then the parameters will change, but the models will fit the data equally well and have the same predicted values and residuals. Therefore, GMC does not change the model, but it changes your interpretation of the parameters. When CWC is used the model is not equivalent to the uncentred model in either the fixed part or the random part (Kreft & de Leeuw, 1998).

As such, one big issue with the choice of centring is that it changes the value of parameter estimates and how they are interpreted. If we take the aforementioned example of comorbidity as a predictor of social anxiety symptoms post-treatment, these are both level 1 variables (measured within clients). If we use GMC for the comorbidity variable then the intercept for a given clinic is interpreted as the predicted symptom score for a client whose comorbidity value is the grand mean of all comorbidity scores. If we use CWC then the intercept for a given clinic is interpreted as the predicted symptom score for a client whose comorbidity value is the mean of comorbidity scores for that clinic. These are two very different interpretations.

The choice of centring should not be determined on a statistical basis because there is no statistically correct choice between not centring, CWC and GMC (Kreft, et al., 1995). Instead, Endler and Tofighi recommend making decisions based on the substantive research question. In short, they make four recommendations when analysing data with a 2-level hierarchy: (1) CWC should be used if the primary interest is in an association between variables measured at level 1 (i.e., the aforementioned relationship between comorbidity and social anxiety symptoms); (2) GMC is appropriate when the primary interest is in the level 2 variable but you want to control for the level 1 covariate (i.e., you want to look at the effect of clinic on social anxiety symptoms while controlling for comorbidity); (3) either GNC and CWC can be used to look at the differential influence of a variable at level 1 and 2 (i.e., is the effect of comorbidity on social anxiety symptoms different at the clinic level to the client level?); and (4) CWC is preferable for examining cross-level interactions (e.g., the interactive effect of clinic and comorbidity on anxiety symptoms).

Doing Multilevel Models

Most commonly used statistics packages (e.g. SPSS, SAS, R, MPLUS) have some facilities for multilevel models and there are specialist packages too (e.g., MLwiN). We are going to focus our tutorials around the package R because it is free and works on both MacOS and Windows. Tutorials for SPSS and SAS using the same data set as Example 1, and a more detailed exploration of using R can be found in textbooks by the first author (Field, 2009; Field & Miles, 2010, In Press).

A Very Brief Guide to R

R (R Development Core Team, 2010) is an environment/language for statistical analysis and is the fastest growing statistics software. It is freely available from cran.r-project.org. If you have never used R it will be valuable to read some introduction to R, for example Chapter 1 of Wright and London (2009) or Chapter 3 of Field & Miles (In Press). There are thousands of R packages that can be freely accessed from within R. The R code for all of our examples is linked from this article as an online resource. R is a command language and so we type in commands that R then executes; when referring to commands typed into R we will use blue text. R is a powerful programming language, but to the uninitiated it appears bewildering. These brief tutorials should get you started.

Example 1: Clinical Trial Data

The first tutorial uses the example, which we have used throughout this paper, of a clinical trial of anxiety. Post-treatment anxiety scores are predicted from the variables intervention (CBT or waiting list) and pre-therapy anxiety scores. The equations already used in this paper define this model in various forms. These data can be found in the file **Field & Wright (CBT).sav**, which is an SPSS data file. We show how to do this with an SPSS data file because SPSS is a popular package.

Initializing the package and importing the data

To enable R to read SPSS files we use two commands; the first is to load a package called *foreign* (R Development Core Team, Sarkar, DebRoy, Bivand, & Others, 2010) in case you do not have this installed, and the second initiates this package in the current session:

```
install.packages("foreign")
library(foreign)
```

Next we need to choose the SPSS data file and import it into R. The following two functions do this; *file.choose* opens a dialog box that enables us to select the file in the usual way in Windows or MacOS, it then sets the object, which we have called ‘filename’ to be this file. If you know the file name or are accessing data from a URL put in the location. The *read.spss* function takes the object ‘filename’ (which we defined before as the file that you have just chosen) and reads it into a data frame that we have called *cbtData*. (You can call the data frame whatever you like but it makes life easier, especially when looking back at code that you wrote months before, if you assign sensible, informative names to objects that you create in R.)

```
filename <- file.choose()
cbtData <- read.spss(filename, use.value.labels=FALSE, to.data.frame=TRUE)
```

We now have a data frame called *cbtData* that contains all of the variables that were in the SPSS files. The variables will have the same name as in the SPSS file, and we can refer to them by appending $ and the name of the variable to the name of the data frame. For example, to access the variable *Post_Anxiety* in the data frame *cbtData* we would write *cbtData$Post_Anxiety*.

Fitting a baseline model

There are several packages that can be used for multilevel models. Two of the most used are: *nlme* (Pinheiro, et al., 2010) and *lme4* (Bates & Maechler, 2010). We are going to focus on the package *nlme* (*non linear mixed effect*) because, unlike *lme4*, it enables you to model the covariance structure, which will be useful in the second example. First, we will predict post treatment anxiety scores ignoring the data hierarchy. This model is identical to performing an ANCOVA with *Intervention* as the fixed effect and *Baseline_Anxiety* as the covariate.

```
fixedOnly <-lm(Post_Anxiety~Intervention + Base_Anxiety, data = cbtData)
summary(fixedOnly)
```

We have used the command *lm* (linear model) to create an object called fixedOnly (short for fixed effect only). The commands in brackets tell lm what model we want to fit; the '~' means 'predicted from'. So, we have specified that we want *Post_Anxiety* predicted from *Intervention* + *Base_Anxiety*. In other words we have simply written out the linear model in equation 7 but without the *b*s. The rest of the commands simply tell lm to fit the model on the data frame that we just created (data = cbtData). Finally summary(fixedOnly) prints the model parameters to the R console.

Figure 3 shows the output of this model (which gives the same values as SPSS Output 19.4 in Field, 2009). We can see from this output that Intervention had a significant effect on post treatment anxiety, $b = -1.70$, $t = -2.01$, $p = .045$, as did baseline anxiety, $b = 0.67$, $t = 14.66$, $p < .001$. The model as a whole fitted the data significantly too, $F(2, 273) = 107.7$, $p < .001$. But all the p values assume the residuals are independent, which we expect they are not, thus these p values should not be trusted.

Fixed Effects Model

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.14702    2.90767   6.241 1.65e-09 ***
Intervention -1.69723    0.84404  -2.011   0.0453 *
Base_Anxiety  0.66504    0.04537  14.659  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.977 on 273 degrees of freedom
Multiple R-squared: 0.4411,   Adjusted R-squared: 0.437
F-statistic: 107.7 on 2 and 273 DF,  p-value: < 2.2e-16
```

Figure 3: Selected output from the lm function in R for a fixed effects model (see text for details).

Fitting a random intercepts model

We can now allow intercepts to vary across clinics by using the *lme* (linear mixed effect) command to create an object that we have named *randomIntercept* (to remind you of what this model represents).

```
randomIntercept <-lme(Post_Anxiety~Intervention + Base_Anxiety, data = cbtData, random =
~1|Clinic, method = "ML", na.action = na.exclude)
summary(randomIntercept)
```

The format of the *lme* command is much the same as before; the only difference is that we have added the instruction ‘random = ~1|Clinic’. In this command, ‘random = ~1’ means ‘make the intercepts random’ and ‘|Clinic’ tells R that intercepts are to vary across clinics (in other words, participants are nested within clinics). We have also added two new commands. The first instructs R to use maximum likelihood estimation (method = “ML”). This command could be changed to use Restricted Maximum Likelihood (method = “REML”), but ML should be used when comparing models. We have also added the command ‘na.action = na.exclude’, which tells R to exclude any missing data. In this example there are no missing data so this command is unnecessary and we will not include it again, but we mention it here because it needs to be specified for data sets with missing data.

Figure 4 shows the output of this model, which was obtained using ‘*summary(randomIntercept)*’, (this gives the same values as SPSS Output 19.5 in Field, 2009). We can see from this output that Intervention no longer has a significant effect on post treatment anxiety, *b* = –0.31, *t* = –0.37, *p* = .711, but baseline anxiety still does, *b* = 0.47, *t* = 9.07, *p* < .001.

Random Intercept Model

```
Linear mixed-effects model fit by maximum likelihood
 Data: cbtData
      AIC      BIC   logLik
  1847.49 1865.592 -918.745

Random effects:
 Formula: ~1 | Clinic
        (Intercept) Residual
StdDev:    3.039264 6.518986

Fixed effects: Post_Anxiety ~ Intervention + Base_Anxiety
                 Value Std.Error  DF   t-value  p-value
(Intercept)  29.563601  3.471879 264  8.515160  0.0000
Intervention -0.312999  0.843145 264 -0.371228  0.7108
Base_Anxiety  0.478630  0.052774 264  9.069465  0.0000
 Correlation:
             (Intr) Intrvn
Intervention  0.102
Base_Anxiety -0.947 -0.222
```

Figure 4: Selected output from the lme function in R for a random intercepts model (see text for details).

Fitting a model with random intercepts and slopes

Finally, we can create an object called *randomSlopesIntercepts* that allow both slopes and intercepts to vary across clinics by writing the command as:

> randomSlopesIntercepts <-lme(Post_Anxiety~Intervention + Base_Anxiety, data = cbtData, random = ~Intervention|Clinic, method = "ML")

Note that all we have changed is ‘random = ~1|Clinic’ to become ‘random = ~Intervention|Clinic’. Figure 5 shows the output of this model, which was obtained using ‘*summary(randomSlopesIntercepts)*’. We can see from this output that Intervention, *b* = –0.65, *t* = –0.31, *p* = .757, does not have a significant effect on post treatment anxiety but baseline anxiety, *b* = 0.31, *t* = 5.80, *p* < .001, does. We can see whether adding random slopes to the random intercepts model improved the model by typing:

> anova(randomIntercept, randomSlopesIntercepts)

which produces fit indices (BIC and AIC) compares the two models using a likelihood ratio test. The resulting output (also in Figure 5) shows a highly significant difference indicating that the fit of the model improves significantly by adding random slopes to a model that already contains random intercepts. Also, note that the AIC and BIC (the measures of goodness-of-fit) are smaller for *randomSlopesIntercepts* than for *randomIntercept* indicating that it is a better fit of the data (remember a smaller value of AIC and BIC indicates a better fit).

Random Slopes and Random Intercepts

```
Linear mixed-effects model fit by maximum likelihood
 Data: cbtData
       AIC      BIC    logLik
  1812.624 1837.967 -899.3119

Random effects:
 Formula: ~Intervention | Clinic
 Structure: General positive-definite, Log-Cholesky parametrization
             StdDev   Corr
(Intercept)  6.132656 (Intr)
Intervention 6.197489 -0.965
Residual     5.912335

Fixed effects: Post_Anxiety ~ Intervention + Base_Anxiety
                Value Std.Error  DF   t-value p-value
(Intercept)  40.10253  3.892945 264 10.301334  0.0000
Intervention -0.65453  2.110917 264 -0.310069  0.7568
Base_Anxiety  0.31022  0.053506 264  5.797812  0.0000
 Correlation:
             (Intr) Intrvn
Intervention -0.430
Base_Anxiety -0.855 -0.063
```

Comparing Models

```
anova(randomIntercept, randomSlopesIntercepts)
```

```
                       Model df   AIC      BIC    logLik   Test  L.Ratio   p-value
randomIntercept            1  5 1847.490 1865.592 -918.7450
randomSlopesIntercepts     2  7 1812.624 1837.966 -899.3119 1 vs 2 38.86626  <.0001
```

Figure 5: Selected output from the lme function in R for a model with random slopes and intercepts (see text for details).

Example 2: Reaction Time Data

The second tutorial uses the example of a visual search task, which is commonly used in child and adult anxiety research (see Donnelly, Hadwin, Manneer, & Richards, 2010, for a review). The data are from a study similar to Field (Field, 2006a, 2006b), in which children were given either verbal threat information or no information about two novel animals to see how this affected their attention to pictures of these animals in a dot-probe task. Field found that after verbal threat information children were more likely to attend to the animal about which they had received threat information. In the current data set (Rohani, Field, & Hutton, 2009), a similar design was used except that a visual search task was used rather than a dot-probe task: after different types of information children had to search for the animals in a jungle seen and respond to say whether they could find the animal or not (i.e., reaction time is the outcome variable). Children received 108 trails presented in two blocks of 54: in one block they searched for the animal about which they had received threat information, and in the other they searched for the animal about which they had received no information (the control). These data can be found in the file **Field & Wright (Visual Search).sav**, which is an SPSS data file. This data set has the following variables: (1) *particnu*, a numeric value that identifies each participant; (2) *block*, whether it was the first (0) or the second (1) block of trials; (3) *trial*, the trial order within each block (trial 1 is the first trial within a block, 54 is the last trial within a block); (4) *Target_Present*, whether the target (the animal) was present on the trial (1) or absent (0); (5) *Trial_Type*, whether the trial involved searching for the control animal (0) or the threat animal (1); and *logrt*, the natural log of their reaction time to respond (log reaction times were used to reduce skew).

These data are effectively a growth model in which we are looking at RTs over time; we are actually looking at RTs over different trials, but the data structure is the same as a time series design in that we would expect RTs close in time (i.e. on trials close together) to be more similar than RTs far apart in time (i.e., on distant trials). Essentially we predict that RTs should be affected by *Trial_Type* (children should be quicker to find the animal they believe is threatening) and *Target_Present* (children will be quicker to make a decision when the animal is actually in the picture than when it is absent). If we were ignoring the trials (for example if we used average RT across trials, as you might do if conducting an ANOVA) then our analysis would be a 2 (Trial Type: threat vs. control) × (Target_Present: present vs. Absent) design. In which we would have two main effects and an interaction term.

Importing the data

We have already initiated the libraries that we need in the previous example, so can load the file into a data frame named 'searchData' as in the previous example:

```
filename <- file.choose()
searchData <- read.spss(filename,use.value.labels=FALSE, to.data.frame=TRUE)
```

Fitting a baseline model

To run the multilevel model we again use the *lme()* function because this enables us to look at the covariance structure of the data. We can begin with a baseline model that simply ignores the effects of *Trial_Type* and *Target_Present*. This model is set up in the same way as before: the outcome is *logrt*, and it is predicted from only the intercept ('logrt ~1'), 'random = ~1|particnu means 'make the intercepts vary across participants (in other words, reaction times are nested within participants), we are using the data frame called *searchData* ('data = searchData'), and using the maximum likelihood method (method = "ML"), we also exclude any missing data ('na.action = na.exclude') although we have included this command only to remind you that it can be used, there are no missing data in this data set and we ignore this command in future models.

```
baseline<-lme(logrt~1, random = ~1|particnu, data = searchData, method = "ML", na.action =
na.exclude)
```

If we want to include the main effects of *Trial_Type* and *Target_Present* we simply add them as predictors: we change 'logrt~1' to be 'logrt~ Target_Present + Trial_Type'. If we want both the main effects and the interaction we can specify a full factorial model by simply changing 'logrt~1' to be 'logrt~ Target_Present*Trial_Type'. This code creates these two models (called *mainEffects* and *fullFactorial* respectively) and compares the three models with the *anova*() command (see earlier). Figure 6 shows the output of this model comparison, and it is clear that adding in these main effects and interactions significantly improves the fit of the model.

```
mainEffects<-lme(logrt~ Target_Present + Trial_Type, random = ~1|particnu, data = searchData,
method = "ML")
fullFactorial<-lme(logrt~ Target_Present*Trial_Type, random = ~1|particnu, data = searchData,
method = "ML”)
anova(baseline, mainEffects ,fullFactorial)
```

Comparing Models (Example 2)

```
              Model df      AIC      BIC    logLik   Test   L.Ratio p-value
baseline          1  3 7764.882 7783.751 -3879.441
mainEffects       2  5 6247.893 6279.342 -3118.946 1 vs 2 1520.9886  <.0001
fullFactorial     3  6 6240.082 6277.821 -3114.041 2 vs 3    9.8105  0.0017
```

anova(fullFactorial, autoRegressive)

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
fullFactorial      1  6 6240.082 6277.821 -3114.041
autoRegressive     2  7 6008.413 6052.442 -2997.206 1 vs 2 233.6695  <.0001
```

anova(fullFactorial, movingPoint1, movingPoint2)

```
               Model df      AIC      BIC    logLik   Test   L.Ratio p-value
fullFactorial     1  6 6240.082 6277.821 -3114.041
movingAverage1    2  7 6008.413 6052.442 -2997.206 1 vs 2 233.66946  <.0001
movingAverage2    3  8 5954.341 6004.660 -2969.171 2 vs 3  56.07164  <.0001
```

> anova(movingAverage2, trialModel1, trialModel2, trialModel3, trialModelFull)

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
movingAverage2     1  8 5954.341 6004.660 -2969.171
trialModel1        2  9 5863.222 5919.830 -2922.611 1 vs 2 93.11949  <.0001
trialModel2        3 10 5855.771 5918.669 -2917.885 2 vs 3  9.45081  0.0021
trialModel3        4 11 5856.826 5926.013 -2917.413 3 vs 4  0.94550  0.3309
trialModelFull     5 12 5852.259 5927.736 -2914.129 4 vs 5  6.56695  0.0104
```

> anova(trialModelFull, randomSlopes1)

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
trialModelFull     1 12 5852.259 5927.736 -2914.129
randomSlopes1      2 14 5588.877 5676.934 -2780.439 1 vs 2 267.3815  <.0001
```

> anova(trialModelFull, randomSlopes2)

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
trialModelFull     1 12 5852.259 5927.736 -2914.129
randomSlopes2      2 14 5837.130 5925.188 -2904.565 1 vs 2 19.12813   1e-04
```

> anova(trialModelFull, randomSlopes3)

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
trialModelFull     1 12 5852.259 5927.736 -2914.129
randomSlopes3      2 17 5571.706 5678.632 -2768.853 1 vs 2 290.5526  <.0001
```

Figure 6: Various model comparisons for Example 2 (see text for details).

Modelling covariance structures

We can look at modelling the covariances over time (*trial*). The default way of doing this is AR(1). This is the first order autoregressive covariance structure described earlier. We can include this structure by updating the model to include 'correlation=corAR1' and then specifying the form of the data. In this case we specify 'form = ~trial|particnu/block', which tells R that *trial* is our repeated measure and that it varies within participants (*particnu*) and block. Note that we have used the 'update' command, which in this case takes the model *fullFactorial* that we created earlier and updates it to include the AR(1) covariance structure. The update function is useful because it means less typing (and therefore fewer typing errors) and it uses information from the previously saved model to expedite estimating the parameters. Both of these can be important for complex models.

```
autoRegressive <- update(fullFactorial, correlation=corAR1(0, form = ~trial|particnu/Block))
```

In the same way, we can use other covariance structures. For example, we can look at a continuous autoregressive structure, which is fairly normal for a time variable, by using correlation=corCAR1:

```
continuousAR<- update(fullFactorial, correlation=corCAR1(.5, form = ~ trial|particnu/Block))
```

We can also look at a moving average correlation structure (the ARMA or Box-Jenkins model), in which *p* determines the autoregressive order (*p* = 1 is a first order, *p* = 2 is a second order and so on). This model is (sort of) a combination of an autoregressive and a continuous autoregressive covariance structure (see Singer & Willett, 2003, for a good discussion of CAR1 and ARMA covariance structures). Put simply, *p* = 1 means all adjacent correlations have the same correlation, ρ, and therefore those 2 apart have correlation ρ^2. If *p* = 2, then it allows for influence from time 1 to time 3 that is not mediated by time 2. Sometimes these higher order structures improve the model fit, but do not make theoretical sense. It is also fair to say that the interpretation of autoregressive orders above second order are lost on most of us mere mortals. These first and second order structures are defined as follows:

```
movingAverage1<- update(fullFactorial, correlation=corARMA(form = ~trial|particnu/Block,p=1))
movingAverage2<- update(fullFactorial, correlation=corARMA(form = ~trial|particnu/Block,p=2))
```

We can again compare these models with covariance structures to those without using ANOVA. In the following lines of code we compare AR(1) to our *fullFactorial* model, and then the two moving average covariance structures (again to the model that did not take account of the covariance structure):

```
anova(fullFactorial, autoRegressive)
anova(fullFactorial, movingAverage1, movingAverage2)
```

The output is in Figure 6 and clearly shows that modeling the covariance structure leads to a significant improvement in the model (this is true for AR(1) and the first and second-order moving average models).

Including the time series variable

Now that we know the kind of covariance structure that *trial* has, we can use these covariance structures to build up the model to include that variable. We choose one of the previous models and then add in *trial* as a covariate. We will choose the *movingAverage2* model because it had the best fit (it had the lowest BIC, see Figure 6). We again use the update command to tell R to update the model called *movingAverage2* by keeping everything the same (.~.) but adding the variable *trial* (+ trial).

```
trialModel1 <- update(movingAverage2, .~. + trial)
```

We can then add in the interactions between trial and our other predictors systematically. Note that each time we update the previous model by adding a new term.

```
trialModel2 <- update(trialModel1, .~. + trial:Target_Present)
trialModel3 <- update(trialModel2, .~. + trial:Trial_Type)
trialModelFull <- update(trialModel3, .~. + trial:Trial_Type:Target_Present)
```

We can again compare these models to see whether adding in each term significantly improves the model using *anova()*.

```
anova(movingAverage2, trialModel1, trialModel2, trialModel3, trialModelFull)
```

Figure 6 shows the comparison of these models. Each model is a significantly better fit than the previous one except for *trialModel3*. This suggests that *trial* significantly interacts with all other predictors apart from *Trial_Type*. (There are two things to note for this effect: (1) the trial × Trial_Type interaction is part of the higher-order three way interaction, which is significant, and (2) the BIC, which indicates the fit of the model to the data, is slightly smaller for the trial × Trial_Type interaction than the model that also includes the three-way interaction, indicating that it is a better fit of the data than when the three way interaction is included.)

Fitting a random slopes model

At the moment we still have only random intercepts in the model. We can add random slopes for the two predictors (again across participants) by again updating the model to include these terms. The code below creates three new models that start with *trialModelFull* (the full factorial model including the second order autoregressive structure) and updates it to add in random slopes *Target_Present* (*randomSlopes1*), *Trial_Type* (*randomSlopes2*) and both (*randomSlopes3*). The *anova()* commands then compare each of these models against the model without random slopes added.

```
randomSlopes1 <- update(trialModelFull, random= ~ Target_Present|particnu)
randomSlopes2 <- update(trialModelFull, random= ~ Trial_Type|particnu)
randomSlopes3 <- update(trialModelFull, random = ~ Target_Present + Trial_Type|particnu)
anova(trialModelFull, randomSlopes1)
anova(trialModelFull, randomSlopes2)
anova(trialModelFull, randomSlopes3)
```

Figure 6 shows the comparison of these models. Based on the BIC, the best fitting model is randomSlopes1, which is the full model but with slopes for *Target_Present* allowed to vary across participants. Figure 7 shows the output from this model, which is obtained using:

```
summary(randomSlopes1)
```

This output shows that all effects were significant; in other words, reaction times were significantly affected by the type of trial (threat or control animal) whether the target was present (or absent), and the interaction of these variables. In addition, these variables interacted significantly with trial, suggesting that their effects varied (became stronger or weaker) as the task progressed.

Summary

In this paper we have tried to convince you that clinical trial data, and data from tasks commonly employed in psychopathology research can be (and should be) thought of as hierarchical: patients are often treated in different clinics, and experimental tasks with multiple trials produce data that are nested within participants. The contexts within which data are collected create dependency between data points. For example, patients treated by the same clinician should produce data more similar than those treated by different clinicians. This dependency can be modelled by using multilevel models.

Using multilevel models not only liberate us from some assumptions of other tests (e.g. homogeneity of regression slopes) and overcome the need for full data sets, but they also enable us to look at whether effects are context dependent (i.e. do slopes or intercepts vary across contexts). The key concept in multilevel models is whether parameters are fixed (they remain constant across contexts) or random (they vary across contexts). In other respects, these models are simply a general linear model (i.e. regression) similar to the many regressions conducted by or studies by many readers. Extending what you know about regression to entertain the notion that regression parameters might not be fixed, but variable, across contexts is not difficult. However, conducting the analysis can be.

The latter half of the paper, therefore, offered a tutorial of multilevel models using the software R. The first example looked at an example of a clinical trial in which treatments were conducted at different clinics (i.e. participants were nested within clinics). The second example looked at a time series design in which multiple trials on a reaction time task were treated as being nested within participants. These are two very different examples that illustrate two quite common situations. Of course we could not discuss every aspect of using R to run multilevel models: we have barely scratched the surface. However, we hope that these tutorials will encourage readers to use and learn more about this powerful method.

```
> summary(randomSlopes1)
Linear mixed-effects model fit by maximum likelihood
 Data: searchData
       AIC      BIC    logLik
  5588.877 5676.934 -2780.439

Random effects:
 Formula: ~Target_Present | particnu
 Structure: General positive-definite, Log-Cholesky parametrization
               StdDev    Corr
(Intercept)    0.2844959 (Intr)
Target_Present 0.3136253 -0.474
Residual       0.4908282

Correlation Structure: ARMA(2,0)
 Formula: ~trial | particnu/Block
 Parameter estimate(s):
     Phi1      Phi2
0.2263254 0.1412544
Fixed effects: logrt ~ Target_Present + Trial_Type + trial +
Target_Present:Trial_Type +      Target_Present:trial + Trial_Type:trial +
Target_Present:Trial_Type:trial
                                    Value  Std.Error   DF   t-value p-value
(Intercept)                      8.843951 0.05469995 3929 161.68116  0.0000
Target_Present                  -0.477579 0.06287402 3929  -7.59581  0.0000
Trial_Type                       0.104100 0.05039686 3929   2.06561  0.0389
trial                           -0.004289 0.00112812 3929  -3.80174  0.0001
Target_Present:Trial_Type       -0.246645 0.06094891 3929  -4.04674  0.0001
Target_Present:trial            -0.005387 0.00135123 3929  -3.98696  0.0001
Trial_Type:trial                -0.003257 0.00159096 3929  -2.04750  0.0407
Target_Present:Trial_Type:trial  0.005225 0.00192203 3929   2.71854  0.0066
 Correlation:
                                (Intr) Trgt_P Trl_Ty trial  Tr_P:T_T Trg_P: Trl_T:
Target_Present                  -0.484
Trial_Type                      -0.459  0.241
trial                           -0.564  0.299  0.614
Target_Present:Trial_Type        0.229 -0.483 -0.504 -0.311
Target_Present:trial             0.287 -0.597 -0.313 -0.512  0.619
Trial_Type:trial                 0.400 -0.212 -0.867 -0.710  0.443    0.364
Target_Present:Trial_Type:trial -0.202  0.420  0.443  0.361 -0.874   -0.703 -0.512

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.84064730 -0.64472095 -0.04676945  0.57494100  4.59444023

Number of Observations: 3983
Number of Groups: 47
```

Figure 7: Output for the final model in Example 2 (see text for details).

Further Reading

An extensive list of books on conducting MLMs using various software packages can be found at http://www.cmm.bristol.ac.uk/learning-training/multilevel-m-support/books.shtml. There are several books that we specifically recommend for beginners, (Bickel, 2007; Kreft & de Leeuw, 1998; Twisk, 2006), for analysing longitudinal data (Singer & Willett, 2003), and for intermediate to more advance readers (Goldstein, 2003; Hox, 2010; Raudenbush & Bryk, 2002; Snijders & Bosker, 1999).

References

Bates, D., & Maechler, M. (2010). lme4: Linear mixed-effects models using S4 classes. R package (Version 0.999375-37). Retrieved from http://CRAN.R-project.org/package=lme4

Beidas, R. S., & Kendall, P. C. (2010). Training Therapists in Evidence-Based Practice: A Critical Review of Studies From a Systems-Contextual Perspective. *Clinical Psychology-Science and Practice, 17*(1), 1-30. doi:10.1111/j.1468-2850.2009.01187.x

Bickel, R. (2007). *Multilevel Analysis for Applied Research: It's Just Regression.* . New York: NY: Guilford Press.

Dennis, T. A., Malone, M. M., & Chen, C. C. (2009). Emotional Face Processing and Emotion Regulation in Children: An ERP Study. *Developmental Neuropsychology, 34*(1), 85-102. doi:10.1080/87565640802564887

Donnelly, N., Hadwin, J. A., Manneer, T., & Richards, H. (2010). The use of visual search paradigms to understand attentional biases in childhood anxiety. In J. A. Hadwin & A. P. Field (Eds.), *Information processing biases and anxiety: a developmental perspective.* (pp. 109-127). Chichester: Wiley-Blackwell. doi:10.1002/9780470661468.ch5

Enders, C. K., & Tofighi, D. (2007). Centering predictor variables in cross-sectional multilevel models: A new look at an old issue. *Psychological Methods, 12*(2), 121-138. doi: Doi 10.1037/1082-989x.12.2.121

Field, A. P. (2006a). The behavioral inhibition system and the verbal information pathway to children's fears. *Journal of Abnormal Psychology, 115*(4), 742-752. doi:10.1037/0021-843X.115.4.742

Field, A. P. (2006b). Watch out for the beast: Fear information and attentional bias in children. *Journal of Clinical Child and Adolescent Psychology, 35*(3), 431-439. doi:10.1207/s15374424jccp3503_8

Field, A. P. (2009). *Discovering statistics using SPSS: And sex and drugs and rock 'n' roll* (3rd ed.). London: Sage.

Field, A. P., & Miles, J. N. V. (2010). *Discovering statistics using SAS: And sex and drugs and rock 'n' roll*. London: Sage.

Field, A. P., & Miles, J. N. V. (In Press). *Discovering statistics using R: And sex and drugs and rock 'n' roll*. London: Sage.

Field, A. P., & Schorah, H. (2007). The verbal information pathway to fear and heart rate changes in children. *Journal of Child Psychology and Psychiatry, 48*(11), 1088-1093. doi:10.1111/j.1469-7610.2007.01772.x

Goldstein, H. (2003). *Multilevel statistical models* (3rd ed.). London: Arnold.

Hadwin, J. A., & Field, A. P. (2010). *Information processing biases and anxiety: a developmental perspective.* Chichester: Wiley-Blackwell. doi:10.1002/9780470661468

Hamilton, B. L. (1977). Empirical-investigation of effects of heterogeneous regression slopes in analysis of covariance. *Educational and Psychological Measurement, 37*(3), 701-712. doi:10.1177/001316447703700313

Hollingsworth, H. H. (1980). An analytical investigation of the effects of heterogeneous regression slopes in analysis of covariance. *Educational and Psychological Measurement, 40*(3), 611-618. doi:10.1177/001316448004000306

Hox, J. J. (2010). *Multilevel Analysis: Techniques and Applications.* (2nd ed.). Hove: Routledge.

Hughes, A. A., & Kendall, P. C. (2007). Prediction of cognitive behavior treatment outcome for children with anxiety disorders: Therapeutic relationship and homework compliance. *Behavioural and Cognitive Psychotherapy, 35*(4), 487-494. doi:10.1017/S1352465807003761

Kendall, P. C., Comer, J. S., Marker, C. D., Creed, T. A., Puliafico, A. C., Hughes, A. A., et al. (2009). In-Session Exposure Tasks and Therapeutic Alliance Across the Treatment of Childhood Anxiety Disorders. *Journal of Consulting and Clinical Psychology, 77*(3), 517-525. doi:10.1037/a0013686

Kreft, I. G. G., & de Leeuw, J. (1998). *Introducing multilevel modeling*. London: Sage.

Kreft, I. G. G., de Leeuw, J., & Aiken, L. S. (1995). The effect of different forms of centering in hierarchical linear models. *Multivariate Behavioral Research, 30*, 1-21. doi:10.1207/s15327906mbr3001_1

Öhman, A., Esteves, F., & Soares, J. J. F. (1995). Preparedness and Preattentive Associative Learning - Electrodermal Conditioning to Masked Stimuli. *Journal of Psychophysiology, 9*(2), 99-108.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, S., & R Development Core Team. (2010). nlme: Linear and Nonlinear Mixed Effects Models. R package version (Version 3.1-97). Retrieved from http://CRAN.R-project.org/package=lme

R Development Core Team. (2010). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

R Development Core Team, Sarkar, S., DebRoy, S., Bivand, R., & Others. (2010). foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase. R package version (Version 0.8-41). Retrieved from http://CRAN.R-project.org/package=foreign

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods.* Thousand Oaks, CA: Sage.

Riddle, M. A., Reeve, E. A., Yaryura-Tobias, J. A., Yang, H. M., Claghorn, J. L., Gaffney, G., et al. (2001). Fluvoxamine for children and adolescents with obsessive-compulsive disorder: A randomized, controlled, multicenter trial. *Journal of the American Academy of Child and Adolescent Psychiatry, 40*(2), 222-229. doi:10.1097/00004583-200102000-00017

Rohani, S. S., Field, A. P., & Hutton, S. (2009). *Acquisition of Fear and Attentional Bias in Children: an Eye Tracking Study.* Paper presented at the Annual Congress of the European Association for Behavioural and Cognitive Therapies, Dubrovnik, Croatia.

Singer, J. D., & Willett, J. B. (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford: OUP.

Snijders, T. A. B., & Bosker, R. J. (1999). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling*. London: Sage.

Southam-Gerow, M. A., Silverman, W. K., & Kendall, P. C. (2006). Client similarities and differences in two childhood anxiety disorders research clinics. *Journal of Clinical Child and Adolescent Psychology, 35*(4), 528-538. doi:10.1207/s15374424jccp3504_4

Twisk, J. W. R. (2006). *Applied multilevel analysis: A practical guide*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511610806

Walkup, J. T., Compton, S. N., & Kendall, P. C. (2009). Behavioral Therapy, Sertraline, or Both in Childhood Anxiety Reply. *New England Journal of Medicine, 360*(23), 2477-2477.

Wright, D. B., & London, K. (2009). *Modern Regression Techniques Using R: A Practical Guide*. London: Sage.

Wu, Y. W. B. (1984). The effects of heterogeneous regression slopes on the robustness of 2 test statistics in the analysis of covariance. *Educational and Psychological Measurement, 44*(3), 647-663. doi:10.1177/0013164484443011

Yang, X. W., Li, J. H., & Shoptaw, S. (2008). Imputation-based strategies for clinical trial longitudinal data with nonignorable missing values. *Statistics in Medicine, 27*(15), 2826-2849. doi:10.1002/sim.3111