

How to Identify and Specify Fixed and Random Effects for Linear Mixed Models in the Statistical Analysis System

Will G Hopkins

Sportscience 27, 4-25, 2023 (sportsci.org/2023/EffectsModels.htm)

Internet Society for Sport Science, Auckland, New Zealand. [Email](#). Reviewer: Basilio Pueo, Faculty of Education, University of Alicante, Spain.

Most analyses require linear mixed modeling to properly account not only for mean effects but also for sources of variability and error in the data. In linear mixed models, means and their differences are specified with fixed effects, while variabilities and errors are specified with random effects (including residuals) and are summarized as standard deviations. In this tutorial article, I explain how variables represent effects in linear mixed models and how identifying clusters of observations in a dataset is the key to identifying the fixed and random effects. I also provide 15 programs written in the language of the Statistical Analysis System (SAS) to simulate data and analyze them with the general linear mixed model, Proc Mixed, which is used when the dependent variable is continuous. The analyses include simple linear regression, comparisons of group means and standard deviations, reliability, time series and controlled trials with individual responses, and within-subject modeling (random intercepts and slopes). Finally, I explain how dependent variables representing counts or proportions require generalized linear mixed models, where the main difference is the specification of the residuals.

Keywords: confidence limits, data, general linear model, generalized linear model, numeric and nominal variables, residual error, simulation.

[Reprint pdf](#) · [Reprint docx](#) · [Programs zip file](#)

Introduction	4
Datasets	6
Numeric and Nominal Variables	6
Effects and Models	6
Clusters (Groups, Subsets) of Observations	7
Fixed Effects	8
Random Effects and Residuals	9
Groups of Residuals	10
Repeated Measurement	10
Failed Analyses	12
Standard Errors	13
Repeated Measurement with Dummy Variables	14
Repeated Measurement with Type=un	15
Groups of Random Effects	17
Controlled Trial with Two Trials	17
Controlled Trial with Three Trials	18
Controlled Trials via Change Scores	19
Random Effects with Multiple Clusters	20
Within-Subject Linear Modeling	22
Generalized Linear Mixed Models	24
References	25

Other resources on mixed modeling at this site: a self-paced [workshop](#) introducing SAS Studio

and basic mixed models; a short [slideshow](#) on mixed models, included in the workshop.

Introduction

This article began as a series of informal Skype tutorials for a colleague who wanted to use mixed modeling. He was envious of my intuitive ability to identify the fixed and random effects for any given study, and to specify the fixed- and random-effect models in the Statistical Analysis System (SAS). That intuitive ability is probably a result of over 35 years of working with such models in SAS, so we were hoping to develop an algorithmic or diagrammatic short-cut for his benefit and for other younger researchers.

The short-cut proved elusive, but I came up with a fresh look at mixed modeling, based on identifying clusters in the data. This approach is almost an algorithm and should work for most newbies. With them and/or you in mind, I start the explanation by going back to the basics of what we mean by data, variables, effects, and models, then I identify the effects and build the models for various analyses where general linear mixed modeling reigns supreme. To make sure the models worked, and to enhance understanding (sometimes my own, even now, as you will see), I simulated data for each analysis. The SAS code to simulate and analyze the data is included in this article as links (below some section headings) to 15 programs that run in the full SAS package or in the free cloud version of [SAS Studio](#).

Each program opens as plain text in a browser. You can either copy and paste the text into a new program in SAS Studio, or download all the programs as this [zipped file](#), unzip it, and upload them all at once into an appropriate new folder in SAS Studio. Run a given program first without any modification. Each is written with macro variables whose values you can then change to see what happens (and thereby enhance understanding) with different sample sizes, means and SDs, effects, errors of measurement, individual responses, and so on.

The dependent variable in all but two of the programs is maximum oxygen uptake (VO₂max), with typical values of around 45 for kids with a range of ages around 13 y. (The variable has the name ObsvdVO₂max, for *observed* VO₂max, in the programs, to distinguish it from TrueVO₂max in the code that generates the simulated data.) For the benefit of non-exercise and non-sport scientists, VO₂max is the maximum rate at which oxygen can be consumed in exer-

cise, here in units of milliliters of oxygen per minute per kilogram body mass. The values are already natural-log transformed and multiplied by 100, which transforms all effects and standard deviations (SDs) to slightly less than the original percent values, when they are ~10% or less. In other words, the numbers you see in the simulations for SDs, differences in means, and slopes (differences per unit of a numeric predictor) are effectively in percent units, and the original raw values are irrelevant. Most variables are like VO₂max, in that they need log transformation for analysis, so this approach to the simulations should make it easy for you to choose SDs and effects in percent units for other such variables when you run the programs. The simulations work just as well for data that don't need log transformation, because I haven't back-transformed the log-transformed effects and SDs to percents or factors. With real data requiring log transformation, you must back-transform.

The focus is the fixed-effect and random-effect models, which, for reasons of clarity, I present in this document with the minimum of options. The SAS code in the accompanying programs includes more options and other stuff with little or no explanation, including simulations, scatterplots, line diagrams and/or bar graphs of the original data, and scatterplots of residuals vs predicted and residuals vs predictors, where relevant. If you haven't used SAS Studio before, follow [this link](#) (Hopkins, 2022a) to a workshop focused more on the practicalities of the use of SAS Studio for mixed models. Download and unzip the workshop, open Read me first.docx, and follow the instructions therein. You may need to work through some of the workshop documents to get a better understanding of the relevant SAS coding or to use SAS in a point-and-click fashion.

The programs also don't include coding to deal with sampling uncertainty beyond confidence limits. If you want to use these programs with your own data, I suggest you insert the resulting effects and confidence limits into my spreadsheet [Combine/compare effects](#) (Hopkins, 2006) on the tab for 1 or more groups or statistics. When you also insert a smallest important value (if necessary, expressed as a factor and log-transformed), the spreadsheet generates probabilities for trivial and substantial magnitudes that can be interpreted in a Bayesian fashion as probabilities of the true magnitude, or the probabilities can be used as p values for testing substantial and non-

substantial hypotheses (inferiority, superiority, and equivalence testing). Note that the smallest and other magnitude thresholds for standard deviations are half those for mean effects (Smith & Hopkins, 2011). Please, don't test the nil hypothesis, don't show the corresponding traditional p value, and don't state whether the effect is statistically significant or non-significant. I've dealt with this issue of sampling uncertainty extensively [elsewhere](#) (Hopkins, 2022b).

The programs all use Proc Mixed in SAS to realize a general linear mixed model, which is appropriate for a continuous dependent variable. I haven't provided programs for dependent variables that need *generalized* linear mixed models, with Proc Glimmix or Proc Phreg in SAS. The main differences with modeling such variables are the specification of the residuals and the interpretation of the effects and SDs as count, odds and hazard ratios, which I do explain in the final section.

Machine learning (or neural-net modeling) is another approach to data analysis, but this approach works only with large datasets, and it does not perform well with repeated measurements. See the section on [monitoring](#) in the 2021 ECSS conference report for more.

To all users of the R stats package, my apologies: a few years ago, I found the mixed model in R too limiting, and I struggled with the coding. I hope someone with R smarts will consider adapting my programs and publishing them.

Datasets

A **dataset** consists of rows of data in a spreadsheet, where the first row is the names of the variables, and the subsequent rows are the values of the variables. In SAS, each row of values is called an **observation**: the single values of all the variables for one **subject** on one testing or assessment occasion. The subject is usually a person, but a dataset might consist of rows of values of something else, such as a sports match. In mixed modeling, the data must be in "long" format; that is, any repeated measurements on subjects must go *down* the spreadsheet with an appropriate variable to identify the repeats (e.g., Trial, with values *pre* and *post*). Repeated measurements must not go *across* the spreadsheet (i.e., in "wide" format) with a different variable for each repeat (e.g., VO2maxPre, VO2maxPost); convert all such repeats to long format, either with paste-transpose in Excel or with a data step in SAS.

Numeric and Nominal Variables

Identify the **numeric variables** in your dataset; for example, Age (e.g., in years), Weight (e.g., in kg), and VO2max (e.g., in ml/min/kg). These are sometimes known as **continuous** variables, which means the numbers can have an infinite number of decimal places. A variable with integers as values, such as a count of injuries, is not strictly continuous, but it is numeric and treated as continuous in most analyses (e.g., its mean can have a decimal place: 2.6 injuries per player per season).

Identify the **nominal variables** in your dataset; for example, ChildID, Sex, Country. Nominal variables have names as values. The different values are also known as **levels**. There must be at least one nominal variable in the dataset: the identity of each observation (which doubles as the identity of the subjects, when there is no repeated measurement). Nominal variables might have numbers to identify the levels, but the numbers are to be treated as character strings. Or to put it another way, if you can replace the numbers with names of people or things, you can keep the numbers, but the variable is to be treated as nominal. With mixed modeling in SAS, you must declare nominal variables as such using a **class** statement (short for *classification variable*), regardless of whether the levels are numbers or names. Note also that you can create a nominal variable from a numeric variable by parsing the numeric values into groups; for example, convert Age into AgeGroup, with levels *prepubertal*, *peripubertal*, *postpubertal*.

Effects and Models

Important research is all about **effects**; for example, what is the effect of sex on fitness, what is the effect of X on Y, or more generally, what is the effect of a **predictor variable** on a **dependent variable**? The way in which a predictor variable affects a dependent variable is quantified with a **model** or equation, in which the values of a dependent variable are equal to some function of the values of one or more predictor variables. Almost all models are **linear**, of the form $Y = a + b \cdot X_1 + c \cdot X_2 + \dots$, where Y is the dependent variable and X_1, X_2, \dots are the predictor variables. The a, b, c, ... are called **parameters** or **coefficients**, and their values are estimated by the stats package. The model predicts the value of the dependent variable for any given values of the predictor variables.

If the dependent variable is continuous, the

model is a **general linear model**, including **simple linear regression** (one numeric predictor), **multiple linear regression** (two or more numeric predictors), **analysis of variance** (one or more nominal predictors) and **analysis of covariance** (one or more nominal and one or more numeric predictors). If the dependent variable is a count or a proportion, the model is known as a **generalized linear model**, of which there are several: **Poisson regression** (for counts); **logistic, log-odds, binomial or multinomial regression** (for proportions); and **log-hazard, proportional-hazards, or Cox regression** (for proportions per unit time).

The first term in a linear model is a constant called the **intercept**; it really is the Y intercept in a simple linear regression, which represents a straight line, $Y = a + b \cdot X$. The effect of each predictor variable on the dependent variable is given by its coefficient (b, c, and so on). Each coefficient is a **slope**; its units are the units of Y per unit of X. In other words, if you change X by one unit, you change Y by an amount given by the coefficient of X. The effect of X *is* its coefficient, but we also refer to X itself as an effect: **predictor variables are also called effects**.

Nominal effects are represented in a linear model by **dummy variables**, one for each level of the variable. Each dummy variable has only two values, 0 and 1. For example, Sex might be represented by three dummy variables: Female (say), having the value 1 if the observation comes from a female and 0 otherwise; Male (say), having the value 1 if the observation comes from a male, and 0 otherwise; and Other (say), having the value 1 if the observation comes from something other than a female or male, and 0 otherwise. Stats packages do this dummy coding for you; all you have to do is specify that the variable is nominal. If there is at least one nominal effect in the model, you don't need an intercept, because each level of a nominal effect represents an intercept. If you include an intercept, it becomes one of the levels of the nominal effect, and that level is effectively removed by giving it a coefficient of zero. (You don't have to understand this process; it's done automatically by the stats package.) Notice that effects of nominal variables are still slopes. For example, the effect of being male is the difference in Y per unit of the dummy variable Male. But the dummy variable has only two values, 0 and 1, so the effect on Y per unit of the dummy variable is the difference between being a male

(Male=1) and not being a male (Male=0).

When you have more than one predictor variable in a linear model, the effect of each predictor is its effect *when you hold the other predictors constant*. In other words, it's the effect uncontaminated by differences or changes in the other variables. This interpretation is an inescapable and wonderful consequence of the fact that the model is a sum of the predictor variables multiplied by their coefficients. Each effect is said to be **adjusted for** or **conditioned on** all the other effects.

Clusters (Groups, Subsets) of Observations

[01 Simple linear regression.sas](#)

[02 Comparison of means.sas](#)

[03 Comparison of means+covariate.sas](#)

Almost all analyses involve **clusters** (i.e., **groups** or **subsets**) of observations defined by nominal variables. For example, Sex can define two or more clusters: the subset of observations for boys, the subset for girls, and the subset(s) for any of the LGBTQIA+ designations. You estimate means of the clusters (e.g., the mean for boys and the mean for girls), differences in the means of the clusters (e.g., the mean for boys minus the mean for girls, which is the effect of Sex), standard deviations representing differences between the values within each cluster (e.g., the SD for the boys and the SD for the girls), and standard deviations representing differences between the means of each cluster, when there are enough of them and the nominal variable is a random effect—see later.

If you have only one nominal variable in the dataset, for example ChildID, then you can't have any clusters, because there is only one observation for each child, and there is no way to get a cluster for each child or clusters of different kinds of children. But you can still do an analysis with the numeric variable(s). For example, what is the effect of Age on VO2max? The simplest analysis for this effect would be a simple linear regression. In SAS you specify this analysis with this simple statement:

```
model VO2max=Age;
```

This model represents the following equation of a straight line:

$$\text{VO2max} = \text{Intercept} + \text{Slope} \cdot \text{Age}.$$

You don't have to state Intercept in the model, because SAS assumes there is always an intercept in a linear model, and it estimates it by default. And you don't have to state Slope* in front of Age, because SAS assumes that you want the

slope (that's what the analysis is all about—the effect of Age as a slope), and it estimates it as a coefficient.

Analyses get more interesting when you have one or more nominal variables additional to the nominal variable representing each observation in the dataset. If you have Sex, each level of Sex represents clusters of different kids. If you have AgeGroup, each level of AgeGroup represents clusters of different kids, and so on. So, once you have *two* nominal variables (e.g., ChildID plus something else), you can have clusters for one of them: the variable that has more than one observation for each of its levels. (Some, but not all, levels can have just one observation, but don't worry about that.) You can now do an analysis formerly known as an analysis of variance (ANOVA), by predicting a dependent variable (e.g., VO2max) with a nominal variable (e.g., Sex). The linear model predicts the mean VO2max for each level of Sex. In SAS, you do it with this statement:

```
model VO2max=Sex;
```

SAS now estimates two or more "intercepts": one for each level of Sex. Here they are just the means for each level.

When you have *two* additional nominals, you can have not only two different kinds of cluster but also a cluster of clusters. For example, if you have ChildID, Sex and AgeGroup, then you can have clusters of different sexes, clusters of different age groups, and clusters of different sexes within each age group (or equivalently, clusters of different age groups within each sex). You represent clusters of clusters with an **interaction** of nominal variables, in this case Sex*AgeGroup: if Sex has two levels, and AgeGroup has three levels, there are up to $2*3=6$ different clusters identified by the levels of Sex and AgeGroup: *boys prepubertal*, *boys peripubertal*, *boys postpubertal*, *girls prepubertal*, *girls peripubertal*, *girls postpubertal*. You can specify the interaction equally as AgeGroup*Sex, in which case the clusters would be *prepubertal boys*, *prepubertal girls*, *peripubertal boys*, *peripubertal girls*, *postpubertal boys*, *postpubertal girls*. Again, you can do an ANOVA, in which you predict a dependent variable with Sex*AgeGroup. Actually, all stats packages allow you to predict the dependent variable with the two nominal variables as well as their interaction. In SAS it would be:

```
model VO2max=Sex AgeGroup
Sex*AgeGroup;
```

In such a model, Sex and AgeGroup are known as **main effects**. With this model, you can use the **least-squares means (lsmeans)** statement to easily get the overall means for the levels of Sex and the difference between the two means, the overall means for the levels of AgeGroup and the three pairwise difference between the three means, and the six means represented by Sex*AgeGroup and all 15 of the pairwise differences between the six means, if you want them. If you just specify Sex*AgeGroup as the predictor, you can still get all those means and comparisons of means, but it takes a bit more programming to get some of them (in SAS, using **estimate** and/or **lsmestimate** statements).

You can add numeric variables to an ANOVA to get what was formerly known as an analysis of covariance (ANCOVA). A numeric variable is known as a **covariate** in such an analysis. As an example of the simplest ANCOVA, predict VO2max with Sex (nominal) and Age (numeric). If these two variables are in the model as main effects only, you effectively fit two straight lines in the scatterplot of VO2max vs Age, with the same slope (specified by Age) and different intercepts for each sex (specified by Sex). If you add the interaction Sex*Age, you get different slopes as well as different intercepts for each sex. You can leave out the main effect for Age, but either way, you must use **estimate** statements to get the slope averaged over the boys and the girls (if you want that). Don't ever leave out the main effect for Sex: I can't imagine a real dataset where there would be only one intercept but different slopes.

Fixed Effects

In the previous examples, the nominal variables Sex and AgeGroup had levels that would not change, if you were to repeat the study. Or to put it another way, those levels would always be represented, if you were to repeat the study. Hence the levels are fixed, and these variables are therefore called **fixed effects** in the linear model. (They can also contribute to random effects—see later.) Fixed effects normally have just a handful of levels.

It's important to identify all the nominal variables in your dataset that are fixed effects. Here are more examples of such variables: Trial in a time series or controlled trial (with levels such as *pre*, *mid*, *post*); Group in a comparison of two or so groups of individuals (e.g., *sedentary*, *active*); and Treatment in a controlled trial or crossover

(e.g., *control*, *experimental*).

Numeric variables such as Age are always fixed effects in a linear model, because they are in the model as a single variable. Age is a bit like a nominal variable with only one level and therefore only one dummy variable, except that the variable has the values of the numeric variable (12.3, 14.6...), not just 0 and 1.

Any interaction of fixed effects is also a fixed effect, including the interaction of nominal fixed effects with numeric variables (e.g., Sex*Age). A numeric variable like Age can also be entered in the model as itself and as interactions with itself, if a **polynomial** non-linear effect of the variable is required; for example a cubic in Age would be specified with Age Age*Age Age*Age*Age. Again, these are all fixed effects.

Random Effects and Residuals

Some nominal variables have levels that are a random sample of values. Or to put it another way, if you were to repeat the study with a different sample, all the levels of the nominal variable would change. The simplest example is the nominal variable representing the identity of the subjects in the dataset: do the study again with a new sample and all the subjects will be different. Such variables are **random effects**; they have usually more than just a handful of values in your sample, and they have an infinite number of values in the population represented by your sample.

It's important to identify all the nominal variables in your dataset that are random effects and to include them appropriately in the linear model. Why? Because you want the results of the analysis to apply to all possible levels of the variables: all children, athletes, teams, matches, and so on. If you include a nominal variable as a fixed effect, the analysis applies only to the levels of the variable in your dataset.

In the previous examples of analyses with fixed effects, ChildID was actually a random effect. But each child was represented only once in the dataset, so there were no clusters of observations within each child. There was therefore no need to specify ChildID in the model, because there is no mean value to calculate for each child. If you tried to specify ChildID as a fixed effect (or a random effect—see below), you would get an error. Why? Because the different values of ChildID appear only once, so specifying it as an effect is asking the linear model to estimate a mean for each level of ChildID. The analysis

can't even get started, because the "mean" for each child is simply the child's observed value, so there is nothing to calculate. Of course, you can estimate means for Sex and AgeGroup, but only if ChildID is not included.

ChildID was not included in the previous analyses, but there is a sense in which it was there all the time. Think about a fixed-effect model, say **VO2max=Sex**. This model predicts two mean values, one for boys and one for girls. But each boy has a value of VO2max that is different from the mean for boys, and ditto each girl; that is, each observed value is different from the mean. The difference between the observed value and the predicted mean value is an error that the model makes about predicting each individual boy and each individual girl. That error is called a **residual** in the analysis: it's the observed value of the dependent variable for each observation minus the predicted value for that observation. In a simple linear regression, multiple linear regression, ANOVA or ANCOVA, the means and slopes are estimated by finding their values that minimize the residuals. Actually, the sum of the squares of the residuals is minimized, and the mean of the residuals is always zero. What you end up with is the best-fitting or **least-squares** model. Fine, so what about ChildID in the model? Each residual is identified uniquely by ChildID: each level of ChildID occurs only once, and each level has a different residual. If you specify ChildID as a random effect, you are specifying the residual, which is forbidden with the random statement, because Proc Mixed has to estimate separate residuals, and there's nothing left to estimate them (so you get weird values—try it and see).

The residuals are not just a nuisance; they have a useful interpretation. For example, in the model **VO2max=Sex**, every boy (and every girl) has either a positive or a negative residual (no residual is ever exactly zero). A boy with a large positive residual has a VO2max that is much greater than the mean for the boys; a girl with a small negative residual has a VO2max that is a bit less than the mean for the girls; and so on. The residuals therefore represent relative **individual differences** from the mean of the boys and from the mean of the girls. Furthermore, the standard deviation of the residuals, the **root-mean-square error** (RMSE) in the language of ANOVA, is a single statistic representing the typical individual differences between boys (and between girls). In SAS, the scatter in the residuals is summarized as a variance, and you have to

take the square root to express the scatter as an SD.

Groups of Residuals

When you have two or more groups of subjects, the residuals could show a different scatter in each group. In the example of boys and girls tested for VO2max once each, there might be a bigger range (more properly, a bigger *scatter*) of fitness among the girls than among the boys. In SAS, there is a simple way to allow for this possibility:

```
repeated/group=Sex;
```

Now you will see two residual variances in the results window, one for each level of Sex. Previously, `repeated;` (i.e., without anything else) was implicit in the random-effects model: the model produced only one residual variance.

The `repeated` statement has been devised presumably to make it easier to deal with data that involve repeated measurement on subjects—more about this shortly. Meantime, think about `repeated` as a way of structuring the residuals, even when there is no repeated measurement. In fact, you can actually use the `repeated` statement to specify the residuals without getting into trouble. In the present example, ChildID is a random effect representing the residuals, and SAS doesn't mind if you state...

```
repeated ChildID;
```

or...

```
repeated ChildID*Sex;
```

...for only one residual variance, or...

```
repeated ChildID/group=Sex;
```

or...

```
repeated ChildID*Sex/group=Sex;
```

...for two residual variances.

The above simple mixed model with two residual variances is doing nothing more than estimating and comparing the mean VO2max of the boys and girls. You could do that with the t-test procedure in SAS or even in a spreadsheet, but what kind of t test is it? The so-called [unequal-variances t test](#): the unequal variances are the variances of the residuals for the boys and for the girls.

If you have AgeGroup and Sex in the fixed-effects model, you specify a different residual variance for each age group with this statement:

```
repeated/group=AgeGroup;
```

Or specify six residual variances with this statement:

```
repeated/group=Sex*AgeGroup;
```

As already noted, Proc Mixed provides a comparison of the means of the groups in several

ways, but it does not provide a comparison of the SDs. Here the SDs are independent residuals, so expressed as variances, their ratios can be compared in a pairwise fashion via the F statistic (the basis of the Levene test for homogeneity of variance). I haven't provided any coding or spreadsheet for such comparisons, but with even modest sample sizes, residual SDs are approximately log-normally distributed; hence you can compare their ratio with my spreadsheet [Combine/compare effects](#) (Hopkins, 2006) via the tab for 2 groups or statistics. You will need to insert the smallest important difference in SDs, which is half that for differences in means (Smith & Hopkins, 2011), as a factor of the reference SD.

Repeated Measurement

[04 Reliability 2 trials.sas](#)

Two (or more) measurements on the same subjects can be represented by a nominal variable such as Trial, with levels 1 and 2 (or *1st* and *2nd*, or *pre* and *post*, or *control* and *experimental*). Trial is, of course, a fixed effect, representing two clusters of measurements, and if you put it in the model, you can estimate the means for each trial and the change in the mean between the trials:

```
model VO2max=Trial;
```

ChildID now represents clusters of measurements; if there are 10 children, then there are 10 clusters, each cluster consisting of two observations. Hence ChildID now must be included in the model to account for the clustering. One way to include it would be as a fixed effect:

```
model VO2max=Trial ChildID;
```

The result would be a two-way ANOVA, with Trial and ChildID being the "two ways", and it's the traditional approach to a straightforward reliability analysis, as in my [spreadsheets for reliability](#). The residual provides the typical error of measurement.

But ChildID is really a random effect, so it should be included with a `random` statement:

```
random ChildID;
```

Although it's stated separate from the fixed effects, it is still a nominal variable in the model, so it is represented by dummy variables, one for each child. Therefore, ChildID estimates a mean value for each child, but it's not the actual mean of the two trials for each child, it's a *relative* mean: the contribution that each child makes to the linear model, when the means for the two levels of trial have been accounted for with the fixed effect for Trial. You can see the individual values estimated for each child by requesting the

random-effect solution for ChildID...

```
random ChildID/solution;
```

...or simply...

```
random ChildID/s;
```

The solution shows values for each child as a positive or negative number, representing the extent to which each child has a mean above or below the grand mean of the trials. The mean of the values of the random-effect solution is zero, just like the mean of the residuals is zero, and the scatter in the positive and negative values is summarized as a standard deviation representing typical differences between children averaged over the two trials. As with the residuals, the scatter is summarized as a variance, and you have to take the square root.

I like to use what I call the **hat metaphor** to understand random effects. When a variable such as ChildID is a random effect, it's as if there is a hat containing an infinite number of pieces on paper, on each of which is written a unique name of a child and a unique numeric value for that child. When you perform a study, the random sample of n kids is obtained by drawing n pieces of paper out of the hat. The mean of the infinite number of values in the hat is zero, and the SD of the values is the population value of the between-subject SD. The mixed model estimates the value for each kid on the pieces of paper in the sample, and the SD of the values is the sample SD. The metaphor is not perfect, because the mean of a random sample of values can never be zero, but the mixed model estimates the individual values partly by making the mean of all the sample values zero. For a slideshow with a succinct summary of mixed modeling and animations illustrating the hat metaphor, follow [this link](#).

An alternative specification of **random ChildID** is as follows...

```
random Intercept/subject=ChildID;
```

...or simply...

```
random Int/subject=ChildID;
```

Why "Intercept"? Recall that for a nominal fixed effect, each level of the fixed effect represents an intercept in a simple linear regression. The same thing happens with a random effect: each level of the random effect (here, each level specified by **subject=ChildID**) is an intercept in a linear regression for each child. It's not the exact intercept, because the mean value for all the intercepts is zero. You have to add it to the fixed-effect intercept(s) to get the actual intercept for each subject.

Now that I have introduced the **random** statement, I will introduce the concept of **negative variance** and **negative SD**. A variance is an SD squared, and the square of any number is always positive, even when the number is negative, yet you should almost always allow the variances estimated with the **random** statement to be negative. Why? Because the mixed model estimates the variances by partitioning the overall variance of the data (the grand SD squared) into various sources, identified with the **random** statement, with any remaining variance assigned to residuals. Now, imagine that the true differences between the kids in the preceding example is tiny; that is, the kids are effectively clones, and the observed differences between them are due almost entirely to error of measurement. You expect to get a very small positive variance for ChildID, sure, but what about its uncertainty? For that you need a sampling distribution for the variance to calculate the lower and upper confidence limits. The residual variance always has a chi-squared distribution, and there are no problems with its confidence limits: the lower limit is always positive, and the upper limit is exactly what you would expect for a variance with whatever degrees of freedom (see below) are left over for the residual. If you use the chi-squared distribution for the ChildID variance, you get an apparently sensible lower confidence limit very close to zero, but the upper confidence limit is an impossibly large value. So, instead, you have to assume that the variance has a *normal* distribution, which automatically means that the lower limit can be negative, and in our example of a very small ChildID variance, it *will* be negative.

You allow a normal distribution for the variance by telling Proc Mixed to allow negative variance, which you do by specifying **nobound** in the **proc mixed** statement. It will then allow negative variance not only for the lower confidence limit, but also for the variance itself and for the upper confidence limit. The confidence interval now looks like a realistic range for the sampling uncertainty, whereas with the limits decided by the chi-squared distribution (i.e., allowing only positive variance when you omit **nobound**), the confidence interval is unrealistic. What's more, if the true variance of ChildID in the population is tiny, then in nearly half of all studies you can expect to get a negative variance, simply because of sampling variation. But that's OK, the upper confidence limit will almost always be some sensible positive number that

properly estimates how different the kids really are from each other. Yes, except that you have to take the square root of the variance and its limits to get estimates of the SD and its limits, so what do you do about negative values? Change the sign to positive, take the square root, then call it a negative SD. No problem. Really.

One more very important point about allowing negative variance... If you allow it, the estimate of the variance will be unbiased; that is, the estimate on average will be the true value. If you don't allow it, then when SAS tries to estimate a negative value, it's not allowed to, so the estimate is set to zero, so on average the estimate is biased high (you only ever see zeros or positive values, the average of which must exceed the true value). The bias is particularly bad when the true variance and its degrees of freedom are both small.

Look, you will hardly ever get negative variance or a negative lower confidence limit for the random effect representing identity of subjects, so there's no harm done by allowing only positive variance for this random effect. Where it really matters is when the random effect is specified with a dummy variable to represent a small extra amount of variance, such as extra error on the first trial in a reliability study, or extra variability representing individual responses in a time series or controlled trial. Then it's possible for the variance to be truly negative, and you need to allow it—see later.

But what about the residuals in this example of repeated measurement? They would be specified by the random effect ChildID*Trial, which uniquely identifies each observation in the dataset. An interaction of a random effect with any other variable is also a random effect, in this case the residual. You need to understand the interaction that would represent the residual in an analysis, and equally you need to remember that you can't specify it with the `random` statement. The individual residual values now represent the extent to which each child's value in each trial differs from the overall mean for the given child in the given trial. The standard deviation of the residuals is therefore a measure of **within-subject variability** from trial to trial. The fixed effect Trial and the random effect ChildID together represent a simple reliability model, with Trial providing an estimate of the change in the mean between trials (the habituation or learning effect) and with the SD of the residuals providing an estimate of the **standard or typical error of**

measurement. The results of this analysis are practically identical with those of the two-way reliability ANOVA.

But maybe such a simple reliability model does not apply to these data. Maybe the residual on the first trial is greater than that on the second trial; for example, the subjects may not be accustomed or habituated to performing the test. In that case, you can specify a different residual variance for the different levels of Trial:

```
random Int/subject=ChildID;
repeated/group=Trial;
parms 100 1 1;
```

Note the `parms` statement. When I ran the analysis with a small sample size (20) without this statement, the program sometimes failed. More about this shortly.

I'm always amazed that the mixed model can estimate differences between subjects with the random ChildID statement, along with different residuals for the two or more trials. Be warned, though: even if you can get it to work with a small sample size, there will be a lot of uncertainty in the estimates of one or more of the variances. The uncertainty is represented by standard errors and confidence limits, coming up after an explanation of...

Failed Analyses

When the analysis in the previous program failed, I got a ☒ on the program tab. When I found the errors in the LOG window, they all related to trying to process an empty dataset. In other words, the mixed model did not run. I scrolled further back in the LOG window and found this for the `proc mixed` part of the program:

```
WARNING: Stopped because of infinite likelihood.
```

This kind of problem can arise when the random-effects model gets complicated, especially when you allow negative variance and/or have small sample sizes. How come?

The mixed model works by an iterative process, whereby it starts by guessing values for all the parameters in the model, then it evaluates the likelihood of the data, given those values, then it tweaks the values to increase the likelihood (i.e., decrease the unlikelihood), evaluates the likelihood, tweaks the values, and so on, until the increase in likelihood is tiny. Then it stops, shows this in the LOG...


Convergence criteria met,

...and gives you the results in the RESULTS window. But when it says "infinite likelihood",

it can't get started. In other words, the algorithm it uses to get starting values for the parameters didn't work. You have to help it by giving it reasonable starting values for the random-effect and residual variances and covariances (covparms), which you do with the `parms` statement. One way to get starting values is to allow only positive variance (by removing `nobound` from the `proc mixed` statement), which usually solves the problem of infinite likelihood, and may explain why the creators of Proc Mixed didn't make negative variance the default. You then add a `parms` statement with the values for the covparms (approximately—to within a factor of 10), including any zeros, and turn on negative variance again before you re-run the program. Or you can leave negative variance turned on, guess sensible values for the SDs, square them, then include them in the `parms` statement. The order of the variances and covariances following `parms` is the order specified in the random statement (with any covariances between the two variances), followed by the residuals.

Did not converge is another problem that can occur with complex random-effects models and small sample sizes. In this case, the iterative procedure could not reach a sufficiently tiny increase in likelihood. The solution is to relax the convergence criteria by putting this term in the `proc mixed` statement: `convh=1E-6 convf=1E-6`; (the default value is -8, so when you don't use this statement in the `proc mixed` statement, you are effectively specifying `convh=1E-8 convf=1E-8`). If it still doesn't converge, try -5 rather than -6. If it does converge, then try -4, check that the estimates for everything don't change to any important extent, then switch back to -5 and re-run the program to get the estimates. If the values of the estimates *do* change substantially, you are sunk! You'll have to try simplifying the model, or maybe there's a weird outlier in the data that needs to be removed. Viewing **residuals vs predicted values** is a great way to spot outliers, but if you have infinite likelihood, get the program to run with a simpler model first to generate the residuals and predicted.

I run the mixed model by suppressing all the results that Proc Mixed provides by default. I get the results I want by creating datasets of the results using options in Proc Mixed, then processing and `proc printing` the datasets. Consequently, if Proc Mixed doesn't work, but it had worked previously, the RESULTS window will show the results from the previous analysis, and

you may think that Proc Mixed *has* worked, especially if the program tab doesn't show  to indicate errors. I avoid this problem by clearing all the datasets arising from any previous analysis with a data step just before the `proc mixed` step.

Standard Errors

The estimates of variances in a mixed model are sample statistics, so they have sampling uncertainty, which represents how much they are expected to change between studies of the same size and sampling method. The sampling uncertainty is estimated as a **standard error** (SE), which is the expected typical variation in the statistic, if the study were repeated again and again. For the residuals, the SE of the variance comes from a chi-squared distribution, and for variances specified with the random statement, the standard errors come from a normal distribution or a chi-squared distribution, with or without `nobound`, respectively.

Standard errors of a chi-squared distribution are specified by a single parameter, **degrees of freedom** (DF), which is the number of independent bits of information that contribute to calculation of a standard deviation or variance. For this reason, I like to see the degrees of freedom, but SAS doesn't provide it, so I have to calculate it from the SE, which SAS *does* provide. The SE of the variance from the chi-squared distribution is a factor SE of $1/\sqrt{2DF}$; that is, the SE of the variance in the units of the variance is $\text{Variance} \cdot (1/\sqrt{2DF})$. Hence $DF=2(\text{Variance}/SE)^2$. SAS also provides $\text{Variance}/SE$ as the Zvalue, the normal-distribution equivalent of the t value, which is kind-of weird, because the residual variances are not normally distributed. But that's OK, just go ahead and estimate degrees of freedom with $2 \cdot \text{Zvalue}^2$ in a data step. The value you get with simple models is what you expect from first principles. I use the same formula to generate the degrees of freedom for the random effect variances, which is also kind-of weird, because I am assuming they are normally distributed with `nobound`. But again, that's OK, because I am seeing the degrees of freedom of the random-effect variances, if they *were* chi-squared distributed, and the value *does* represent the amount of information in the variance. The mixed model also provides standard errors for the fixed effects, using the residuals and random effects, and the sampling distributions for fixed effects are always t distributions.

The standard errors and sampling distribution

are used to calculate confidence limits, which you should present for most effects and some SDs. There are several approaches to then interpreting the sampling uncertainty represented by the confidence limits or the sampling distribution, as indicated in the Introduction.

Repeated Measurement with Dummy Variables

[04 Reliability 2 trials.sas](#)

[05 Reliability 3 trials.sas](#)

[06 Time series.sas](#)

In the previous model, instead of specifying a different residual variance on each trial, it's possible to specify the same residual on each trial and extra error variance on one or more trials using one or more dummy variables. In the case of a simple reliability study with two trials, where you expect extra error on the first trial, you specify extra error on the first trial. First, define a dummy variable with a suitable name in a `data` step before the `proc mixed` step:

```
data dat2;
set dat1;
XvarTrial1=0; *dummy variable to
  estimate extra error on Trial 1;
if Trial=1 then XvarTrial1=1;
```

The fixed-effects model is the same as before:

```
model VO2max=Trial;
```

The random-effects model is this...

```
random ChildID XvarTrial1*ChildID;
...or this...
```

```
random Int XvarTrial1/subject=ChildID;
...because...
random Whatever/subject=SubjectID
...means the same as...
```

```
random Whatever*SubjectID
...but not when type=un; more on this soon.
```

The term `XvarTrial1*ChildID` represents something only on the first trial, because it's zero on the second trial. And on the first trial, it represents a random effect, just like `ChildID`, so SAS estimates a value for every child, additional to the value it is already estimating for `random ChildID`: there's a hat labeled `ChildID`, and another hat labeled `XvarTrial1*ChildID`; every child gets a number out of the `ChildID` hat, and every child gets a number out of the `XvarTrial1*ChildID` hat when it's the child's first trial. In other words, every child gets an additional random error on the first trial. There is still the residual variance, but we don't state it with a `repeated`, because we want the same residual variance on Trial 1 and Trial 2. The analysis is equivalent to estimating a separate residual on

Trial 1 and Trial 2, with the residual variance for Trial 1 given by adding together the variance for `XvarTrial1*ChildID` and the residual variance. So as before, you need a big sample size for this analysis to work well, when there are only two trials. But with three trials, if you assume the same residual variance on Trials 2 and 3, the analysis works OK with 10 or so subjects. (That doesn't mean that only 10 or so subjects are enough!)

Instead of representing a reliability study, the two trials could represent a **time series**, in which measurements are taken before and during or after an **intervention**. In this case, you would make a dummy variable for the second trial, representing extra variability arising from **individual responses** to the intervention. The random-effect solution for the dummy are the values of the individual responses (around the mean, provided by the fixed effect `Trial`), and the SD for the dummy summarizes the individual responses as typical between-subject differences around the mean effect of the intervention.

With three or more trials, the data could represent a time series in which you take a baseline measurement (Trial 1), then do an intervention and see what happens at later time points (Trial 2, 3, and so on). Let's limit it to three trials. You can expect individual responses in Trials 2 and 3, but the responses may differ between the trials and yet be correlated: subjects who start to respond well in Trial 2 may respond even more in Trial 3; subjects who don't respond well in Trial 2 may not respond well in Trial 3; and so on for different subjects. Dummy variables now allow you to model such correlated individual responses. What you need is a dummy variable that will estimate the same individual response on both trials for a given subject, a dummy variable that will estimate the extra individual response in Trial 2, and another dummy variable that will estimate the extra individual response in Trial 3. Here is the data step to create the dummy variables:

```
data dat2;
set dat1;
XvarTrial23=0; *dummy variable to
  estimate extra variance shared by
  Trial 2 and 3;
if Trial=2 or Trial=3 then
  XvarTrial23=1;
XvarTrial2=0; *dummy variable to
  estimate extra variance on Trial 2;
if Trial=2 then XvarTrial2=1;
XvarTrial3=0; *dummy variable to
```

```
estimate extra variance on Trial 3;
if Trial=3 then XvarTrial3=1;
```

And here is the random statement to estimate the extra variances:

```
random Int XvarTrial2 XvarTrial23
XvarTrial3/subject=ChildID;
```

To get the actual individual responses on Trial 2, you have to add the variance for XvarTrial2 to the variance for XvarTrial23 (and then take the square root); to get the actual individual responses on Trial 3, you have to add the variance for XvarTrial3 to the variance for XvarTrial23 (and then take the square root). Calculating the SEs for combined variances is complicated.

Once again, you need a big sample size to get adequate precision, and the analysis will sometimes fail with a small sample size (e.g., 10). It's much better if you include a control group in which you assume there are no individual responses. See later.

There is a more elegant way to estimate the individual responses, using a `random` statement with `type=un`. But first I will introduce the use of `type=un` with the `repeated` statement.

Repeated Measurement with Type=un

[04 Reliability 2 trials.sas](#)

[05 Reliability 3 trials.sas](#)

[06 Time series.sas](#)

Consider again the reliability model with only two trials. The random-effects model, consisting of a between-subject variance (specified with `random ChildID`) and different residual variances in the two trials (specified with `repeated/group=Trial`) can be specified another way, using just the `repeated` statement to represent the repeated measurements:

```
repeated Trial/subject=ChildID
type=un;
```

Let's unpack this statement bit by bit.

First, `repeated Trial` is easy enough: the levels of Trial represent repeated measurements. But repeated measurements on whom? The kids of course, so we have to state `subject=ChildID`. But if that's all we had in the statement, it's the equivalent of `repeated Trial*ChildID`, which specifies a residual that changes for every observation (every observation is uniquely identified by a level of Trial and a level of ChildID). It's as if the second trial consists of different kids, so the analysis is just like having Sex instead of Trial: we've lost the sense of repeated measurement, and there is only one residual variance.

We therefore need something extra to tell SAS

that we have indeed got repeated measurement, and that we want to account for differences between kids (which we got previously with `random ChildID`) as well as a different residual error of measurement on each trial (which we got previously with `repeated/group=Trial`). The something extra is `type=un`: the `type=` stands for the type of random-effect structure you want, and the `un` is short for *unstructured*, which is a complete misnomer, because what it specifies is the most general or complicated structure of all, as follows...

`Type=un` instructs SAS to estimate a different value for each subject on Trial 1 and a different value for each subject on Trial 2, but allowing for the possibility that the set of values in Trial 1 are **correlated** with the set of values in Trial 2. They are indeed correlated, because kids with a high, low or moderate value of VO2max overall will tend to have a high, low or moderate value on Trial 1 and Trial 2. The correlation is accounted for with an estimate of **covariance**, which measures the extent to which the two variances have a shared or common variance. The shared variance represents the true differences between the kids, the difference that was estimated previously with `random ChildID`: If you add that shared variance to the residual variance on Trial 1 or Trial 2 (estimated previously with `repeated/group=Trial`), you get the observed variance on Trial 1 or Trial 2. The variances that SAS produces for random effects with the `repeated` or `random` statements are called **covariance parameters** (covparms, for short), because in general, random effects can represent not just variances but also covariances, when the random effects are correlated.

The `repeated Trial` statement tells Proc Mixed how to estimate residuals. In the absence of any specified random effects, the residuals are just the difference between the observed values and the mean values predicted by the fixed effects. So, the residuals are bigger with this approach than with a `random` statement that estimates differences between subjects. But the correlation between the residuals specified with `subject=ChildID type=un` properly accounts for the within-subject variability in the data and gives the right answers for the fixed effects. (For more on the difference between `repeated` and `random`, visit [this link](#).)

Unfortunately, using `repeated` in this manner does not produce estimates of the usual typical-error variances or individual-response variances:

all you see is the variance in the two trials, and the covariance, so you have to subtract the covariance from the variances to see how much error there is on each trial. The variances and covariance are not labeled with the levels of Trial, but are instead labeled UN(1,1), UN(2,1) and UN(2,2), representing respectively the variance for Trial 1, the covariance, and the variance for Trial 2. Furthermore, SAS does not produce a random-effect solution for the covariance, so you can't see which kids are fitter than average and which kids are less fit.

So why bother with this repeated approach? For researchers interested only in fixed effects with repeated measurement, it's an easy way to account fully for all the potentially correlated sources of variability, especially in a controlled trial with three or more levels for Trial or clustered repeated measurements. And it always works, and it's fast with big sample sizes. But researchers should be interested in sources of error in reliability studies, in individual responses in interventions, and in the error of measurement in the interventions to check on the reliability of the dependent variable. For these you need dummy variables, as we've already seen.

Now let's return to the time series with three trials. Here's an elegant way to specify and estimate correlated individual responses on Trials 2 and 3...

```
random Int/subject=ChildID;
random XvarTrial2 XvarTrial3
/subject=ChildID type=un;
```

The first random statement estimates the differences between the kids. The second random statement draws a number out of a hat labeled XvarTrial2*ChildID for every child, representing the individual responses to Trial 2, and there's another hat for Trial 3, while the type=un estimates their covariance (the individual responses shared between the trials). So, with this syntax, the covariance has already been added to the *extra* individual responses on the two trials. Note that if you omitted type=un, you would get estimates of individual responses on the two trials, but SAS would assume they were independent, which is not correct, so the estimates of everything would change to some extent and would be untrustworthy. Note also that when you have type=un, you must use the subject= syntax.

This is a good opportunity to mention other type= covariance structures that are used with subject=. When you don't want the random effects to be correlated, the type of structure is

called **variance components**, and it is requested with type=vc. In fact, type=vc is the default and doesn't need to be stated. So, to be clear...

```
random XvarTrial2
XvarTrial3/subject=ChildID type=vc;
...is the same as...
random XvarTrial2
XvarTrial3/subject=ChildID;
```

I guess the reasoning behind this obscure designation is that there is variability in the data, and you want to partition the variability into different uncorrelated components.

There are many other types of covariance structure, but the only one I have used (very rarely) is called **spatial covariance**. It's easiest to understand with the repeated statement. Imagine you had many repeated measurements of Trial on each child. With...

```
repeated Trial/subject=ChildID
type=un;
```

...you are specifying the most general covariance matrix, with different variances on every trial and different covariances between every pair of trials. But with longer time between pairs of trials, it's possible that the covariances will be smaller, or to put it another way, the correlations between trials close together may be high, and the further apart the trials, the lower the correlation. This scenario would arise if the kids changed gradually with time, and they differed in their rates of change. This scenario also captures the notion that reliability is generally lower, the longer the time between trials. Well, fair enough, but I have not found spatial covariance models to be useful when dealing with athletes over extended periods. Instead, I allow for one residual error variance within a season or over a short period of time, and an additional source of within-athlete variability between seasons or between blocks of measurements separated by longer periods of time, to represent the possibility that some athletes get relatively better or worse between seasons or blocks. Of course, if relevant, a fixed effect for a *gradual* change over time can be included (estimated as a slope), and a random effect can be included to specify individual differences in the changes (an SD for the slope). That's coming up shortly, in the guise of random slopes and intercepts. When you account for individual differences in gradual changes in this manner, there is less, if any, need for a spatial covariance structure.

Here is another way to specify correlated individual responses on Trials 2 and 3. I'll leave you

to figure it out:

```
random Int/subject=ChildID;
random XvarTrial23*Trial
  /subject=ChildID type=un;
```

It produces a slightly different covariance table, with zeros for the variances and covariances involving Trial 1. Surprisingly, the estimates differ slightly from those of the previous model. They are both right, but this model produces less uncertainty for the residual variance with small sample sizes (and sometimes the previous model won't estimate a standard error for the residual). I don't know why.

Groups of Random Effects

[07 Reliability 2 trials+sex.sas](#)

Let's now add Sex to a dataset consisting of repeated measurement; for example, values of VO2max of children (ChildID) tested twice (Trial). The fixed-effect model that allows for means of boys and girls to differ on each trial is this...

```
model VO2max=Sex Trial Sex*Trial;
...or this...
```

```
model VO2max=Sex*Trial;
```

Previously the following random-effects model estimated an SD representing differences between the children and a residual SD representing variability within each child between trials:

```
random ChildID;
```

But boys and girls are different creatures. Obviously, the mean for boys could differ substantially from the mean for girls, but the SD for boys could be different, too: there might be more scatter in the values for girls than for boys, for example. SAS allows you to specify this possibility:

```
random ChildID/group=Sex;
```

The analysis will now produce a variance for boys and a variance for girls; take the square root and you will get the two SDs. Take advantage of the assumed normal distribution of the variances to compare them inferentially with the spreadsheet [Combine/compare effects](#) (Hopkins, 2006) via the tab for 2 groups or statistics (for which you will need the square of the smallest important SD). Alternatively, run the analysis to see which SD is greater, then create a dummy variable for that group (e.g., XvarGirls) and estimate its variance by adding it to the random statement:

```
random Int XvarGirls/subject=ChildID;
```

With this approach (not shown in the simula-

tion), use the tab for 1 or more groups or statistics to evaluate the magnitude of the variance inferentially.

Don't forget to allow for different residuals for the boys and girls with...

```
repeated/group=Sex;
```

...which is based on the assumption of a residual error for the boys that is the same in Trials 1 and 2, and a different residual error for the girls that is the same in Trials 1 and 2. If you want to allow for a different residual error in each trial that is the same for each sex, use this...

```
repeated/group=Trial;
```

...or this, if you want to allow for a different residual in each trial and sex...

```
repeated/group=Sex*Trial;
```

...but you need a big sample size for the analysis to run properly. The following statement, without `random ChildID` in the model, will run much more quickly and achieve the same results for the fixed effects, but it will leave you in the dark about errors of measurement (previously given by the residuals) and the individual differences between kids (previously given by the random-effect solution for ChildID):

```
repeated Trial/subject=ChildID type=un
  group=Sex;
```

Controlled Trial with Two Trials

[08 Controlled trial 2 trials.sas](#)

The previous model, with boys and girls tested twice, becomes a controlled trial, if we change Sex (with levels *boy*, *girl*) to Treat (with levels *control*, *exptal*). The fixed-effects model is:

```
model VO2max=Treat*Trial;
```

(There is no point in specifying the main effects for Treat and Trial in a controlled trial, because the overall means for Treat and the overall means for Trial have no useful interpretations.)

For the random-effects model, we want to specify individual responses on the second trial in the experimental group. To do that, we need a previous data step:

```
data dat2;
```

```
set dat1;
```

```
XvarExpTrial2=0; *individual responses
```

```
  in exptal group;
```

```
if Treat="exptal" and Trial=2 then
```

```
  XvarExpTrial2=1;
```

Let's jump straight to the most parsimonious and informative random-effects model:

```
random Int XvarExpTrial2
```

```
  /subject=ChildID type=un group=Treat;
```

```
parms 100 0 0 100 0 5 10;
```

The Int specifies the ChildID clusters and thereby estimates each kid's true mean value.

The `XvarExpTrial2` specifies a randomly chosen value for each kid in the experimental group on the second trial, thereby specifying the individual responses.

The `type=un` allows for the individual responses (`XvarExpTrial2`) to be correlated with the subject's true value (`Int`). For example, in a study of the effect of a new kind of training on performance, athletes with higher true values of performance might have less headroom to improve, or their existing training might include elements of the new training, so they may experience a smaller effect; in this example, the covariance specified by `type=un` will be negative. The covariance is converted to a slope representing the individual response per unit of true value by dividing it by the true variance (`Int`), and its confidence limits have to be derived by parametric bootstrapping (as shown in the program: the standard errors of the numerator and denominator are used to generate the bootstrapped samples).

The `group=Treat` allows for different random effects in the two groups. The model works without this term, but including it makes the random-effect solution easier to interpret.

Note the `parms` statement: it's needed here to ensure the analysis works with small sample sizes. There's no `repeated` statement, so this model estimates a single residual variance representing the same error of measurement in both groups. We can make that assumption, because the kids were randomized or assigned to the two groups to make the groups as equal as possible.

It's possible to estimate additional error on the first trial (e.g., arising from habituation), but you need a large sample size for the model to work properly and give reasonable uncertainties in the random effect. If there is such error in the data, the above model incorporates it into the residual and still produces good estimates of the individual responses.

This model assumes zero individual responses to the control treatment. If there are such responses, this model appears to do a good job at estimating the individual responses in the experimental group additional to whatever is in the control group. You could include an extra random effect to estimate individual responses in the control group, but the covariances get complicated, and again, you would need a large sample size.

Finally, if all you care about is the fixed effects, then the following random-effects model

will work with small sample sizes, it runs very quickly, and it allows for individual responses in both groups and different errors of measurement in both groups, but it won't estimate them:

```
repeated Trial/subject=ChildID type=un
group=Treat;
```

The estimates and uncertainty for the fixed-effects with this model are almost exactly the same as with the previous model.

Controlled Trial with Three Trials

[09 Controlled trial 3 trials.sas](#)

With three trials, you could have two baseline trials and one intervention trial, in which case you might allow for extra error in the first baseline trial (due to familiarization), and for extra error in the intervention trial in the experimental group (due to individual responses). But let's instead have one baseline and two intervention trials, with extra error in the first trial for familiarization, and extra error in the second and third trials for the experimental group, but allowing for correlated individual responses on those trials. The correlation would arise through each subject in the experimental group having an individual response that is shared (i.e., is the same) for Trials 2 and 3, with something extra for Trial 2 and something extra for Trial 3. These three errors are independent of each other, and that's how the data are simulated. I have also simulated an additional shared component of individual responses correlated with the subjects' true value, as in the controlled trial with two trials.

The total shared component of individual responses is estimated as the covariance between the individual responses on Trials 2 and 3. Here is the data step to create dummy variables to estimate the various components of variability, including extra error on Trial 1 arising from familiarization:

```
data dat2;
set dat1;
XvarTrial1=0;
if Trial=1 then XvarTrial1=1;
XvarExpTrial2=0; *indiv responses in
  exptal group on Trial 2;
if Treat="exptal" and Trial=2 then
  XvarExpTrial2=1;
XvarExpTrial3=0; *indiv responses in
  exptal group on Trial 3;
if Treat="exptal" and Trial=3 then
  XvarExpTrial3=1;
```

The fixed-effects model is of course this:

```
model V02max=Treat*Trial;
```

Here is the random-effects model that accounts for extra error on the first trial in both

```

groups, and individual responses in the second
and third trial in the experimental group:
random XvarTrial1/subject=ChildID;
random Int XvarExpTrial2 XvarExpTrial3
/subject=ChildID type=un group=Treat;
parms 5 100 0 0 0 0 0 100 5 5
5 5 5 5;

```

The first `random` statement estimates the additional error on Trial 1, which it is now possible to estimate with a modest sample size. The second `random` statement estimates the subjects' true values (Int) and the individual responses on Trials 2 and 3 (XvarExpTrial2, XvarExpTrial3). The `type=un` estimates three covariances representing the modifying effect of Int on Trial 2, the modifying effect of Int on Trial 3, and the individual responses shared between Trials 2 and 3. The modifying effects of Int are subsequently expressed as slopes, with confidence limits generated by bootstrapping, as described in the previous section. The `group=Treat` produces one lot of covparms for the control group and another lot for the experimental group. All but one of the covparms in the control group are zero, because of the way XvarExpTrial2 and XvarExpTrial3 are defined, hence the values shown for the `parms` statement. You will have to run the simulation and view the program and results to see how the unstructured labels are reassigned meaningful names.

I'm really sorry that the random-effects model has become so complicated. In practice, you will probably have a sample size that is too small to get adequate precision for the SDs representing individual responses anyway, so focus on only the fixed effects by running this `repeated` statement:

```

repeated Trial/subject=ChildID type=un
group=Treat;

```

You can still deal with individual responses by estimating the extent to which the treatment effect is modified by a subject characteristic. If the characteristic is X, say, then the full fixed-effect model is:

```

model V02max=Treat*Trial
Treat*Trial*X;

```

Use `estimate` statements to determine the modifying effect of one unit of X and/or 2 SD of X in all the Treat*Trial groups and their differences, the most important being `exptal` minus control for post minus pre. I have not provided a simulation for this analysis.

Unfortunately, you can't include the baseline value of the dependent variable as a modifier, be-

cause it's already in the model as one of the values on the left-hand side of the equals sign. Of course, you are already estimating the modifying effect of true score via the covparms, but you can instead use the baseline as a modifier in a simpler model: analysis of...

Controlled Trials via Change Scores

[10 Controlled trial via change scores.sas](#)

In a controlled trial, it's important not only to quantify individual responses to the treatment but also to determine the extent to which any individual responses are explained by one or more subject characteristics. The easiest way to see if a subject characteristic modifies a treatment is to plot the changes in the dependent variable against the subject characteristic. An analysis of change scores reproduces what you see in the scatter plot.

One of the most important characteristics is the subject's true value of the dependent variable before the intervention: as explained in the previous sections, subjects with a high value might experience less effect from an intervention. When analyzing change scores, the true value can't be estimated from the random-effects model, so the observed baseline (pre-intervention) value is used as a fixed-effect modifier.

With baseline as the modifier, you will often see a negative slope in the plot of change scores vs baseline. This negative slope arises from [regression to the mean](#), and it becomes noticeable when the measure is "noisy" (error of measurement is comparable to the between-subject SD): a baseline score that is lower (or higher) than average is at least partly due to random error of measurement being negative (or positive), so scores will on average increase (decrease) on retest. In a controlled trial, both groups will show regression to the mean, but any real modifying effect of baseline will be apparent as different slopes in the two groups, so the difference in the slopes (`exptal` minus control) is what you want.

Here's the code to produce change scores for two post tests (Trial 2 and 3) and to make the baseline scores available for the post tests. The data need to have been sorted by ChildID Treat Trial (in the simulation they are generated in that sort order):

```

data dat2;
set dat1;
retain Baseline;
if Trial=1 then Baseline=V02max;
DeltaV02max=V02max-Baseline;

```

The value of Baseline is the value of observed

performance in Trial 1, and `retain` keeps the value available for Trials 2 and 3. The observation for Trial 1 (change score = 0) is subsequently deleted. In the simulation program, the change scores for Trials 2 and 3 are analyzed separately, "by Trial". The fixed-effect model is simply...

```
model DeltaVO2max=Treat
    Treat*Baseline;
```

...where DeltaVO2max is the post minus the pre value, Treat has values *control* and *exptal* and estimates two means, Baseline is the pre-test value of VO2max, and Treat*Baseline estimates two slopes. `Estimate` statements provide the effects of Treat in the two groups, the slopes due to Baseline in the two groups, and the differences between the groups. The `estimate` statements for Treat are simplified by having a preceding `proc standard`, in which the grand mean of Baseline is rescaled to zero; the SD can also be rescaled to 0.5, to estimate the modifying effect of 2 SD of Baseline with unity coefficients in the estimate statements.

The random-effects model is this simple...

```
random XvarExp/subject=ChildID;
```

...where XvarExp is a dummy variable with values 0 in the control and 1 in the experimental group. Its SD summarizes the individual responses. There is no `random Int`, because there is only one observation per subject.

The magnitude of the residual SD depends on the magnitude of the standard or typical error of measurement relative to the SD representing true differences between subjects. If the error is much less than the SD, the residual will approach $\sqrt{2}$ times the error of measurement, because it represents residual variability of change scores. If the error is much more than the SD, the residual will approach the error of measurement, because the baseline fixed effect will account for regression to the mean.

This approach to individual responses using change scores with a baseline effect modifier has the advantage of being simpler and more visual than the approach using original scores. With noisy measures (typical error comparable to the true SD), the estimates of fixed effects also have somewhat better precision. Unfortunately, the noise attenuates the modifying effect of baseline; that is, the slope on average is smaller in magnitude than the true slope (although it correctly estimates the effect of the observed value in Trial 1). Furthermore, the estimate of the SD representing individual responses can't be interpreted

as such, when the measure is noisy and there is a substantial contribution of baseline to the individual responses, because that contribution is not completely removed by the baseline covariate.

In stats text books or sites, you may see one other approach to account for a modifying effect of baseline, using original scores rather than change scores: simply replace DeltaVO2max with VO2max in the above fixed-effects model. All the results are the same, except that the slopes representing the modifying effect of Baseline in the two groups are close to 1.0, regression to the mean is evident as slopes less than 1.0, and you can't easily see the modifying effect of baseline on the change scores. The modifying effects of any other subject characteristic included in the model are also hard to visualize. I don't recommend this approach.

So what's it to be: analysis of raw scores with unstructured random-effects models, or analysis of change scores with a baseline covariate? The former gives unbiased estimates of everything, but it needs a large sample size for adequate precision of the individual-response SDs. The latter gives better precision but biased estimates of the modifying effect of baseline and individual responses. If your sample size is small, go with change scores, especially for the modifying effects of other subject characteristics.

One final note on controlled trials...I used to think that the change scores in Trials 2 and 3 could be analyzed with a single mixed model that accounted for shared individual responses in the two trials. In studying the simulations, I realized that the shared individual responses are contaminated by error of measurement on Trial 1 being carried through into the change scores on Trials 2 and 3. The contamination can be removed with a subject-identity random effect, but that also removes the individual responses shared by Trials 2 and 3. The analysis still produces unbiased estimates of mean changes and unbiased estimates of modifying effects of other subject characteristics, but on balance I decided against providing the code for such analyses. Just analyze change scores "by Trial", as shown in the simulation.

Random Effects with Multiple Clusters

[11a Reliability 3x2 trials.sas](#)

[11b Time series 3x2 trials.sas](#)

[11c Time series 3 trials+repeats.sas](#)

I stated earlier that you need to identify all the clusters in your dataset, so let's consider the clusters when measurements are repeated on subjects

on more than one occasion. For example, imagine each child is measured three times on one occasion in the first week of a study, then three times on a second occasion a week later. Now you have another nominal variable Occasion, with two levels (*Week1*, *Week2*). Occasion is a nominal fixed effect. If the kids are of only one sex, the full fixed effect model is specified in SAS as:

```
model VO2max=Occasion Trial
  Occasion*Trial;
```

Occasion*Trial allows for and estimates the possibility that all six means (two occasions times three trials) are different from each other. Occasion estimates the extent to which the mean VO2max changes between the two testing occasions (on average, the kids might improve between occasions, for example). Trial estimates the extent to which the mean for Trial changes between the three trials on both testing occasions.

What about the clusters involving ChildID? There are six measurements on each child, or to be more precise, up to six measurements, because mixed modeling allows missing values. There are three different types of clusters of repeated measurements, represented by ChildID, ChildID*Occasion, and ChildID*Trial. (The interaction ChildID*Occasion*Trial uniquely identifies each observation; hence it does not represent clusters, and it is in fact the residual.) The ChildID clusters represent the relative mean of each child, where the mean is taken over all repeated measurements: some children had high VO2max values across all six measurements, some had low values. The Child*Occasion clusters represent the relative mean of each child, where the mean is taken over the three trials on each occasion: some children did well on the first occasion but less well on the second. The Child*Trial clusters represent the relative mean of each child, where the mean is taken over the two occasions for each trial: the kids differed in the extent to which they changed between the three trials (for example, the pattern of fatigue across the three trials differed between kids). In principle, we should include all three clusters in the analysis as their corresponding random effects:

```
random ChildID ChildID*Occasion
  ChildID*Trial;
```

Or we can use this notation:

```
random Int Occasion Trial
  /subject=ChildID;
```

I am happy that Occasion in the above model estimates extra error over the longer time period (and that each child has a different value on both occasions). But Trial represents only one "hat" for each trial on both occasions: for a given child, you take a number out of the hat for Trial 1, and it's the same number for both occasions, and the SD is the same for each trial. That structure may be too complex or too simple.

So, the decision about how to include clusters of repeated measurements as random effects depends on what you think is going on. If individual differences in the pattern of means between trials is not an issue, a simple reliability model would be this:

```
random Int Occasion/subject=SubjectID;
Int gives the real differences between subjects,
Occasion gives the within-subject between-occasions
variability, and the residual gives the typical error
between trials (the same for each trial on both
occasions). At the other extreme, when you are
interested only in the fixed effects, the following
model allows for the most general random-effects
structure...
```

```
repeated Occasion*Trial
  /subject=ChildID type=un;
```

...but as before, Proc Mixed provides subject variances for each level of Occasion*Trial and pairwise covariances between the levels, so you can't disentangle the sources of variability and error.

In the simulation 11a Reliability 3x2 trials.sas, there is no intervention, so the mean would change only a bit between trials; I have made a small positive mean change after Trial 1, but if it really was a VO2max test repeated on the same day, the kids would probably show fatigue in the second trial and even more in the third trial. I have assumed that there is extra familiarization error on the first trial regardless of occasion, so for each child, a number comes out of a hat for the first trial on the first occasion and a number comes out of that hat again on the second occasion. There is also a hat for Occasion, as described, and obviously a hat for ChildID. The residual hat represents the typical error in all six trials, after the extra error on Trial 1 and the within-child between-occasions error have been estimated. Here is the random-effects model:

```
random Int XvarTrial1*Occasion
  Occasion/subject=SubjectID;
```

Int estimates the true between-child SD, XvarTrial1*Occasion estimates the extra error on Trial 1, and Occasion estimates the within-child

variability between occasions. If instead of `XvarTrial1*Occasion`, you made it simply `XvarTrial1`, you would be estimating individual differences in familiarization that are consistent between occasions: a kid who does better on Trial 1 on the first occasion does better on Trial 1 on the second occasion. I think it's safer to go with `XvarTrial1*Occasion`.

In the simulation 11b Time series 3x2 trials.sas, I have imagined that there is an acutely ergogenic treatment after Trial 1, such as caffeine, producing an increase in the mean on Trial 2 and a further increase on Trial 3. The fixed-effects model is the same as above. I have assumed the same individual responses to the treatment on both occasions, so for each child, a number comes out of each of three hats: Trial 2, Trial 3, and Trial 2&3 (the individual response shared between Trials 2 and 3), and those numbers are applied to Occasion 1 and Occasion 2. I have also simulated a modifying effect of the kids' true scores on part of the shared individual responses, in the same manner as for the controlled trials. Here's the random-effects model that captures that error structure:

```
random XvarTrial1*Occasion Occasion
  /subject=SubjectID;
random Int XvarTrial2 XvarTrial3
  /subject=SubjectID type=un;
parms 5 5 100 5 5 5 5 5 5;
```

The first random statement is the same as for the reliability analysis, allowing for extra error on Trial 1 and within-child variability between occasions. The second random statement gives the modifying effect of true score on the individual responses in Trials 2 and 3, and correlated individual responses in Trials 2 and 3.

The other simulation, 11c Time series 3 trials+repeats.sas, illustrates modeling when the test for each trial is repeated immediately, rather than repeating the trials a week later. If you can convince your subjects to do the extra tests, it's a great way to improve precision. Obviously, you couldn't do six VO2max tests on the same day, so I have imagined it's a test of sprint speed, with an acute intervention after the two sprints of the first trial. I have introduced a variable `Rep` (short for *repetition*) to estimate the means for each repetition within each trial (but I haven't bothered to simulate any changes in the mean between repetitions). Hence the fixed-effects model is:

```
model Speed=Trial Rep Trial*Rep;
```

I have added some extra residual random error

on both repetitions in the first trial, to simulate familiarization, and I have allowed for this extra error and the possibility that the residual error is different for the repetitions in the second and third trial with `repeated/group=Trial`. Only one random statement is needed to estimate each kid's true score, the modifying effect of true score on the individual responses in Trials 2 and 3, and the correlated individual responses in Trials 2 and 3. Here is the random-effects model:

```
random Int XvarTrial2 XvarTrial3
  /subject=ChildID type=un;
repeated/group=Trial;
parms 100 5 5 5 5 5 10 10 10;
```

Within-Subject Linear Modeling

[12 Within subject modeling.sas](#)

[13 Within & between subject modeling.sas](#)

The first model I presented in this article was a simple linear regression, in which a numeric dependent variable (e.g., VO2max) is predicted by a numeric predictor variable (e.g., Age) with an equation representing a straight line:

```
model VO2max=Age;
```

There are no random effects, because each observation comes from a different kid.

But suppose each kid is measured every year over a period of years. Or, if the subjects are athletes, imagine their performance in tests or in competitions has been monitored for months or years. You could fit a different straight line to each kid (or athlete), in SAS simply by sorting the data `by ChildID` and adding `by ChildID` to the `proc mixed` step. Then you would have to do an additional complicated analysis to get the average slope and its uncertainty (complicated, because it should be a meta-analytic model, to account for the uncertainty in each kid's slope). Things would get even more complicated if you had `Sex` in the model, and you wanted to compare the average slopes of the boys and the girls, or to compare the predicted means of boys and girls at different ages.

So, instead of doing additional mind-numbing analyses, why not use a mixed model to put all the data into one analysis and allow for every kid to have a different straight line? The fixed-effect model is the same, and it gives the mean slope and intercept for all the kids. Here's the random-effects model:

```
random Int Age/subject=ChildID
  type=un;
```

It's called a **random intercepts and slopes model**, because the `Int` represents a different relative intercept and the `Age` represents a different

relative slope, both relative to the mean intercept and slope specified by the fixed-effects model. Think about the hats metaphor here: every kid has one number from the Int hat (the ChildID hat) and one number from the Age hat (the Age*ChildID hat). Age is a numeric effect, so it has only one level within each child, hence only one number for each child. (If Age had been nominal, e.g., AgeGroup with three levels, then there would be up to three levels within each child.) So, we have a set of values of intercepts and a set of values of slopes, *and they could be correlated*. In fact, they are *definitely negatively correlated*, if Age has its usual value in years, starting at zero. To understand why, imagine a kid's straight line, with a positive slope. The intercept represents the predicted value of VO2max at an age of zero. I guess you would predict a VO2max of zero, but that's not what the data will predict. For example, imagine that all the kids have the same mean age of, say, 12 y, and a range of values from 8 to 16 y. Now, kids who stay sedentary over that age range might change their VO2max very little, so the slope will be only slightly positive (it could even be negative, if the kid's body mass increases through deposition of fat). If you extrapolate back to age zero, the VO2max will be positive and only a little less than the value at age 12 (or even more than at age 12!). Other kids, who gradually get more active, will increase their VO2max, and rising levels of hormones that differ between kids will also make a difference. Their slopes will be more positive, so when you extrapolate back to age zero, the intercept will be less positive, or even negative. In other words, kids who differ in their slopes will differ inversely in their intercepts: the more positive the slope, the less positive or even negative the intercept, i.e., there is a negative correlation or a negative covariance between the random effect for intercept and Age. Hence the need for `type=un`.

It's important to realize that the covariance in this example is an **artifact** arising from the fact that the intercept represents a value extrapolated well below the values of the predictor. It's boring, because it has no useful interpretation, but we absolutely have to include it, or the values of the fixed and random effects and their uncertainties will be untrustworthy. The fixed- and random-effect intercepts are also boring. Who cares about the mean predicted VO2max at age zero

(the fixed-effect intercept)? Who cares about differences between kids at age zero (the random-effect intercepts)?

There is a way to make the intercepts and the covariance interesting. Imagine again that all the kids have the same mean age of around 12 y. Now let's draw the Y axis through that mean age; in other words, let's **rescale** Age so that its mean is zero. You rescale simply by subtracting the mean age from each kid's age, or you can subtract a reference age close to the mean. Now the intercepts represent the mean and individual differences in VO2max at the mean or reference age, and there is little or no artifactual covariance between the intercepts and the slopes. Hence any covariance you do see will represent something potentially useful. For example, in a monitoring study of performance of athletes with a lot of repeated measurement over the monitoring period, you may see that athletes with overall higher performance (a higher positive value for the intercept) have a smaller positive slope, because they have less headroom to improve. In the first program, I have used a covariance factor to add a component of individual differences in the slope to reflect this scenario for the linear trends in the kids' VO2max vs age: kids with overall higher VO2max improve less. You can change the sign of the covariance factor to make kids with higher VO2max improve even more.

It's also important to realize that with repeated measures and a random effect for subject identity, all the effects are within-subject effects. There might be positive slopes within each subject and therefore an overall mean within-subject positive slope, yet if you plot the means of the dependent and predictor variable for each subject, you won't see exactly the same positive slope between the means; that is, the *within-subject* relationship will be different from the *between-subject* relationship. They could even have opposite signs. The second program simulates this scenario, for a dependent variable and data more like what you might encounter when monitoring performance of athletes. Mostly the between-subject relationship doesn't matter, because mostly you are interested in within-subject effects, and with most data, the between-subject slope doesn't affect the within-subject slope substantially. Of course, if the subject means for the predictor are all the same (the repeated measurements of the subjects all have the same age range, for example, as in the first program), you

can't have a between-subject relationship anyway. If the means *aren't* all the same, then to be certain that the between-subject slope doesn't affect the within-subject slope, you have to rescale each subject's mean to zero. The second program includes this rescaling, along with a covariance factor to add a component of individual differences in the slope correlated with the subjects' true ability.

If you want an estimate of the between-subject relationship (the relationship between the subject means of the predictor and the subject means of the dependent), you can simply derive the means and do a separate simple linear regression with them, as shown in the second program. The observed between-subject slope is attenuated to some extent, which happens when there is "noise" in values of the predictor relative to the true between-subject SD of the predictor. When the noise is random and unrelated to the predictor, there is a simple formula to adjust for it. Here the noise is the standard error of the mean of each subject's values of the predictor; this noise is related to the dependent by the within-subject relationship, so there is no simple adjustment. The noise is also responsible for making the observed standard error of the estimate greater than the simulated value, again with no simple adjustment to remove the bias.

What about the residuals in models with random intercepts and slopes? They represent the within-subject variability between repeated measurements, like the typical error in a reliability study, but here they are the within-subject standard error of the estimate. To allow for real differences in the residuals between subjects (i.e., real differences in the standard error of the estimate of each subject's straight line), you would include this:

```
repeated/group=ChildID;
```

If there are boys and girls in the data, you could make it...

```
repeated/group=ChildID*Sex;
```

...or if you have only a few repeated measurements for each child, make it simply...

```
repeated/group=Sex;
```

The above examples of within-subject modeling assume a simple linear relationship within each subject. You should check for non-linearity by plotting residual vs predictor (not predicted) values, where non-linearity will be revealed as non-uniformity in the scatter. In that case, you might have to play with a polynomial relationship. For example, here is a quadratic:

```
model V02max=Age Age*Age;
random Int Age Age*Age/subject=ChildID
type=un;
```

Quadratic effects are a bit hard to interpret, but a huge advantage is that they allow prediction of a maximum or a minimum overall and for each individual (e.g., some kids might top out post puberty), assuming that a quadratic fits the data reasonably well. A cubic might give a better fit for curves that trend towards a plateau, and it will still give maxima or minima, if there are any, but you have to solve a quadratic in the coefficients to derive them. The mean and individual lines (simple linear) or curves (polynomial) can and should be visualized by plotting the predicted mean and individual values available as data sets via options in the model statement. I haven't done that with the simulations (yet).

Instead of polynomials, you could parse Age into several age groups and use these models:

```
model V02max=AgeGroup;
random Int AgeGroup/subject=ChildID
type=un;
```

The fixed and random effects for AgeGroup will be easier to interpret than those for a numeric quadratic. The fixed effects obviously provide the mean changes between the age groups. The random effects aren't too difficult to understand, either: UN(1,1) (corresponding to Int) is the differences between the kid means, while UN(2,2), UN(3,3)... are the individual differences from the mean for the first, second... age group. Forget about interpreting the covariances. Again, process predicted values to show the mean and individual trends, here with line segments connecting the predicted means and each kid's predicted values in the age groups. Again, I haven't done that in the simulations.

Generalized Linear Mixed Models

In all the foregoing analyses, the dependent variable is continuous, the distribution of values (or more properly, the distribution of the residuals) is assumed to be approximately normal, the fixed and random effects predict group and individual means, the random effects represent SDs, and the linear model was realized with Proc Mixed. But some dependent variables are decidedly non-normal: a count of events (such as a count of injuries in each individual), a count of something expressed as a proportion of the total (such as proportion of a team experiencing an injury), and a proportion or probability that something happens per unit of time (such as the risk, or more properly the *hazard*, of injury per month

or per game). These variables need to be analyzed with Proc Glimmix.

You tell Proc Glimmix how your data are distributed: Poisson for counts and binomial for proportions and proportions per unit time. The linear model predicts mean values, but the effects are factors: we speak of a 30% increase in the risk of dying if you do or don't do such-and-such, which means a factor increase of 1.3. Hence there has to be some kind of log transformation to go from a multiplicative model to an additive or linear model. The transformation is specified as a **link function**: **log** for counts, **logit** (log-odds) for proportions, and **cloglog** (complementary log-log) for proportions per unit time. Glimmix doesn't actually transform the data, but it fits the data to the transformed model, and it back-transforms the effects to factors. The **log** transformation is simple enough, but **logit** is a bit mysterious: what you actually model is not the log of the proportion (p), but the log of proportion expressed as odds, $\log(p/(1-p))$. Hence the effects back-transform to odds ratios, and they need further transformation to proportion ratios or proportion differences, if you are to make any sense of them. The **cloglog** transformation is even more mysterious: here you are going from data representing a real proportion in a real period of time to an infinitesimal proportion in an infinitesimal unit of time, achieved with the link function $\log(-\log(1-p))$; the effects back-transform to hazard ratios (ratios of proportions per unit time, representing instantaneous relative risks). A more advanced procedure for dealing with logs of hazards is called proportional-hazards regression, realized with Proc Phreg.

The fixed effects model with Proc Glimmix is the same as with Proc Mixed, but you are specifying a linear model predicting log counts, log odds, or log hazards. The random-effects model is also the same, but the residuals are different: they are automatically estimated as residuals you would expect for an observed number of counts,

or for an observed proportion of events for a given number of trials, using the respective distributions, Poisson or binomial. The actual distribution may differ from the expected Poisson or binomial by being *over-dispersed* or *under-dispersed*, meaning that the residual variance is greater or less than the expected variance. So, you allow for that by specifying:

```
random _residual_;
```

There is no **repeated** statement, but you can specify different residual variances with, for example:

```
random _residual_/group=Sex;
```

Incidentally, you can get Glimmix to analyze normally distributed data, by specifying a normal distribution. Glimmix assumes the unit normal distribution, i.e., a variance of 1, hence you must use **random _residual_** to get the actual variance of the residuals, otherwise the analysis won't make sense. The link function would be either the identity link, or if your data need log transformation and you let Glimmix do it, the log link. But don't go there; use Proc Mixed.

References

- Hopkins WG. (2006). A spreadsheet for combining outcomes from several subject groups. *Sportscience* 10, 51-53.
<https://sportsci.org/2006/wghcom.htm>
- Hopkins WG. (2022a). Mixed-modeling workshop in SAS Studio. *Sportscience* 26, ii-iii.
- Hopkins WG. (2022b). Replacing statistical significance and non-significance with better approaches to sampling uncertainty. *Frontiers in Physiology* 13:962132, doi: 10.3389/fphys.2022.962132.
- Smith TB, Hopkins WG. (2011). Variability and predictability of finals times of elite rowers. *Medicine and Science in Sports and Exercise* 43, 2155-2160.

Acknowledgements: Thanks to Basilio Pueo for his interest and enthusiasm in our Skype sessions, and for his patience and valuable feedback with the simulations and manuscript.

Published Sept 2023 ©2023