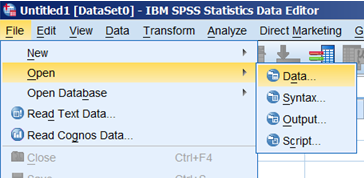
**SPSS Basics and Reliability Mixed Models**

This document gets you started with SPSS, including basic stats and simple reliability and validity analyses. You then find yourself in the deep end of the pool with mixed models for reliability. Don't do t tests and such simple analyses with SPSS, because my spreadsheets do them much better.

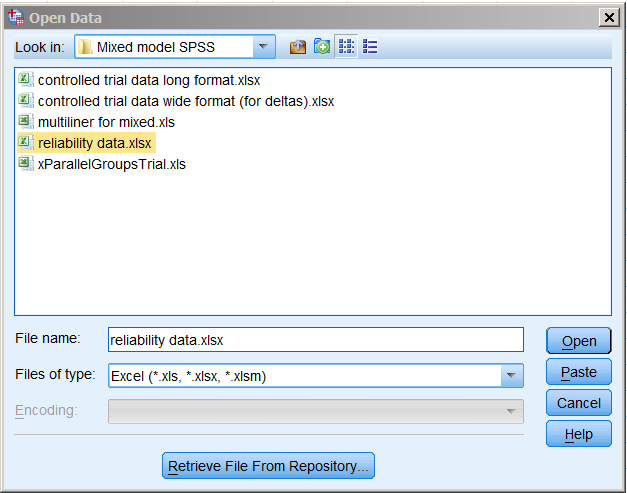
[Getting Started](#start)  
[Simple Stats](#simplestats)  
[Graphs](#graphs)  
[Reliability with the SPSS Built-in Procedure](#relySPSS)  
[Validity](#validity)  
[Mixed Modeling for Reliability](#mixedrely)  
[Extending the Reliability Mixed Model to Monitoring and Clustering](#extending)

**Getting Started**

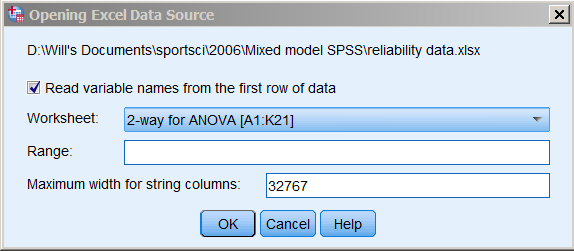
1. It's best to get your data all ready to analyze in Excel, then import.
2. Let's start with a straightforward reliability analysis using SPSS's built-in procedure based on ANOVA. For this procedure, data have to be in "wide" format; that is, the repeated measurements go across the page, with each repeat as a new variable (Trial1, Trial2, or whatever).
3. Open SPSS and import data from the file "reliability data.xlsx", as follows:



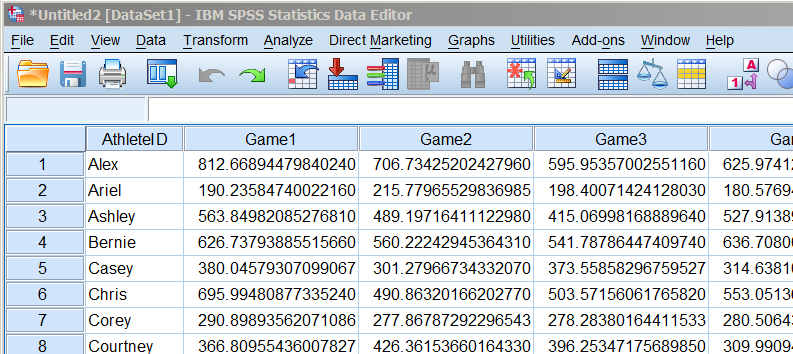
Choose files of type **Excel (\*.xls, \*.xlsx, \*.xlsm)**, find the file, and click **Open**:



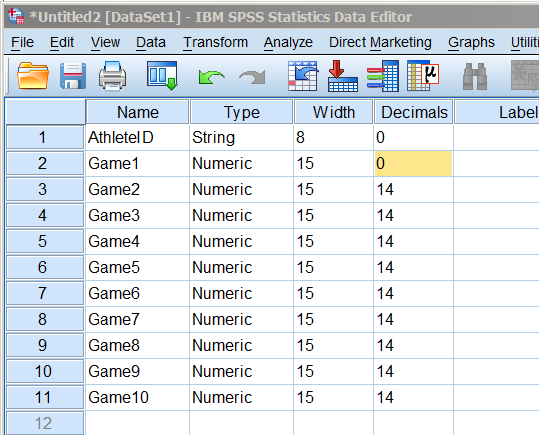
Choose the worksheet "2-way for ANOVA", then OK:



1. The ridiculous number of decimal places is due to the way I generated the data in Excel:

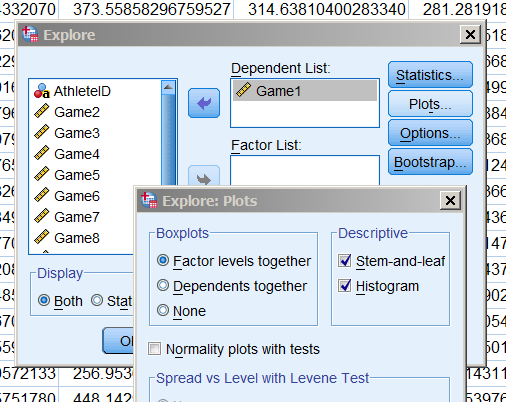


1. Switch to **Variable View** (a tab at bottom of the window) and get rid of all the decimals, if you want to:



**Simple Stats**

1. From the menu bar, select **Analyze/Descriptive Statistics/Descriptives…** for simple stats. You play with it
2. From the menu bar, select **Analyze/Descriptive Statistics/Explore**, select Game1 and click it into the Dependent List, click Plots and tick Histogram as well as the default Stem-and-Leaf:



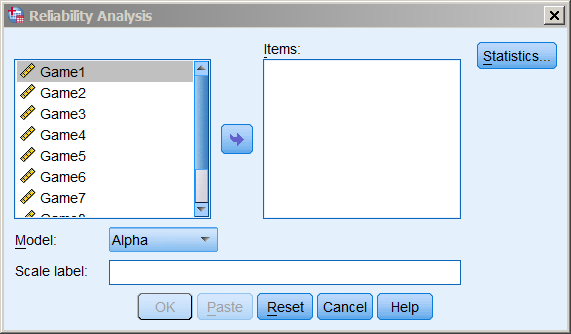
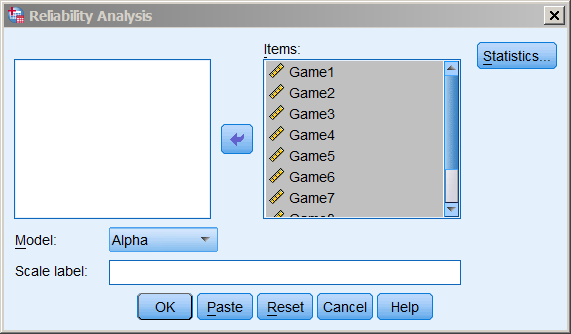
1. Then click **Continue** and **OK**.
2. You can figure out the output. This resource is not a lesson in elementary statistics!

**Graphs**

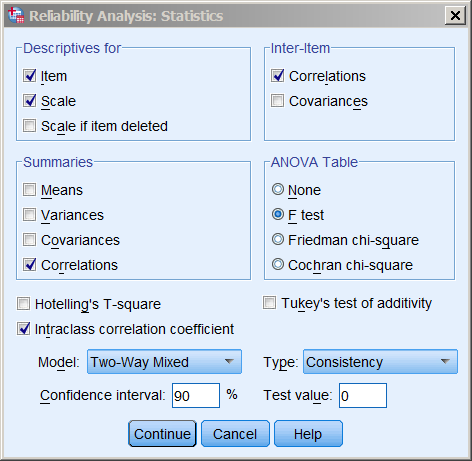
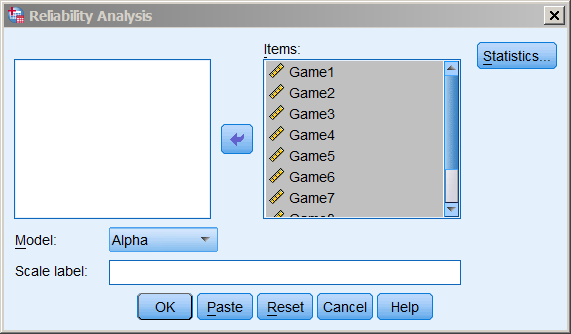
1. You can copy graphs from SPSS into PowerPoint and take them apart to clean them up. Write click on the histogram in SPSS, select **Copy Special…**, select **Metafile (WMF, EMF)**, then **OK**. (You can leave the other formats ticked.).
2. Now go into PowerPoint, select **Home/Paste/Paste Special...**, then select **Picture (Enhanced Metafile)**, then **OK**. (I customize my Office ribbons to have Paste Special and heaps of other tasks I perform frequently within each of the Office programs; e.g., Group and Ungroup–see next paragraph.)
3. Drag a corner of the graph to make it the size you want, then **Home/Arrange/Ungroup**. Click **Yes**, then **Ungroup** it again. Now you can clean things up. Unfortunately text coming in from SPSS can't be edited, so you might have to delete it and replace with tidier text in PowerPoint. I suggest Arial Narrow for most purposes. Make the font sizes LARGE enough: preferably 24 pt for slides.

**Reliability with the SPSS Built-in Procedure**

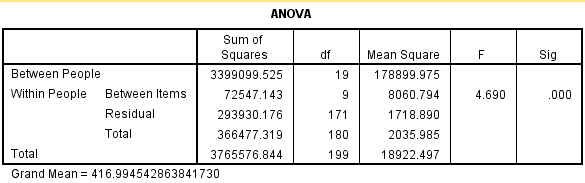
1. Use this procedure to get alpha reliability of a set of Likert-scale items from a questionnaire. Use my reliability spreadsheet for consecutive pairwise analyses of any test or other measures where familiarization or habituation or time between trials could modify reliability. Use a long-form mixed model when there are clusters of trials.
2. From the menu bar, select **Analyze/Scale/Reliability Analysis…**, and put all the games into the Items:

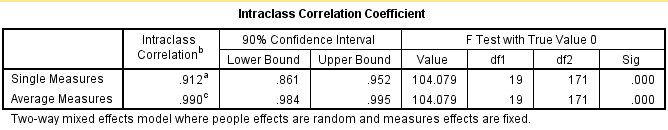
1. Keep the default model **Alpha** (which also works out the alpha reliability correlation when the selected items are those from a set of items from a psychometric inventory).
2. Click **Statistics…**, and select **Descriptives** for Item and Scale, Inter-item Correlations, Summaries Correlations, ANOVA Table F test, Intraclass correlation coefficient, the default Model Two-Way Mixed, the default Type Consistency, and 90% confidence limits, as shown. Click **Continue**, then **OK.**

1. Most things are obvious in the output, but the typical error has to come from the ANOVA table. It is the square root of the Mean Square for the Residual term (here √1718.890–you have to do it in Excel).



1. The intraclass correlation coefficient (equivalent to the average of all pairwise correlations) is the value for Single Measures. Cronbach's alpha is the value for Average Measures:



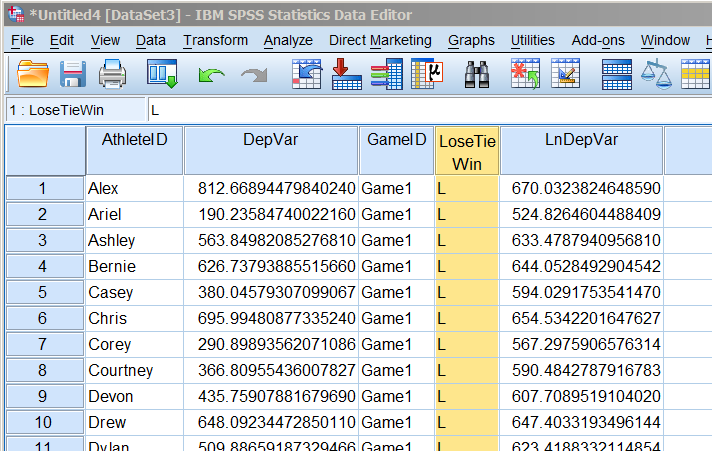
1. We'll see how these values compare when we do the analysis with a mixed model. Meantime compare with the values provided by the reliability spreadsheet.

**Validity**

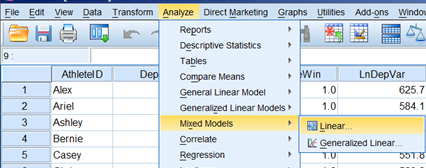
1. Briefly, because my validity spreadsheet does it better!
2. From the menu bar, select **Analyze/Regression/Linear**.
3. Select the criterion as the Dependent and the practical as the Independent.
4. Click **Statistics**, and select Estimates, Confidence Intervals, Model fit and Descriptives, then **Continue**.
5. The defaults for options are OK.
6. Click **OK**. What you want in the output should be obvious.

**Mixed Modeling for Reliability**

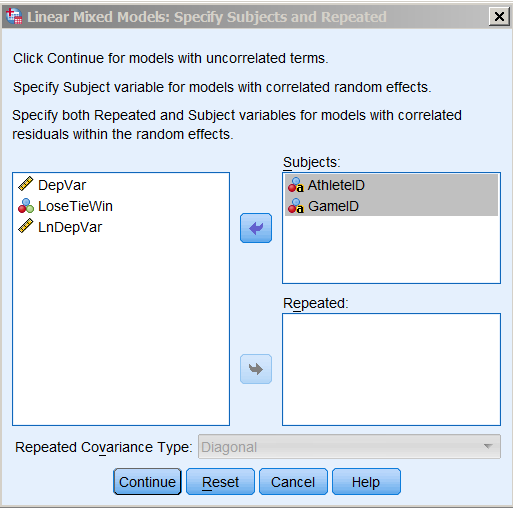
1. The simplest mixed model with repeated measurement is a reliability model. Open SPSS and import the file "reliability data.xlsx", and choose the worksheet "2-way for mixed model".
2. Make the LoseTieWin column a bit wider so you can read it:



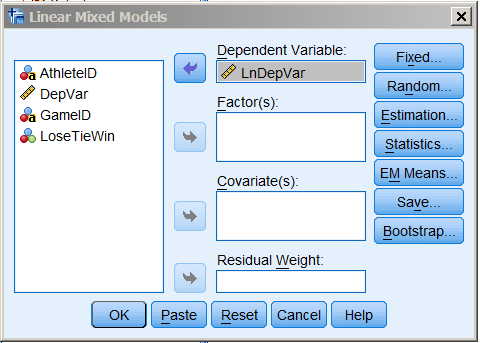
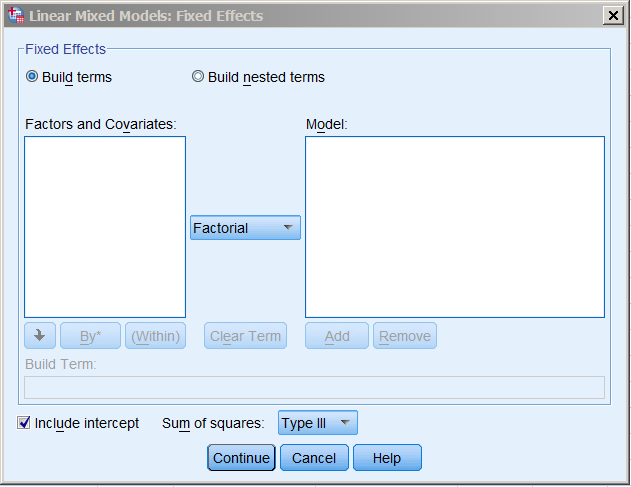
1. Scroll down to see that you have imported the same data as previously, but in "long" format, and with the additional LoseTieWin variable that we can have fun with soon. Also you now have LnDepVar, which is 100 times the natural log of DepVar, the dependent variable representing something like meters of high-speed running in a game.
2. Choose **Analyze/Mixed Models/Linear…**:



1. Make AthleteID and GameID the **Subjects** (we will use AthleteID only, first), then **Continue**:

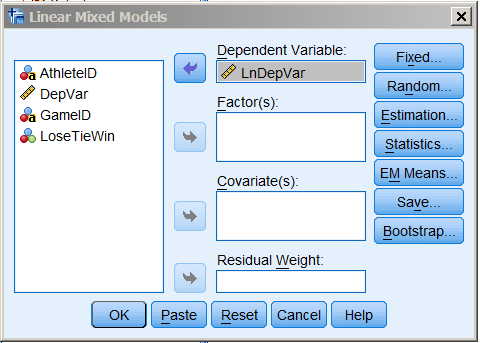


1. Choose LnDepVar as the **Dependent Variable**. then **Fixed**.

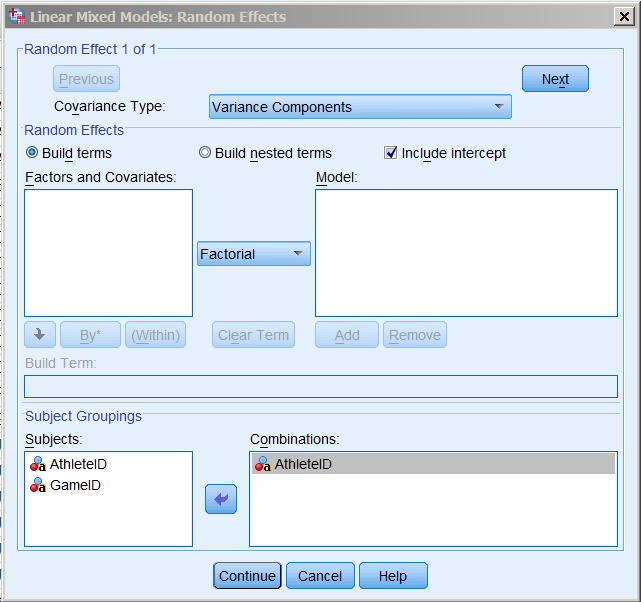
 

Nothing to do here. **Continue**.

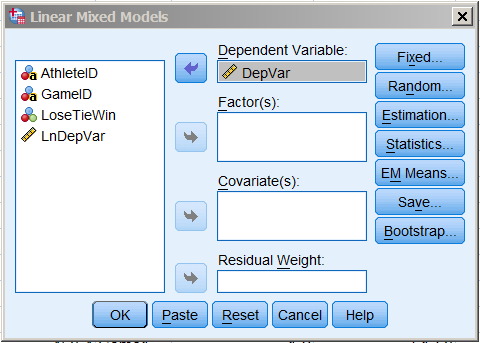
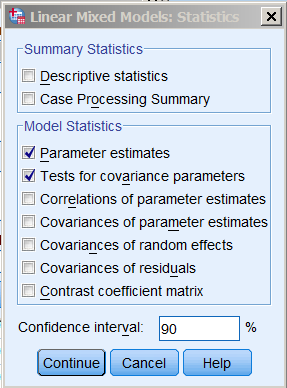
1. Click **Random**:



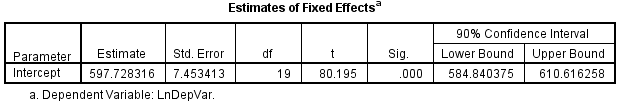
1. Put AthleteID into **Combinations** and select **Include intercept** (too difficult to explain this jargon; all it means is that we're going to estimate a variance for the athletes). At this stage we won't run GameID as a random effect. So click **Continue**:



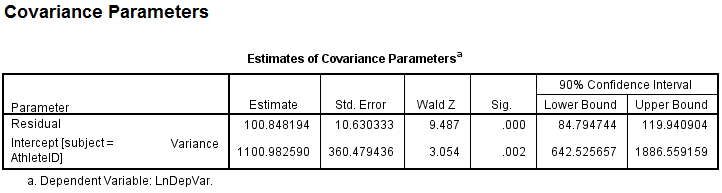
Now click on **Statistics**:

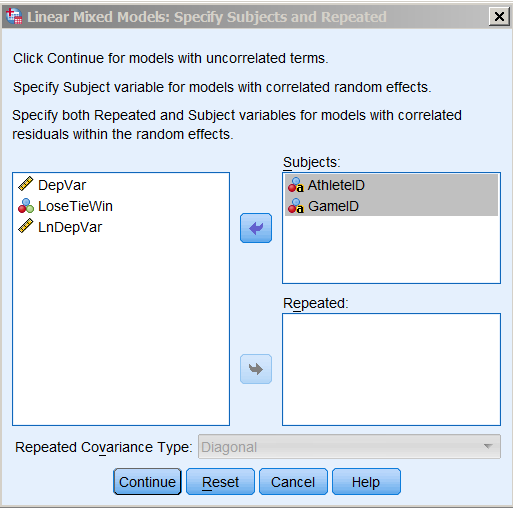
1. Choose the above two options, change the confidence interval to 90%, then **Continue**, then **OK**:
2. This bit of the output is just the overall mean:



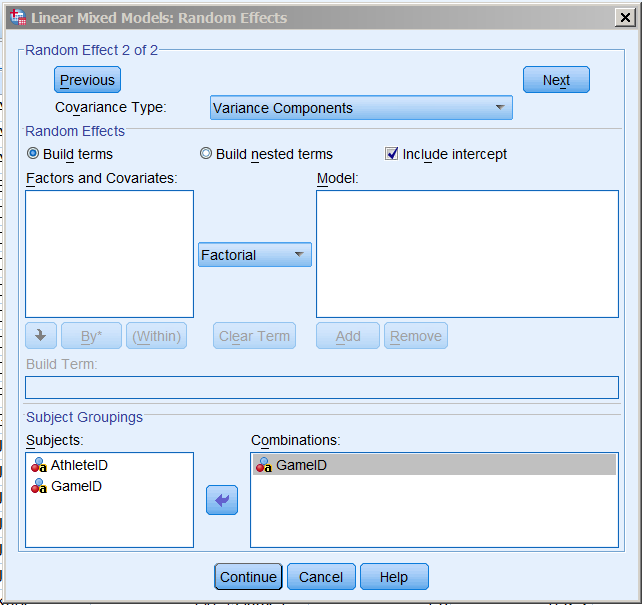
1. Here's the part of the output with the random effects (aka Covariance Parameters):



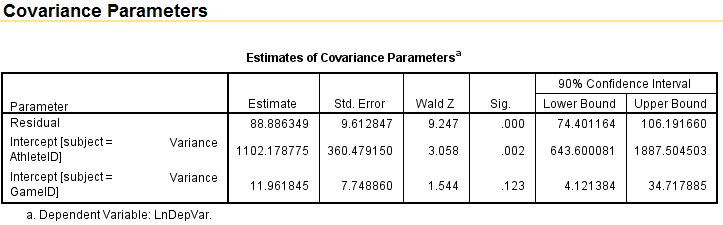
1. These are variances. Compare with the analysis in the 1-way reliability spreadsheet.
2. You have to take square roots of the estimate and confidence limits, then back-transform them to percents. The ICC is given by (AthleteID variance)/(AthleteID variance + Residual). Confidence limits for the ICC are a bit more difficult.
3. Now let's do the analysis with GameID included as a random effect. In other words, imagine that the games are also a random sample.
4. Choose **Analyze/Mixed Models/Linear** again. You should see this, again:



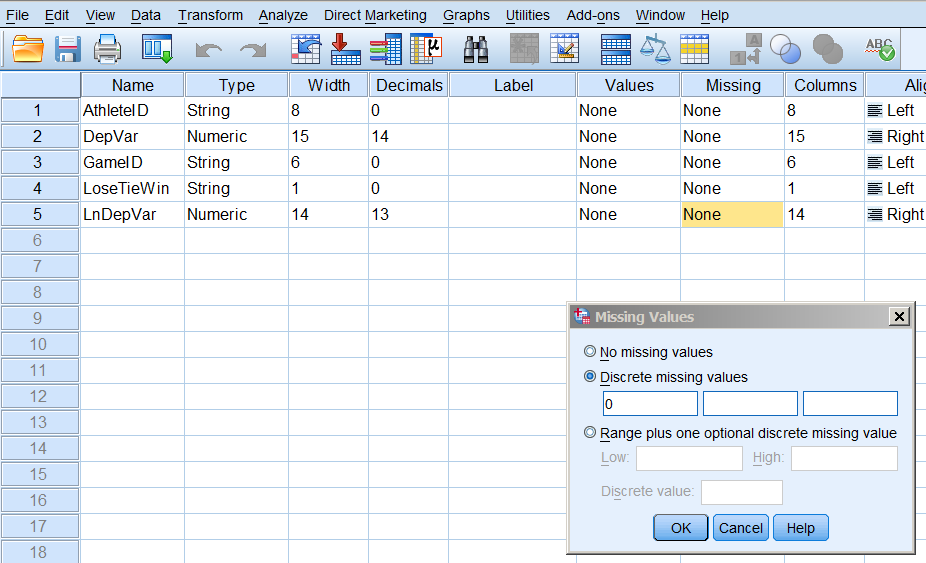
1. Click **Continue**, and choose **Random**. Owing to a bug, you will have to do something to allow you to click **Next**. I unselected Include intercept, then selected it again. Now click **Next** and put GameID into **Combinations**. Make sure you tick **Include intercept** before you **Continue**. Then **OK**.



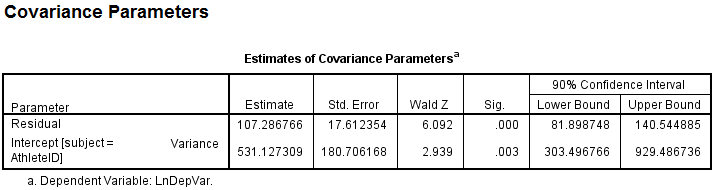
1. Here's the output you want:



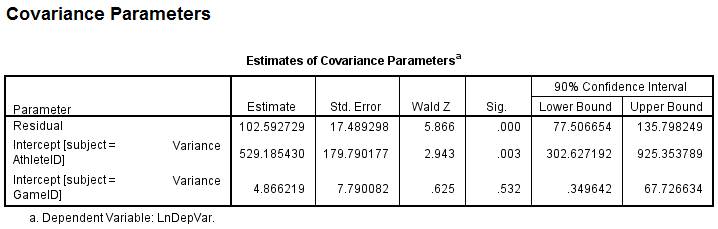
1. Compare with the analysis in the 2-way reliability spreadsheet.
2. As before, these are variances. You have to take square roots of the estimates and confidence limits, then back-transform them to percents. The ICC is again given by (AthleteID variance)/(AthleteID variance + Residual). Confidence limits for the ICC are even more difficult.
3. Now, for practice, repeat the above with data with missing values. Use the same workbook and import the spreadsheet "2-way missing for mixed model". Missings are shown as zero in that spreadsheet (this time it's a Microsoft bug), so when you have imported the data, switch to **Variable View** (bottom left-hand end of the data window) and set values of 0 to missing:



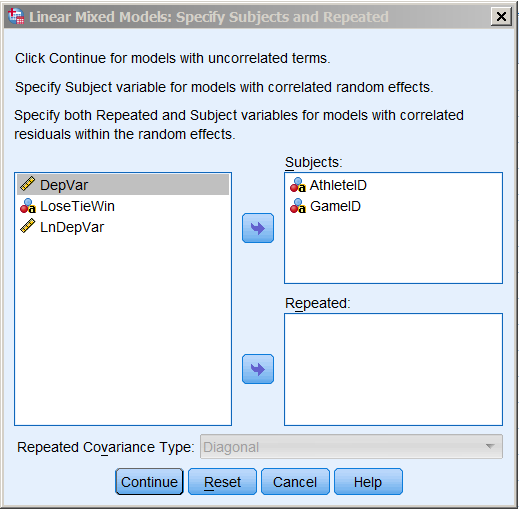
1. Alternatively, delete all rows with values of zero for LnDepVar, or just make the zeros blank.
2. Here's the output from the 1-way analysis, which you can compare with the spreadsheet:



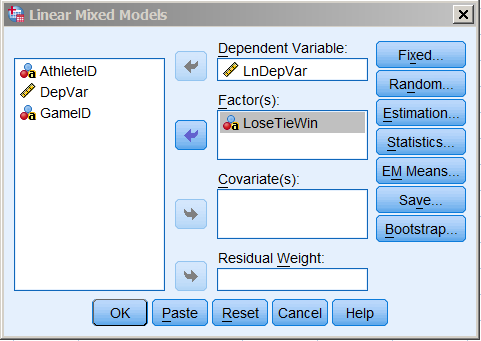
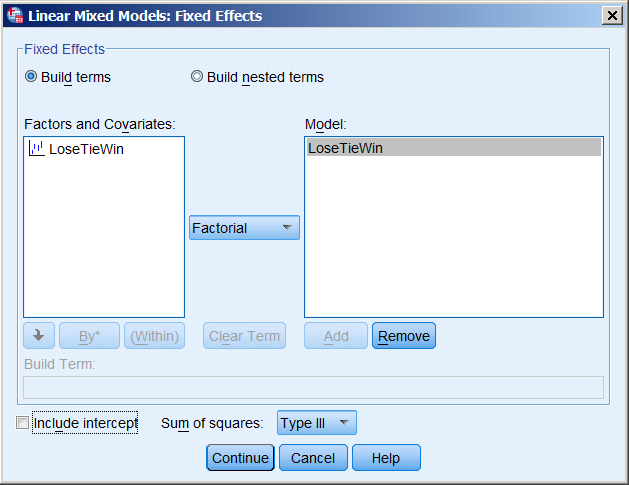
1. And here's the output from the 2-way, which I have put in the spreadsheet, but which doesn't work:



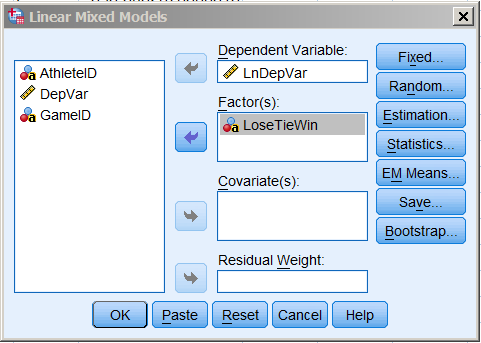
1. Now let's estimate the effect of WinTieLose, first of all with the dataset without missing values. Bring that dataset to the front, or you will end up analysing the dataset with missing values.
2. Choose **Analyze/Mixed Models/Linear** again. You should see this again:



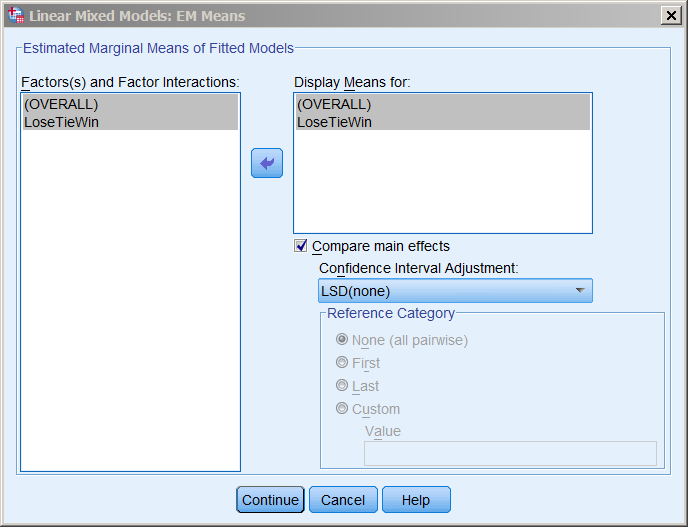
1. Click **Continue**, and put LoseTieWin into **Factor(s)**. Then **Fixed**:

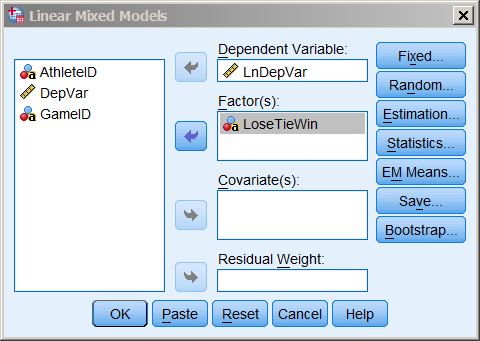
1. Add LoseTieWin to the **Model**. The **Factorial** button doesn't matter here. Also, it doesn't matter whether you **Include Intercept** or not, but it's slightly less confusing if you unselect it. (If you select it, the last of the three means for LoseTieWin is assigned zero in the Parameter Estimates. But we will use EM Means next, and these work regardless.) **Continue**:
2. Select **EM Means**. This stands for estimated marginal means, another name for least-squares means (when the analysis is via ANOVA; here it's via restricted maximum likelihood). The EM means are the means of the levels of nominal predictors (factors) evaluated at the mean value of all covariates and averaged over all levels of all other fixed and random nominal predictors.



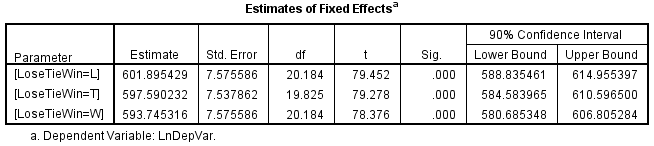
1. Put Overall and LoseTieWin into **Display Means for**, and tick **Compare main effects**. Then **Continue**:

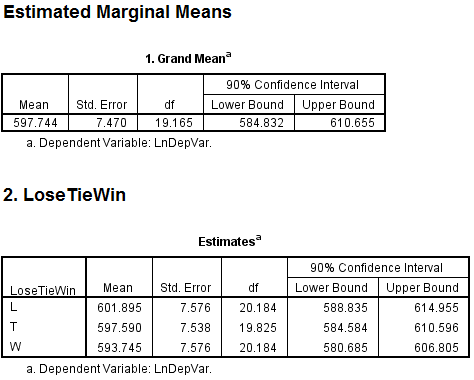


1. Click **OK**:

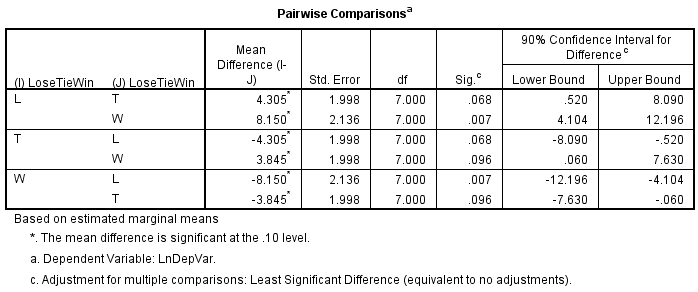


1. The Estimates of Fixed Effects is the output of the Parameter Estimates (one of the boxes we ticked for statistics long ago). These are the same as the Estimated Marginal Means.

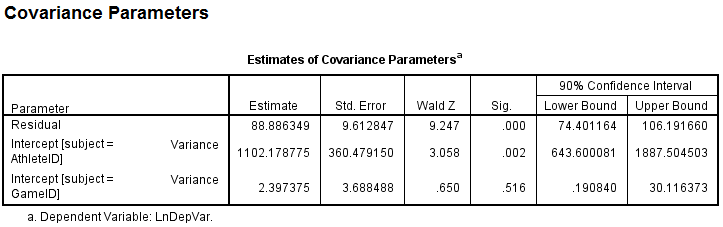




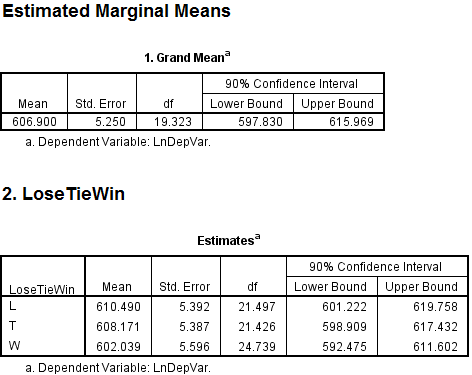
1. And here's the comparisons of the levels of LoseTieWin. These have to be back-transformed, but they are almost exactly percents. Our population values for these were 5%, which are contained within the confidence intervals:

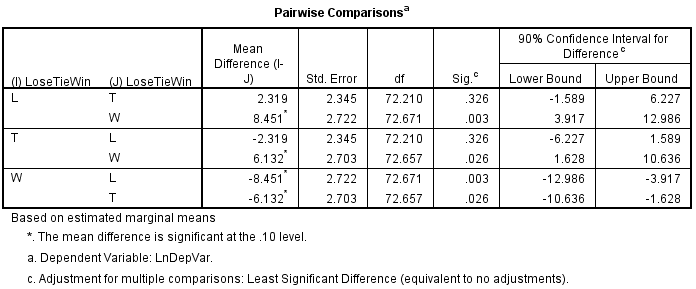


1. Here's the random effects. Notice that the value for GameID is now smaller (because we have accounted for some of the differences between games with the LoseTieWin fixed effect):

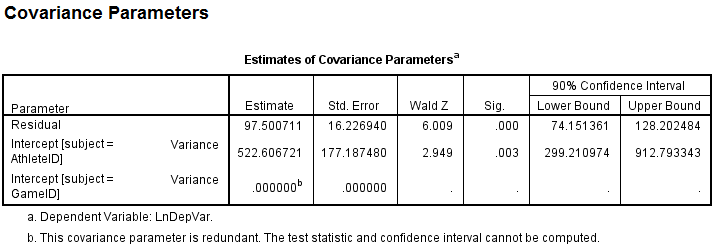


1. Now bring the dataset with missing values to the front, and repeat the analysis. You get similar answers for the fixed effect, but with wider confidence intervals, because there are less data:





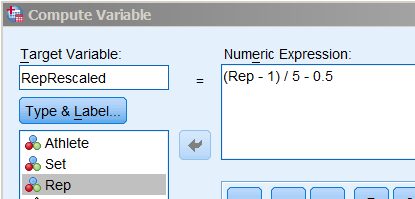
1. The random effects now show something a bit strange…



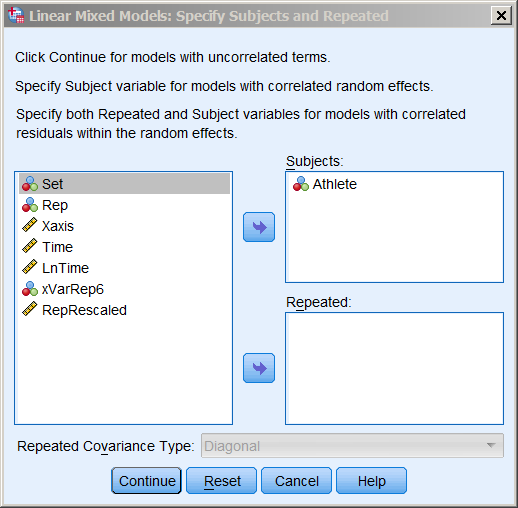
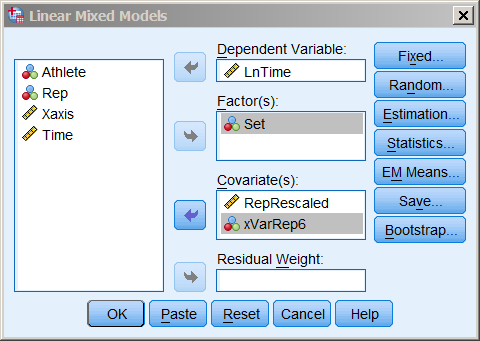
According to this output, the random effect for GameID is redundant. That is, the model is overspecified, or there aren't enough data to estimate all the parameters that we've asked for. Well, that's not strictly true. What it really means is that, owing to sampling variation, the variance for this random effect is negative, and SPSS can't estimate negative variance, so it sets it to zero. SAS can estimate negative variance, and the confidence limits for the variance would tell us the extent to which the variance could be positive. The upper confidence limit would almost certainly be positive. In this case SAS would give better estimates for all the other effects, although the differences would probably be negligible.

**Extending the Reliability Mixed Model to Monitoring and Clustering**

1. Imagine that, in addition to LoseTieWin, we have a variable for date of each game, and we have all the games in a season. We could include Date as a numeric fixed effect and model a seasonal trend in whatever dependent variable. Or instead of numeric Date, we could have a nominal fixed effect with levels representing different phases of the season.
2. Because these extra effects have values that change within each subject, we could also model individual differences in their effects, by including them interacted with the subject random effect. We could have done that with the WinTieLose effect: maybe some subjects don't change much between games where the team loses instead of winning.
3. Here's a fairly simple example of real data with clusters and trends. The data come from a study of the effects of caffeine on repeated sprints, done as a crossover with placebo and several doses of caffeine, each a week apart. On each of the three testing days, the athletes performed four sets of six all-out 30-m sprints. To keep it simple, I've included the data only for the placebo. The aim of the analysis is to quantify how much the subjects fatigue in each set of six sprints, taking into account and estimating individual differences in the fatigue. We'll also account for any consistent changes in mean sprint time between sets (i.e., is there fatigue between sets as well as within sets), and any variability in each subject's mean sprint time from set to set (which probably won't make sense to you until we develop the model). Oh, and most subjects make a bigger effort on the last sprint of a set, so we'll account for and estimate that, too, along with individual differences!
4. Import "sets of reps long format.xlsx" and inspect the data. Most of the variables are obvious. I added Xaxis in the hope that someone can show me/us how to plot the means and SD for each rep spread out over the four sets, so we can see the fatigue, the changes between sets, and the faster last sprint. I also added a dummy variable for the last sprint in each set.
5. For reasons that will become clear, we need to produce a new Rep variable that goes from -0.5 to 0.5 rather than 1 to 6. We could do that easily in Excel, of course, but let's see how to do it in SPSS.
6. Select **Transform/Compute Variable…**, and make a variable RepRescaled, as shown. Check its values.

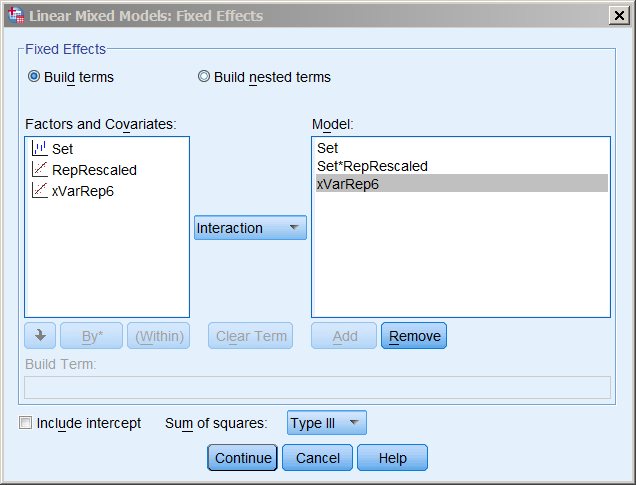


1. **Analyze/Mixed Models/Linear…** and select Athlete, **Continue** and choose the variables shown:

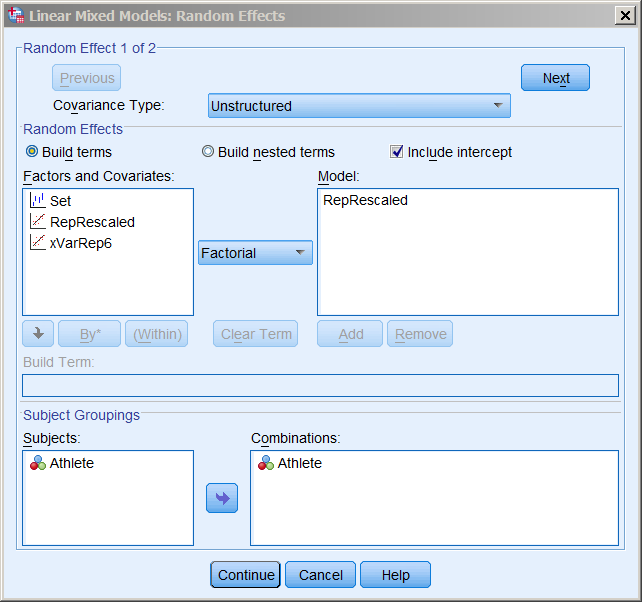
 

Set is a factor, because we'll look at the mean for each set. RepRescaled in a covariate, because fatigue usually develops in a linear fashion with repeated sprinting, and we want an overall estimate of what I call linearized fatigue from first to last sprint. That is, we will estimate the linear component of fatigue over six sprints. If we had used Rep rather than RepRescaled, we'd have to multiply its coefficient by 5 (yes, 5, not 6) to get the overall fatigue in six sprints. OK that's not a big deal, but it makes the individual differences in the slope a lot easier to interpret, as we will see.

1. Click **Fixed** and build this model. Make sure Include intercept is not ticked:

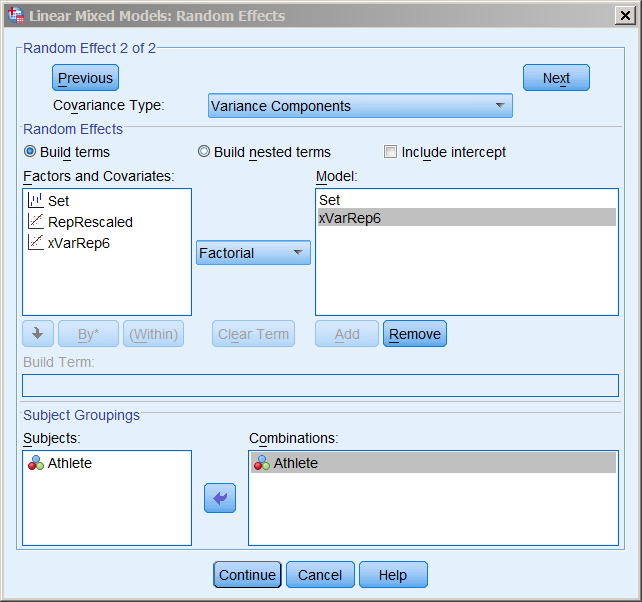


1. This model implies we're going to estimate a different mean for each set, a different mean slope for each set, and a different mean time for the last rep across all the sets. Because we're already estimating mean time in each rep in each set with the other two terms, the effect for this dummy will be the amount by which the athletes on average go faster than a linear fatigue model predicts for the last rep.in each set.
2. Click **Continue**, then **Random**…, and create this model:

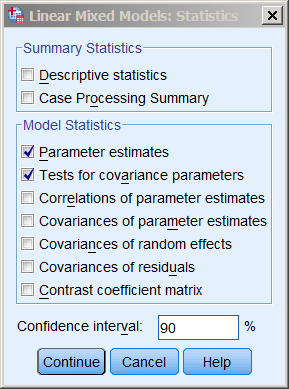
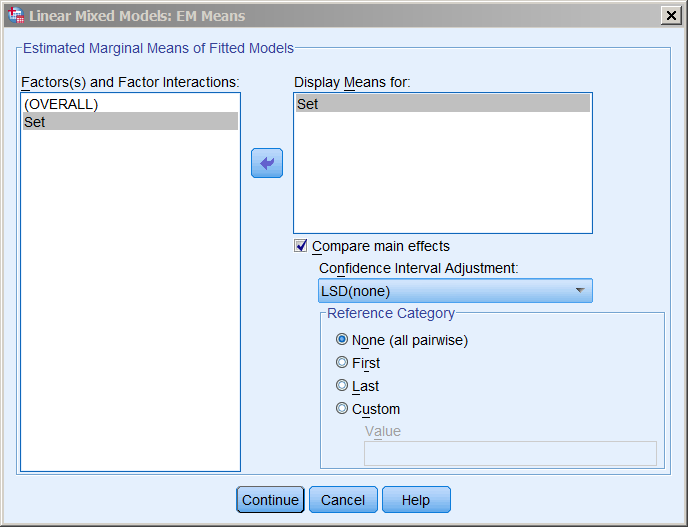


Note especially **Unstructured** and tick Include intercept. This model generates a random effect for athlete (they differ in how fast they sprint), and a random effect for Athlete\*RepRescaled (they differ in how much they fatigue). How cool is that?! You have to make it unstructured to allow these two effects to be correlated: faster athletes might fatigue more, for example. It's called an individual-slopes model.

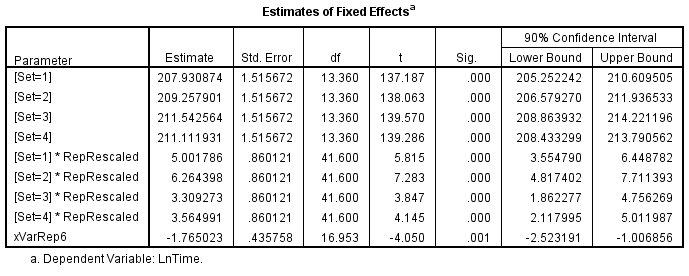
1. Click Next and build the random effects shown. Set (i.e., Athlete\*Set) represents the variability in an athlete's mean time between sets, which might not be much here, but it would be more if there were days rather than minutes between sets. xVarRep6 (i.e., Athlete\*xVarRep6) represents individual differences (consistent over the 4 sets) from the mean value provided by the fixed effect for xVarRep6. More below.



1. Click **Continue** and choose the usual things for **Statistics**, then **Continue**. For **EM Means** tick **Compare main effects** with no adjustment for multiple comparisons (**LSD(None)**). **Continue** and **OK**.

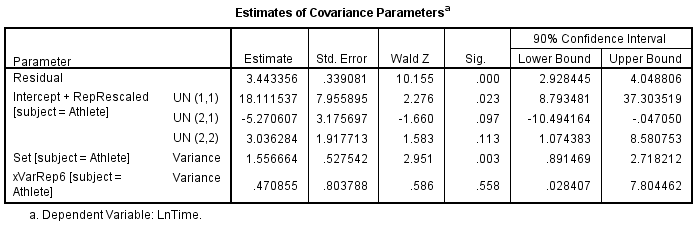
 

1. Let's deal with the output in the order produced.

  
**[Set = 1]** etc. are the means of each set, sort of. The marginal means are better. See later.  
  
**[Set = 1]\*RepRescaled** etc. are the fatigue slopes evaluated over the six sprints, and they are small enough to be almost exactly percent increases in time: 100\*exp(5.00/100)-100 = 5.1%. So there is a ~5-6% fatigue in the first two sets, but only around 3.5% in the last two.  
  
**xVarRep6** is -1.8%, which means the athletes went faster than predicted in the last rep overall by 1.8%.

If the smallest important difference or change in sprint time is 1.0% (see below), the fatigue effects and the faster effect in the last rep are all clearly substantial. The 6.2% is actually large.

1. Here are the random effects, which need to be square rooted and back-transformed to make them easier to interpret. See below.



**UN(1,1)** is what SPSS and other stats packages call the variance of the first random effect in the list of effects that are included in the unstructured set of effects. Here that's the subject random effect. Think: each athlete gets their very own unique number out of this hat, representing how much faster or slower they are than the average.

**UN(2,2)** is the variance for the second in the list, here Athlete\*RepRescaled. Think: RepRescaled is a slope, and a different value has to come out the hat for each athlete. It therefore represents individual differences from each of the mean slopes in the four reps. Note that we are estimating a single consistent difference from each of the four different means. It would be possible to estimate a different individual difference for each rep, but what's the point? We want some indication of how consistently an athlete differs from other athletes across all four reps.

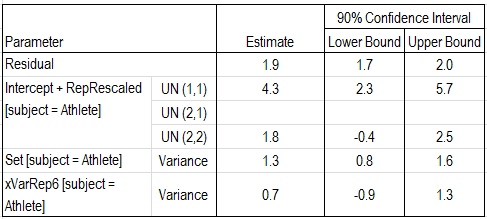
**UN(2,1)** is the covariance of these two effects, basically a term like a variance indicating how much they go together. (It could also be written UN(1,2).) When it's negative, as here, it means they go in opposite directions: the faster an athlete, the bigger the drop-off.

**Set [subject = Athlete]** is the Athlete\*Set random effect. Think: if you change the athlete or you change the set, another random number has to come out of this random-effect hat, so this represents the set-to-set variability an athlete shows in the mean of the athlete's reps of each set. This is a classic within-subject random effect when you have clusters of repeated measurements (here sets of repeated sprints).

**xVarRep6 [subject = Athlete]** is the Athlete\*xVarRep6 random effect. Think: xVarRep6 is zero for all but Rep 6, when it is 1, so each athlete gets a single value out of this hat, and it represents the athlete's way of dealing with Rep 6, either faster or slower than the mean reduction we've seen earlier.

**Residual** is the random error that accompanies every sprint. Think: for every sprint, a number has to come out of this hat. It's analogous to the typical error in a straightforward reliability study. By the way, the residual is equivalent to a random effect specified by Athlete\*Set\*Rep.

Here are the square roots and properly back-transformed estimates and confidence limits in percent units (except for UN(2,1), which is not modified and can be ignored):

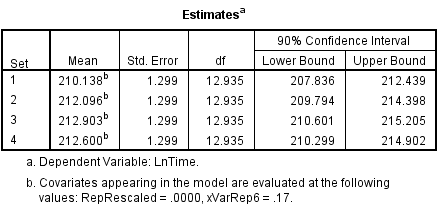


The calculations were done with "convert SPSS covparms to SDs and CLs.xlsx". Check it out. I justify the formulae for the confidence limits in the next resource, on controlled trials.

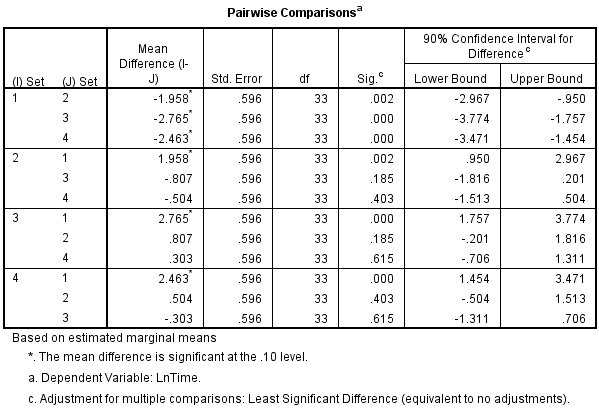
1. To properly assess these SDs, we need a smallest important difference or change. For team-sport athletes (which these were), the default is 0.2 of the *observed* between-athlete SD in any given testing occasion. Normally this SD would be given by adding up all the variances that contribute to a single sprint in any given testing session, and here it would be the variance for the stable (or true or pure) between-athlete differences = UN(1,1) plus the variance an athlete shows in any given set = Set[subject = Athlete], plus the variance an athlete shows in any given sprint = Residual. After back-transformation, the value is 4.9%, as shown in the spreadsheet. With RepRescaled deliberately having a value of 0 in the middle of the set of sprints, the individual responses in the fatigue don't contribute to the SD in the middle of the sprints, but they would on either side. The negative covariance means that athletes with faster sprints have greater fatigue, so that apparently means there would be a bigger observed between-subject SD at the start of the sprints and a smaller one at the end. The SD in the middle is therefore a reasonable value to use, and 0.2 of 4.9% is 1.0%.

The other thresholds as factors of the between-athlete SD are 0.6, 1.2, 2.0 and 4.0 for moderate, large, very large and extremely large. Hence the full set of thresholds here are near-enough to 1%, 3%, 6%, 10% and 20%. Actually you are supposed to multiply the 0.2 etc. by the log-transformed SD, then back-transform them. See the spreadsheet.

1. But wait, you have to double SDs to interpret their magnitudes, or equivalently halve the thresholds. You don't evaluate the Athlete random effect or the UN(2,1), but you evaluate everything else. I can see that the observed effects range from small to moderate, and only the xVarRep6 is unclear. I don't normally do a full probabilistic magnitude-based inference on an SD. I think it's enough to interpret first the observed value, then the upper confidence limit to see how big it could be, then the lower confidence limit to see if it could be trivial or negative. Some negative SDs occur simply because of sampling variation when the true value is actually trivial or small. Other negative SDs can represent real negative variance, especially individual responses to treatments. See later.
2. The Estimated Marginal Means for Set are the means you would want to use, if you were to report them.

  
They back-transform to means of exp(210.1/100) = 8.2 s, etc. Notice that they are evaluated at the mean values of the covariates.

1. The pairwise comparisons are more interesting:



These are all <3%, so they don't need back-transforming, and the observed magnitudes are trivial to small. You can see that there was 2.0% fatigue between the first and second set, then another 0.8% (trivial) between the second and third, but they revived a little (trivially) in the fourth set, by 0.3%. With a smallest important of 1%, all of these effects would be clear at different levels of likelihood for trivial or substantial.

You can do full magnitude-based inference on these effects via the log-transformed mean difference and its p value with the "Confidence limits and clinical chances" spreadsheet at Sportscience. Get the p value to a few more significant digits by double-clicking the panel of results in SPSS. The smallest important log-transformed value goes into the cell under "benefit or ???". Check that the spreadsheet gives the same confidence limits as shown here. If it doesn't, you've made a mistake. You can also do the MBI via the mean difference and its confidence limits with the spreadsheet "Combine/compare effects". Click on the tab at the bottom to select the sheet for 1 or more groups or statistics, and use a single custom weight with a value of 1.