

Multilevel linear models

Self-test answers



- Using what you know about *ggplot2*, produce the graph described above. Display the levels of **Surgery_Text** in colours, and use **Clinic** to produce different graphs within a grid.

To demonstrate exactly how the graph is built up, we will work through the commands in steps; however, in reality you would probably just use a single command. First, we create an object (which I've called *pgrid*) using the *ggplot()* function. Within this function, we specify the dataframe that we want to use (*surgeryData*) and the variables on the x- and y-axes (**Base_QoL** and **Post_QoL**, respectively).

```
pgrid <- ggplot(surgeryData, aes(Base_QoL, Post_QoL))
```

We then add a title using the *opts()* function. It should be self-evident from the example how this function works.

```
pgrid + opts(title="Quality of Life Pre-Post Surgery at 10 Clinics")
```

Next we want to add some data points. To do this we use the *geom_point()* function, which, left to its own devices, will simply add points on the graph reflecting each value of baseline quality of life against its corresponding value of quality of life after surgery. However, we'd like these data points to differentiate whether a person had surgery or was on the waiting list. We have a variable in the file that specifies, as text, whether someone had surgery or was on the waiting list (**Surgery_Text**). We can specify that the colour of the point is determined by the value of the variable **Surgery_Text**. This is done by including *aes(colour = Surgery_Text)* within the *geom_point()* function.

```
pgrid + geom_point(aes(colour = Surgery_Text))
```

So far, so good. It would be fun, though, to plot a regression line as a summary of the relationship between baseline and post-surgery quality of life. We can do this using the *geom_smooth()* function. As with our data points, we want separate lines for those who had surgery and those on the waiting list. We achieve this in exactly the same way by including *aes(colour = Surgery_Text)*. We also need to tell *geom_smooth()* which type of model we want to fit; we want a linear model, so we include *method = "lm"*, and to keep things simple we don't want to plot the confidence interval around the regression line, so we set *se = F*, to switch these off (*se = F* means 'standard error = FALSE'; if you want to switch the confidence intervals on you can change this option to TRUE, or *se = T*, or simply omit the option because the default option is to plot the standard errors).

```
pgrid + geom_smooth(aes(colour = Surgery_Text), method = "lm", se = F)
```

We also wanted to plot different graphs for each clinic. An efficient way to plot multiple graphs showing the same thing is to use the *facet_wrap()* function, which will create a grid of graphs with each grid location representing a level of the specified variable. In this case we specify **Clinic** as the variable. This will create a grid of graphs; however, I specifically want a grid with five columns and two rows, so I have specified that I want five columns by typing *ncol = 5*.

```
pgrid + facet_wrap(~Clinic, ncol = 5)
```

Lastly, we can tidy up the axis labels by specifying the text that we wish to appear (the text will be printed exactly how you type it within the quotes, so remember to capitalize as you see fit and check your spelling!

```
pgrid + labs(x = "Quality of Life (Baseline)", y = "Quality of Life (After Surgery)")
```

We can write all of this in a single command as follows:

```
pgrid <- ggplot(surgeryData, aes(Base_QoL, Post_QoL)) + opts(title="Quality of Life  
Pre-Post Surgery at 10 Clinics")
```

```
pgrid + geom_point(aes(colour = Surgery_Text)) + geom_smooth(aes(colour =
Surgery_Text), method = "lm", se = F) + facet_wrap(~Clinic, ncol = 5) + labs(x =
"Quality of Life (Baseline)", y = "Quality of Life (After Surgery)")
```



- Using what you know about ANOVA, conduct a one-way ANOVA using **Surgery** as the predictor and **Post_QoL** as the outcome.

You can run the ANOVA by using the following commands to create an object called *surgeryANOVA* and then using the summary function to show the output from the ANOVA:

```
surgeryANOVA<-aov(Post_QoL~Surgery, data = surgeryData)
summary(surgeryANOVA)
```



- Using what you know about ANCOVA, conduct a one-way ANCOVA using **Surgery** as the predictor, **Post_QoL** as the outcome and **Base_QoL** as the covariate.

You can run the ANCOVA by using the following commands to create an object called *surgeryANCOVA* and then using the *summary* function to show the output from the ANCOVA. Remember, that the *aov()* function produces Type I sums of squares so we use the *Anova()* function to get the Type III sums of squares that you'll be used to seeing:

```
surgeryANCOVA<-aov(Post_QoL~Base_QoL + Surgery, data = surgeryData)
summary(surgeryANCOVA)
Anova(surgeryANCOVA, type="III")
```



- We have used the *update* function in this second example. To get some practice at specifying multilevel models, try building each of the models in this example but specifying each one in full.

```
intercept <-glsl(Life_Satisfaction~1, data = restructuredData, method = "ML", na.action
= na.exclude)
```

```
randomIntercept <-lme(Life_Satisfaction ~1, data = restructuredData, random =
~1|Person, method = "ML", na.action = na.exclude, control = list(opt="optim"))
```

```
timeRI<-lme(Life_Satisfaction~Time, data = restructuredData, random = ~1|Person,
method = "ML", na.action = na.exclude, control = list(opt="optim"))
```

```
timeRS<-lme(Life_Satisfaction~Time, data = restructuredData, random = ~Time|Person,
method = "ML", na.action = na.exclude, control = list(opt="optim"))
```

```
ARModel<-lme(Life_Satisfaction~Time, data = restructuredData, random = ~Time|Person,
correlation = corAR1(0, form = ~Time|Person), method = "ML", na.action = na.exclude,
control = list(opt="optim"))
```

```
timeQuadratic<-lme(Life_Satisfaction~Time + I(Time^2), data = restructuredData, random
= ~Time|Person, correlation = corAR1(0, form = ~Time|Person), method = "ML", na.action
= na.exclude, control = list(opt="optim"))
```

```
timeCubic <-lme(Life_Satisfaction~Time + I(Time^2) + I(Time^3), data =
restructuredData, random = ~Time|Person, correlation = corAR1(0, form = ~Time|Person),
method = "ML", na.action = na.exclude, control = list(opt="optim"))
```

```
polyModel<-lme(Life_Satisfaction~poly(Time, 3), data = restructuredData, random =
~Time|Person, correlation = corAR1(0, form = ~Time|Person), method = "ML", na.action =
na.exclude, control = list(opt="optim"))
```

Oliver Twisted

Please Sir, can I have some more ... ICC?

The following article appears in:



Field, A. P. (2005). Intraclass correlation. In B. Everitt & D. C. Howell (Eds.), *Encyclopedia of behavioral statistics* (Vol. 2, pp. 948–954). Hoboken, NJ: Wiley.

It appears in adapted form below.

Commonly used correlations such as the *Pearson product-moment correlation* measure the bivariate relation between variables of different measurement classes. These are known as *interclass* correlations. By ‘different measurement classes’ we really just mean variables measuring different things. For example, we might look at the relation between attractiveness and career success. Clearly one of these variables represents a class of measures of how good looking a person is, whereas the other represents the class of measurements of something quite different: how much someone achieves in their career. However, there are often cases in which it is interesting to look at relations between variables *within* classes of measurement. In its simplest form, we might compare only two variables. For example, we might be interested in whether anxiety runs in families, and we could look at this by measuring anxiety within pairs of twins (Eley & Stevenson, 1999). In this case the objects being measured are twins, and both twins are measured on some index of anxiety. As such, there is a pair of variables both measuring anxiety, therefore, from the same class. In such cases, an intraclass correlation (ICC) is used and is commonly extended beyond just two variables to look at the consistency between judges. For example, in gymnastics, ice skating, diving and other Olympic sports, contestant’s performance is often assessed by a panel of judges. There might be 10 judges, all of whom rate performance out of 10; therefore, the resulting measures are from the same class (they measure the same thing). The objects being rated are the competitors. This again is a perfect scenario of an intraclass correlation.

Models of Intraclass Correlations

There are a variety of different intraclass correlations (McGraw & Wong, 1996; Shrout & Fleiss, 1979) and the first step in calculating one is to determine a model for your sample data. All of the various forms of the intraclass correlation are based on estimates of mean variability from a one-way repeated-measures *analysis of variance*.

All situations in which an intraclass correlation is desirable will involve multiple measures on different entities (be they twins, Olympic competitors, pictures, sea slugs, etc.). The objects measured constitute a random factor in the design (they are assumed to be random exemplars of the population of objects). The measures taken can be included as factors in the design if they have a meaningful order, or can be excluded if they are unordered as we shall now see.

One-Way Random Effects Model

In the simplest case we might have only two measures (think back to our twin study on anxiety). When the order of these variables is irrelevant (for example, with our twins it is arbitrary whether we treat the data from the first twin as being anxiety measure 1 or anxiety measure 2). In this case, the only systematic source of variation is the random variable representing the different objects. As such, we can use a one-way ANOVA of the form:

$$x_{ij} = \mu + r_i + e_{ij}$$

in which r_i is the effect of object i (known as the row effects), j is the measure being considered, and e_{ij} is an error term (the residual effects). The row and residual effects are random, independent and normally distributed. Because the effect of the measure is ignored, the resulting intraclass correlation is based on the overall effect of the objects being measured (the mean between-object variability MS_{Rows}) and the mean within-object variability (MS_w). Both of these will be formally defined later.

Two-Way Random Effects Model

When the order of measures is important then the effect of the measures becomes important. The most common case of this is when measures come from different judges or raters. Hodgins and Makarchuk (2003), for example, show two such uses; in their study they took multiple measures of the same class of behaviour (gambling) but also took measures from different sources. They measured gambling both in terms of days spent gambling and money spent gambling. Clearly these measures generate different data so it is important to which measure a datum belongs (it is not arbitrary to which measure a datum is assigned). This is one scenario in which a two-way model is used. However, they also took measures of gambling both from the gambler and a collateral (e.g. spouse). Again, it is important that we attribute data to the correct source. So, this is a second illustration of where a two-way model is useful. In such situations the intraclass correlation can be used to check the consistency or agreement between measures or raters.

In this situation a two-way model can be used as follows:

$$x_{ij} = \mu + r_i + c_j + rc_{ij} + e_{ij}$$

in which c_j is the effect of the measure (i.e. the effect of different raters, or different measures), and rc_{ij} is the interaction between the measures taken and the objects being measured. The effect of the measure (c_j) can be treated as either a fixed effect or a random effect. How it is treated doesn't affect the calculation of the intraclass correlation, but it does affect the interpretation (as we shall see). It is also possible to exclude the interaction term and use the model:

$$x_{ij} = \mu + r_i + c_j + e_{ij}$$

We shall now turn our attention to calculating the sources of variance needed to calculate the intraclass correlation.

Sources of Variance: An Example

In the chapter in the book on repeated-measures ANOVA, there is an example relating to student concerns about the consistency of marking between lecturers. It is common that lecturers obtain reputations for being 'hard' or 'light' markers which can lead students to believe that their marks are not based solely on the intrinsic merit of the work, but can be influenced by who marked the work. To test this we could calculate an intraclass correlation. First, we could submit the same eight essays to four different lecturers and record the mark they gave each essay. Table 1 shows the data, and you should note that it looks the same as a one-way repeated-measures ANOVA in which the four lecturers represent four levels of an 'independent variable' and the outcome or dependent variable is the mark given (in fact I use these data as an example of a one-way repeated-measures ANOVA).

Table 1:

Essay	Dr Field	Dr Smith	Dr Scrote	Dr Death	Mean	S^2	$S^2(k-1)$
1	62	58	63	64	61.75	6.92	20.75
2	63	60	68	65	64.00	11.33	34.00
3	65	61	72	65	65.75	20.92	62.75
4	68	64	58	61	62.75	18.25	54.75
5	69	65	54	59	61.75	43.58	130.75
6	71	67	65	50	63.25	84.25	252.75
7	78	66	67	50	65.25	132.92	398.75
8	75	73	75	45	67.00	216.00	648.00
Mean:	68.88	64.25	65.25	57.38	63.94	Total:	1602.50

There are three different sources of variance that are needed to calculate an intraclass correlation which we shall now calculate. These sources of variance are the same as those calculated in one-way repeated-measures ANOVA. (If you don't believe me, consult Smart Alex's answers to Chapter 13 to see an identical set of calculations!).

The Between-Object Variance (MS_{Rows})

The first source of variance is the variance between the objects being rated (in this case the between-essay variance). Essays will naturally vary in their quality for all sorts of reasons (the natural ability of the author, the time spent writing the essay, etc.). This variance is calculated by looking at the average mark for each essay and seeing how much it deviates from the average mark for all essays. These deviations are squared because some will be positive and others negative and so would cancel out when summed. The squared errors for each essay are weighted by the number of values that contribute to the mean (in this case the number of different markers, k). So, in general terms we write this as:

$$SS_{\text{Rows}} = \sum_{i=1}^n k_i (\bar{X}_{\text{Row } i} - \bar{X}_{\text{all rows}})^2$$

Or, for our example, we could write it as:

$$SS_{\text{Essays}} = \sum_{i=1}^n k_i (\bar{X}_{\text{Essay } i} - \bar{X}_{\text{all essays}})^2$$

This would give us:

$$\begin{aligned} SS_{\text{Rows}} &= 4(61.75 - 63.94)^2 + 4(64.00 - 63.94)^2 + 4(65.75 - 63.94)^2 + 4(62.75 - 63.94)^2 + \dots \\ &\quad + 4(61.75 - 63.94)^2 + 4(63.25 - 63.94)^2 + 4(65.25 - 63.94)^2 + 4(67.00 - 63.94)^2 \\ &= 19.18 + 0.014 + 13.10 + 5.66 + 19.18 + 1.90 + 6.86 + 37.45 \\ &= 103.34 \end{aligned}$$

This sum of squares is based on the total variability and so its size depends on how many objects (essays in this case) have been rated. Therefore, we convert this total to an average known as the mean squared error (MS) by dividing by the number of essays (or in general terms the number of rows) minus 1. This value is known as the *degrees of freedom*.

$$MS_{\text{Rows}} = \frac{SS_{\text{Rows}}}{df_{\text{Rows}}} = \frac{103.34}{n-1} = \frac{103.34}{7} = 14.76$$

The mean squared error for the rows in the table is our estimate of the natural variability between the objects being rated.

The Within-Judge Variability (MS_W)

The second variability in which we're interested is the variability within measures/judges. To calculate this we look at the deviation of each judge from the average of all judges on a particular essay. We use an equation with the same structure as before, but for each essay separately:

$$SS_{\text{Essay}} = \sum_{k=1}^p (\bar{X}_{\text{Column } k} - \bar{X}_{\text{all columns}})^2$$

For essay 1, for example, this would be:

$$SS_{\text{Essay}} = (62 - 61.75)^2 + (58 - 61.75)^2 + (63 - 61.75)^2 + (64 - 61.75)^2 = 20.75$$

The degrees of freedom for this calculation are again one less than the number of scores used in the calculation: the number of judges, k , minus 1.

We have to calculate this for each of the essays in turn and then add these values up to get the total variability within judges. An alternative way to do this is to use the variance within each essay. The equation mentioned above is equivalent to the variance for each essay multiplied by the number of values on which that variance is based (in this case the number of judges, k) minus 1. As such we get:

$$SS_W = s_{\text{Essay1}}^2 (k_1 - 1) + s_{\text{Essay2}}^2 (k_2 - 1) + s_{\text{Essay3}}^2 (k_3 - 1) + \dots + s_{\text{Essay } n}^2 (k_n - 1)$$

Table 1 shows the values for each essay in the last column. When we sum these values we get 1602.50. As before, this value is a total and so depends on the number essays (and the number of judges). Therefore, we convert it to an average, by dividing by the degrees of freedom. For each essay we calculated a sum of squares that we saw was based on $k - 1$ degrees of freedom. Therefore, the degrees of freedom for the total within-judge variability are the sum of the degrees of freedom for each essay:

$$df_W = n(k - 1)$$

in which n is the number of essays and k is the number of judges. In this case it will be $8(4 - 1) = 24$. The resulting mean squared error is, therefore:

$$MS_W = \frac{SS_W}{df_W} = \frac{1602.50}{n(k - 1)} = \frac{1602.50}{24} = 66.77$$

The Between-Judge Variability (MS_{Columns})

The within-judge or within-measure variability is made up of two components. The first is the variability created by *differences* between judges. The second is unexplained variability (error, for want of a better word). The variability between judges is again calculated using a variant of the same equation that we've used all along, only this time we're interested in the deviation of each judge's mean from the mean of all judges:

$$SS_{\text{Columns}} = \sum_{k=1}^p n_i (\bar{X}_{\text{Column } i} - \bar{X}_{\text{all columns}})^2$$

or:

$$SS_{\text{Judges}} = \sum_{k=1}^p n_i (\bar{X}_{\text{Judge } i} - \bar{X}_{\text{all judges}})^2$$

in which n is the number of things that each judge rated. For these data we'd get:

$$\begin{aligned} SS_{\text{Columns}} &= 8(68.88 - 63.94)^2 + 8(64.25 - 63.94)^2 + 8(65.25 - 63.94)^2 + 8(57.38 - 63.94)^2 \\ &= 554 \end{aligned}$$

The degrees of freedom for this effect are the number of judges, k , minus 1. As before, the sum of squares is converted to a mean squared error by dividing by the degrees of freedom:

$$MS_{\text{Columns}} = \frac{SS_{\text{Columns}}}{df_{\text{Columns}}} = \frac{554}{k - 1} = \frac{554}{3} = 184.67$$

The Error Variability (MS_E)

The final variability is the variability that can't be explained by known factors such as variability between essays or judges/measures. This can be easily calculated using subtraction because we know that the within-judges variability is made up of the between-judges variability and this error:

$$\begin{aligned} SS_W &= SS_{\text{Columns}} + SS_E \\ SS_E &= SS_W - SS_{\text{Columns}} \end{aligned}$$

The same is true of the degrees of freedom:

$$\begin{aligned} df_W &= df_{\text{Columns}} + df_E \\ df_E &= df_W - df_{\text{Columns}} \end{aligned}$$

So, for these data we get:

$$\begin{aligned}
 SS_E &= SS_W - SS_{\text{Columns}} \\
 &= 1602.50 - 554 \\
 &= 1048.50
 \end{aligned}$$

and

$$\begin{aligned}
 df_E &= df_W - df_{\text{Columns}} \\
 &= 24 - 3 \\
 &= 21
 \end{aligned}$$

We get the average error variance in the usual way:

$$MS_E = \frac{SS_E}{df_E} = \frac{1048.50}{21} = 49.93$$

Calculating Intraclass Correlations

Having computed the necessary variance components, we shall now look at how the intraclass correlation is calculated. Before we do so, however, there are two important decisions to be made.

Single Measures of Average Measures

So far we have talked about situations in which the measures we've used produce single values. However, it is possible that we might have measures that produce an average score. For example, we might get judges to rate paintings in a competition based on style, content, originality, and technical skill. For each judge, their ratings are averaged. The end result is still ratings from a set of judges, but these ratings are an average of many ratings. Intraclass correlations can be computed for such data, but the computation is somewhat different.

Consistency or Agreement?

The next decision involves whether you want a measure of overall consistency between measures/judges. The best way to explain this distinction is to return to our lecturers marking essays. It is possible that particular lecturers are harsh in their ratings (or lenient). A consistency definition views these differences as an irrelevant source of variance. As such the between-judge variability described above (MS_{Columns}) is ignored in the calculation (see Table 2). In ignoring this source of variance we are getting a measure of whether judges agree about the relative merits of the essays without worrying about whether the judges anchor their marks around the same point. So, if all the judges agree that essay 1 is the best, essay 5 is the worst (or their rank order of essays is roughly the same) then agreement will be high: it doesn't matter that Dr Field's marks are all 10% higher than Dr Death's. This is a consistency definition of agreement.

The alternative is to treat relative differences between judges as an important source of disagreement. That is, the between-judge variability described above (MS_{Columns}) is treated as an important source of variation and is included in the calculation (see Table 2). In this scenario disagreements between the relative magnitude of judge's ratings matters (so, the fact that Dr Death's marks differ from Dr Field's will matter even if their rank order of marks is in agreement). This is an absolute agreement definition. By definition the one-way model ignores the effect of the measures and so can have only this kind of interpretation.

Equations for ICCs

Table 2 shows the equations for calculating ICC based on whether a one-way or two-way model is assumed and whether a consistency or absolute agreement definition is preferred. For illustrative purposes, the ICC is calculated in each case for the example used in this entry. This should enable the reader to identify how to calculate the various sources of variance. In this table MS_{Columns} is abbreviated to MS_C and MS_{Rows} is abbreviated to MS_R .

Table 2:

ICC for Single Scores			
Mod	Interpretati	Equation	ICC for example data

el	on		
One-way	Absolute Agreement	$\frac{MS_R - MS_W}{MS_R + (k-1)MS_W}$	$\frac{14.76 - 66.77}{14.76 + (4-1)66.77} = -0.24$
Two-Way	Consistency	$\frac{MS_R - MS_E}{MS_R + (k-1)MS_E}$	$\frac{14.76 - 49.93}{14.76 + (4-1)49.93} = -0.21$
	Absolute Agreement	$\frac{MS_R - MS_E}{MS_R + (k-1)MS_E + \frac{k}{n}(MS_C - MS_E)}$	$\frac{14.76 - 49.93}{14.76 + (4-1)49.93 + \frac{4}{8}(184.67 - 49.93)} = -0.15$
ICC for average scores			
One-way	Absolute Agreement	$\frac{MS_R - MS_W}{MS_R}$	$\frac{14.76 - 66.77}{14.76} = -3.52$
Two-Way	Consistency	$\frac{MS_R - MS_E}{MS_R}$	$\frac{14.76 - 49.93}{14.76} = -2.38$
	Absolute Agreement	$\frac{MS_R - MS_E}{MS_R + \frac{MS_C - MS_E}{n}}$	$\frac{14.76 - 49.93}{14.76 + \frac{184.67 - 49.93}{8}} = -1.11$

Significance Testing

The calculated intraclass correlation can be tested against a value under the null hypothesis using a standard *F*-test (see *analysis of variance*). McGraw and Wong (1996) describe these tests for the various intraclass correlations we've seen, and Table 3 summarizes their work. In this table ICC is the observed intraclass correlation whereas ρ_0 is the value of the intraclass correlation under the null hypothesis. That is, it's the value against which you wish to compare the observed intraclass correlation. So, replace this value with 0 to test the hypothesis that the observed ICC is greater than zero, but replace it with other values such as .1, .3 or .5 to test that the observed ICC is greater than known values of small medium and large effect sizes, respectively.

Table 3:

ICC for Single Scores				
Model	Interpretation	<i>F</i> -ratio	df1	df2
One-way	Absolute Agreement	$\frac{MS_R}{MS_W} \times \frac{1 - \rho_0}{1 + (k-1)\rho_0}$	$n - 1$	$n(k-1)$
Two-Way	Consistency	$\frac{MS_R}{MS_E} \times \frac{1 - \rho_0}{1 + (k-1)\rho_0}$	$n - 1$	$(n-1)(k-1)$
	Absolute Agreement	$\frac{MS_R}{aMS_C + bMS_E}$ in which $a = \frac{k\rho_0}{n(1-\rho_0)}$ $b = 1 + \frac{k\rho_0(n-1)}{n(1-\rho_0)}$	$n - 1$	$\frac{(aMS_C + bMS_E)^2}{\frac{(aMS_C)^2}{k-1} + \frac{(bMS_E)^2}{(n-1)(k-1)}}$
ICC for Average Scores				
One-way	Absolute Agreement	$\frac{1 - \rho_0}{1 - ICC}$	$n - 1$	$n(k-1)$

Two-Way	Consistency	$\frac{1 - \rho_0}{1 - \text{ICC}}$	$n - 1$	$(n - 1)(k - 1)$
	Absolute Agreement	$\frac{MS_R}{cMS_C + dMS_E}$ in which $c = \frac{\rho_0}{n(1 - \rho_0)}$ $b = 1 + \frac{\rho_0(n - 1)}{n(1 - \rho_0)}$	$n - 1$	$\frac{(cMS_C + dMS_E)^2}{\frac{(cMS_C)^2}{k - 1} + \frac{(dMS_E)^2}{(n - 1)(k - 1)}}$

Fixed versus Random Effects

I mentioned earlier on that the effect of the measure/judges can be conceptualized as a fixed or random effect. Although it makes no difference to the calculation, it does affect the interpretation. Essentially, this variable should be regarded as random when the judges or measures represent a sample of a larger population of measures or judges that could have been used. Put another way, the particular judges or measures chosen are not important and do not change the research question you're addressing. However, the effect of measures should be treated as fixed when changing one of the judges or measures would significantly affect the research question (see *fixed and random effects*). For example, in the gambling study mentioned earlier it would make a difference if the ratings of the gambler were replaced: the fact that gamblers gave ratings was intrinsic to the research question being addressed (do gamblers give accurate information about their gambling?). However, in our example of lecturer's marks, it shouldn't make any difference if we substitute one lecturer with a different one: we can still answer the same research question (do lecturers, in general, give inconsistent marks?). In terms of interpretation, when the effect of the measures is a random factor, the results can be generalized beyond the sample; however, when they are a fixed effect, any conclusions apply only to the sample on which the ICC is based (McGraw & Wong, 1996).

References

- Eley, T. C., & Stevenson, J. (1999). Using genetic analyses to clarify the distinction between depressive and anxious symptoms in children. *Journal of Abnormal Child Psychology*, 27(2), 105–114.
- Hodgins, D. C., & Makarchuk, K. (2003). Trusting problem gamblers: Reliability and validity of self-reported gambling behaviour. *Psychology of Addictive Behaviors*, 17(3), 244–248.
- McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods*, 1(1), 30–46.
- Shrout, P. E. F., & Fleiss, J. L. (1979). Intraclass correlations: uses in assessing reliability. *Psychological Bulletin*, 86, 420–428.

Please Sir, can I have some more ... centring?

We'll use the **Cosmetic Surgery.dat** data to illustrate the two types of centring discussed in the book chapter. Load this file into **R**. Let's assume that we want to centre the variable **BDI**.

Grand mean centring



Grand mean centring is really easy since we can simply use the *scale()* function that we encountered in the book. To create a new variable in the *surgeryData* dataframe, we simply access this dataframe:

```
surgeryData = read.delim("Cosmetic Surgery.dat", header = TRUE)
```

and then use the *scale* function to create a new variable in *surgeryData* called

BDI_Centred:

```
surgeryData$BDI_Centred <- scale(surgeryData$BDI, scale = F)
```

Within the *scale()* function itself, *surgeryData\$BDI* just specifies the variable that you'd like to scale (in this case **BDI**), and *scale = F* just tells it not to scale the variable (the F stands for FALSE). Scaling the variable means converting to a standardized score (by dividing by the standard deviation). There is another option, *center = TRUE/FALSE*, which tells **R** whether or not to centre the variable. The default is TRUE, so we don't actually need to specify it; by not specifying it, we are telling **R** to create a new variable based on **BDI** that is centred but not scaled. The end result is a grand mean centred variable. If you look at the data using *print(surgeryData)*, you'll find a variable has been added (**BDI_Centred**), which is **BDI** but grand mean centred.

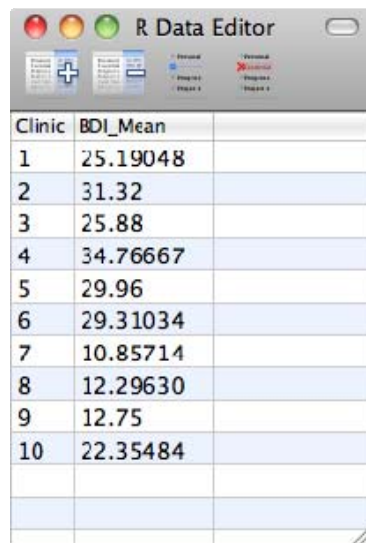
	L	Base_QoL	Clinic	Surgery	Reason	Age	Gender	BDI	Surgery_Text	Reason_Text	Gender_Text	BDI_Cen
1	3	73	1	0	0	31	0	12	Waiting List	Change Appearance	Female	-11.0543
2	0	74	1	0	0	32	0	16	Waiting List	Change Appearance	Female	-7.0543
3	0	80	1	0	0	33	0	13	Waiting List	Change Appearance	Female	-10.0543
4	9	76	1	0	0	59	1	11	Waiting List	Change Appearance	Male	-12.0543
5	0	71	1	0	0	61	1	11	Waiting List	Change Appearance	Male	-12.0543
6	5	72	1	0	1	32	0	10	Waiting List	Physical reason	Female	-13.0543
7	0	71	1	0	1	33	0	11	Waiting List	Physical reason	Female	-12.0543
8	0	73	1	0	1	35	0	15	Waiting List	Physical reason	Female	-8.0543
9	5	80	1	1	0	25	0	30	Cosmetic Surgery	Change Appearance	Female	6.9456
10	0	64	1	0	0	55	1	36	Waiting List	Change Appearance	Male	12.9456
11	0	71	1	0	0	57	1	37	Waiting List	Change Appearance	Male	13.9456
12	9	72	1	0	0	29	0	34	Waiting List	Change Appearance	Female	10.9456
13	0	68	1	0	1	31	0	30	Waiting List	Physical reason	Female	6.9456
14	0	65	1	0	1	32	0	31	Waiting List	Physical reason	Female	7.9456
15	5	66	1	0	0	43	0	41	Waiting List	Change Appearance	Female	17.9456
16	0	76	1	0	1	45	0	34	Waiting List	Physical reason	Female	10.9456
17	0	69	1	0	0	46	0	36	Waiting List	Change Appearance	Female	12.9456
18	1	73	1	1	0	18	0	30	Cosmetic Surgery	Change Appearance	Female	6.9456
19	0	66	1	1	1	19	0	25	Cosmetic Surgery	Physical reason	Female	1.9456
20	0	61	1	1	1	20	0	31	Cosmetic Surgery	Physical reason	Female	7.9456
21	0	66	1	0	0	51	1	35	Waiting List	Change Appearance	Male	11.9456
22	2	70	2	0	1	40	0	27	Waiting List	Physical reason	Female	3.9456
23	0	91	2	0	1	41	1	13	Waiting List	Physical reason	Male	-10.0543
24	0	73	2	0	1	43	1	15	Waiting List	Physical reason	Male	-8.0543

Group mean centring

Group mean centring is a little more complicated, but still pretty easy. We'll again use **BDI** as the variable that we want to centre. If we want to centre this around the group mean for each clinic, then we first need to know what the mean BDI was within each clinic. We do this by creating a new dataframe that contains the mean BDI score for each clinic using the *aggregate()* function. This function works by first specifying the variable that we want to analyse, in this case the **BDI** variable within the *surgeryData* dataframe (*surgeryData\$BDI*), we then specify a variable by which we want to aggregate the data. In this case, we want to aggregate across clinics, so we need to specify the **Clinic** variable from the *surgeryData* dataframe. We have to specify this variable in list form because it's possible to specify more than one variable to do the aggregation. So, if you wanted to aggregate across variables *x*, *y* and *z* you would specify *list(x, y, z)*. In our case, we are aggregating by only one variable, so we specify this single variable but in list form, *list(surgeryData\$Clinic)*. We then specify the summary statistic that we want from the aggregation process, and this is simply the mean so we type *mean*. The next function, *names()*, is used to rename the variables in the new dataframe. The new dataframe will contain two variables: one specifying the levels of **Clinic** (this will be the same as the variable **Clinic** in the *surgeryData* dataframe), and the other containing the mean BDI score for each clinic. Therefore, we name these two variables **Clinic** and **BDI_Mean**. It is important that we gave **Clinic** the same name as the corresponding variable in *surgeryData* because we will use this variable to merge the new dataframe with *surgeryData*.

```
groupMeans<-aggregate(surgeryData$BDI, list(surgeryData$Clinic), mean)
names(groupMeans)<-c("Clinic", "BDI_Mean")
```

To get a feel for what has happened, let's look at the contents of the *groupMeans* dataframe:

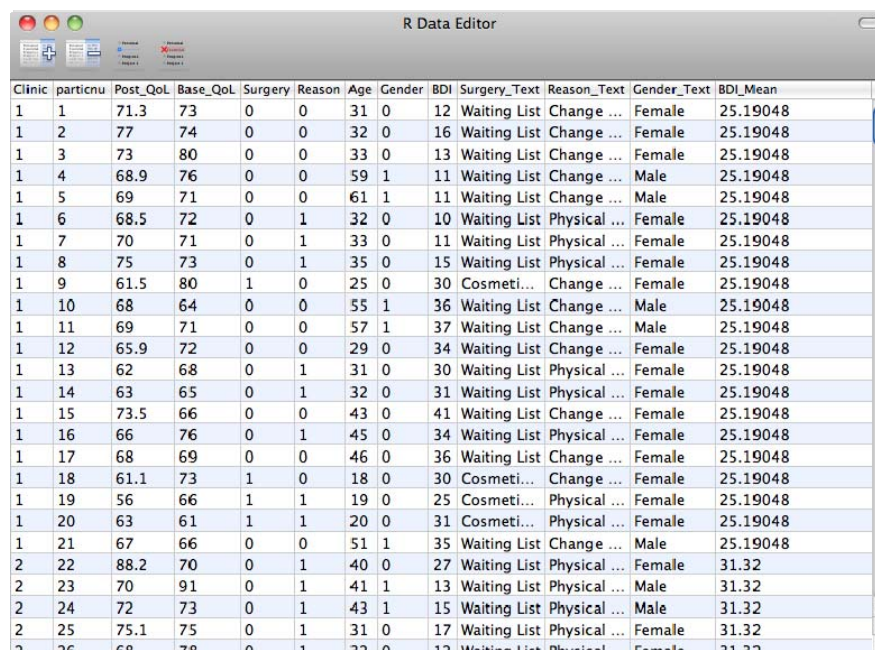


Clinic	BDI_Mean
1	25.19048
2	31.32
3	25.88
4	34.76667
5	29.96
6	29.31034
7	10.85714
8	12.29630
9	12.75
10	22.35484

You can see there are two variables, one defining the clinic (**Clinic**) and the other showing the mean BDI score for that clinic (**BDI_Mean**). The next step is to use these clinic means in the aggregated dataframe to centre the BDI variable in our main dataframe. To do this we need to use the *merge()* function. The *merge()* function takes the form *merge(x, y, by = common variable)* in which *x* and *y* are the two dataframes that you want to merge. In this case we want to merge *surgeryData* with the *groupMeans* dataframe that we have just created. These dataframes have only one variable in common (**Clinic**) so we define this variable in the *by* option. Rather than create a whole new dataframe, we will simply overwrite the existing *surgeryData*.

```
surgeryData<-merge(surgeryData, groupMeans, by = "Clinic")
```

The result is that we recreate *surgeryData* to include the variable **BDI_Mean** from the groupMean dataframe. Again, let's take a quick look to get an idea of what we have done:

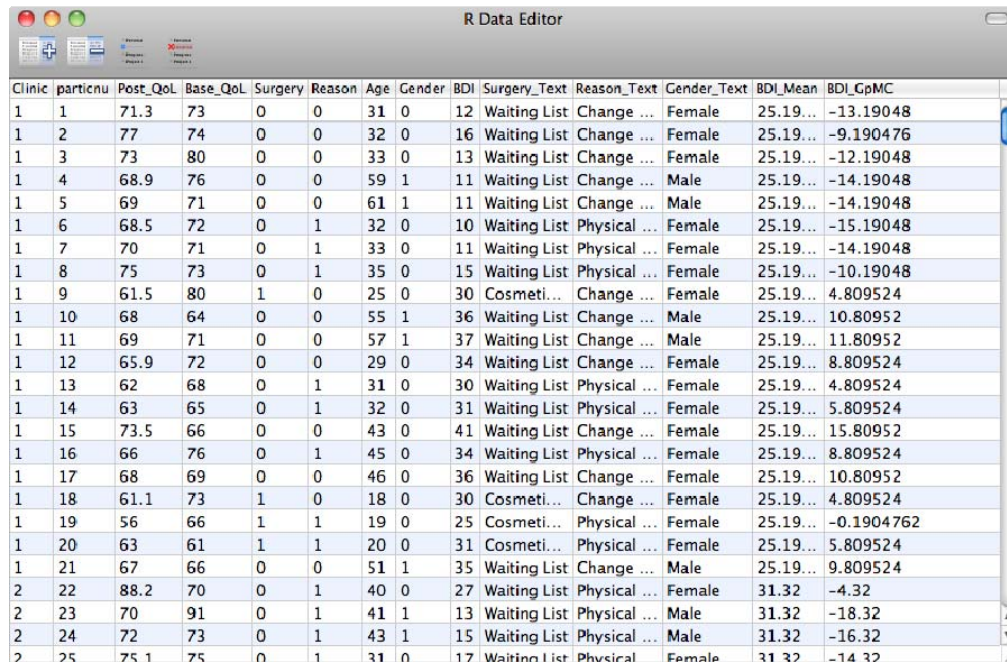


Clinic	particu	Post_QoL	Base_QoL	Surgery	Reason	Age	Gender	BDI	Surgery_Text	Reason_Text	Gender_Text	BDI_Mean
1	1	71.3	73	0	0	31	0	12	Waiting List	Change ...	Female	25.19048
1	2	77	74	0	0	32	0	16	Waiting List	Change ...	Female	25.19048
1	3	73	80	0	0	33	0	13	Waiting List	Change ...	Female	25.19048
1	4	68.9	76	0	0	59	1	11	Waiting List	Change ...	Male	25.19048
1	5	69	71	0	0	61	1	11	Waiting List	Change ...	Male	25.19048
1	6	68.5	72	0	1	32	0	10	Waiting List	Physical ...	Female	25.19048
1	7	70	71	0	1	33	0	11	Waiting List	Physical ...	Female	25.19048
1	8	75	73	0	1	35	0	15	Waiting List	Physical ...	Female	25.19048
1	9	61.5	80	1	0	25	0	30	Cosmeti...	Change ...	Female	25.19048
1	10	68	64	0	0	55	1	36	Waiting List	Change ...	Male	25.19048
1	11	69	71	0	0	57	1	37	Waiting List	Change ...	Male	25.19048
1	12	65.9	72	0	0	29	0	34	Waiting List	Change ...	Female	25.19048
1	13	62	68	0	1	31	0	30	Waiting List	Physical ...	Female	25.19048
1	14	63	65	0	1	32	0	31	Waiting List	Physical ...	Female	25.19048
1	15	73.5	66	0	0	43	0	41	Waiting List	Change ...	Female	25.19048
1	16	66	76	0	1	45	0	34	Waiting List	Physical ...	Female	25.19048
1	17	68	69	0	0	46	0	36	Waiting List	Change ...	Female	25.19048
1	18	61.1	73	1	0	18	0	30	Cosmeti...	Change ...	Female	25.19048
1	19	56	66	1	1	19	0	25	Cosmeti...	Physical ...	Female	25.19048
1	20	63	61	1	1	20	0	31	Cosmeti...	Physical ...	Female	25.19048
1	21	67	66	0	0	51	1	35	Waiting List	Change ...	Male	25.19048
2	22	88.2	70	0	1	40	0	27	Waiting List	Physical ...	Female	31.32
2	23	70	91	0	1	41	1	13	Waiting List	Physical ...	Male	31.32
2	24	72	73	0	1	43	1	15	Waiting List	Physical ...	Male	31.32
2	25	75.1	75	0	1	31	0	17	Waiting List	Physical ...	Female	31.32
2	26	68	78	0	1	22	0	12	Waiting List	Physical ...	Female	31.32

Note that all that has changed is that there is a new variable that for each case of data contains the mean BDI score within the clinic to which a person belonged. The final stage is to do the centring. All we do when we centre is take the actual score (in this case **BDI**) and subtract from it the mean BDI score for the group (**BDI_Mean**). We, therefore, create a new variable in the *surgeryData* dataframe called **BDI_GpMC** that is simply the BDI score minus the group BDI score:

```
surgeryData$BDI_GpMC<-surgeryData$BDI-surgeryData$BDI_Mean
```

We can again look at how *surgeryData* has changed:



Clinic	particu	Post_QoL	Base_QoL	Surgery	Reason	Age	Gender	BDI	Surgery_Text	Reason_Text	Gender_Text	BDI_Mean	BDI_GpMC
1	1	71.3	73	0	0	31	0	12	Waiting List	Change ...	Female	25.19...	-13.19048
1	2	77	74	0	0	32	0	16	Waiting List	Change ...	Female	25.19...	-9.190476
1	3	73	80	0	0	33	0	13	Waiting List	Change ...	Female	25.19...	-12.19048
1	4	68.9	76	0	0	59	1	11	Waiting List	Change ...	Male	25.19...	-14.19048
1	5	69	71	0	0	61	1	11	Waiting List	Change ...	Male	25.19...	-14.19048
1	6	68.5	72	0	1	32	0	10	Waiting List	Physical ...	Female	25.19...	-15.19048
1	7	70	71	0	1	33	0	11	Waiting List	Physical ...	Female	25.19...	-14.19048
1	8	75	73	0	1	35	0	15	Waiting List	Physical ...	Female	25.19...	-10.19048
1	9	61.5	80	1	0	25	0	30	Cosmeti...	Change ...	Female	25.19...	4.809524
1	10	68	64	0	0	55	1	36	Waiting List	Change ...	Male	25.19...	10.80952
1	11	69	71	0	0	57	1	37	Waiting List	Change ...	Male	25.19...	11.80952
1	12	65.9	72	0	0	29	0	34	Waiting List	Change ...	Female	25.19...	8.809524
1	13	62	68	0	1	31	0	30	Waiting List	Physical ...	Female	25.19...	4.809524
1	14	63	65	0	1	32	0	31	Waiting List	Physical ...	Female	25.19...	5.809524
1	15	73.5	66	0	0	43	0	41	Waiting List	Change ...	Female	25.19...	15.80952
1	16	66	76	0	1	45	0	34	Waiting List	Physical ...	Female	25.19...	8.809524
1	17	68	69	0	0	46	0	36	Waiting List	Change ...	Female	25.19...	10.80952
1	18	61.1	73	1	0	18	0	30	Cosmeti...	Change ...	Female	25.19...	4.809524
1	19	56	66	1	1	19	0	25	Cosmeti...	Physical ...	Female	25.19...	-0.1904762
1	20	63	61	1	1	20	0	31	Cosmeti...	Physical ...	Female	25.19...	5.809524
1	21	67	66	0	0	51	1	35	Waiting List	Change ...	Male	25.19...	9.809524
2	22	88.2	70	0	1	40	0	27	Waiting List	Physical ...	Female	31.32	-4.32
2	23	70	91	0	1	41	1	13	Waiting List	Physical ...	Male	31.32	-18.32
2	24	72	73	0	1	43	1	15	Waiting List	Physical ...	Male	31.32	-16.32
2	25	75.1	75	0	1	31	0	17	Waiting List	Physical ...	Female	31.32	-14.32

We have created a new variable (**BDI_GpMC**) which is the group mean centred BDI score; for example, for participant 1, the value of **BDI_GpMC** is $12 - 25.19 = -13.19$ – in other words, the BDI score minus the group mean of BDI for the clinic.

Labcoat Leni's real research

A fertile gesture

Problem

Miller, G., Tybur, J. M., & Jordan, D. B. (2007). *Evolution and Human Behavior*, 28, 375–381.



Most female mammals experience a phase of 'estrus' during which they are more sexually receptive, proceptive, selective and attractive. As such, the evolutionary benefit to this phase is believed to be to attract mates of superior genetic stock.

However, some people have argued that this important phase became uniquely lost or hidden in human females. Testing these evolutionary ideas is exceptionally difficult, but Geoffrey Miller and his colleagues came up with an incredibly elegant piece of research that did just that. They reasoned that if the 'hidden-estrus' theory is incorrect then men should find women most attractive during the fertile phase of their menstrual cycle compared to the pre-fertile (menstrual) and post-fertile (luteal) phase.

To measure how attractive men found women in an ecologically valid way, they came up with the ingenious idea of collecting data from women working at lap-dancing clubs. These women maximize their tips from male visitors by attracting more dances. In effect the men ‘try out’ several dancers before choosing a dancer for a prolonged dance. For each dance the male pays a ‘tip’. Therefore, the greater the number of men choosing a particular woman, the more her earnings will be. As such, each dancer’s earnings are a good index of how attractive the male customers have found her. Miller and his colleagues argued, therefore, that if women do have an estrus phase then they will be more attractive during this phase and therefore earn more money. This study is a brilliant example of using a real-world phenomenon to address an important scientific question in an ecologically valid way.

The data for this study are in the file **Miller et al. (2007).dat**. The researchers collected data via a website from several dancers (**ID**), who provided data for multiple lap-dancing shifts (so for each person there are several rows of data). They also measured what phase of the menstrual cycle the women were in at a given shift (**Cyclephase**), and whether they were using hormonal contraceptives (**Contraceptive**) because this would affect their cycle. The outcome was their earnings on a given shift in dollars (**Tips**).

A multilevel model can be used here because the data are unbalanced: the women differed in the number of shifts they provided data for (the range was 9 to 29 shifts); multilevel models can handle this problem.

Labcoat Leni wants you to carry out a multilevel model to see whether **Tips** can be predicted from **Cyclephase**, **Contraceptive** and their interaction. Is the ‘estrus-hidden’ hypothesis supported? Answers are in the additional material on the companion website (or look at page 378 in the original article).

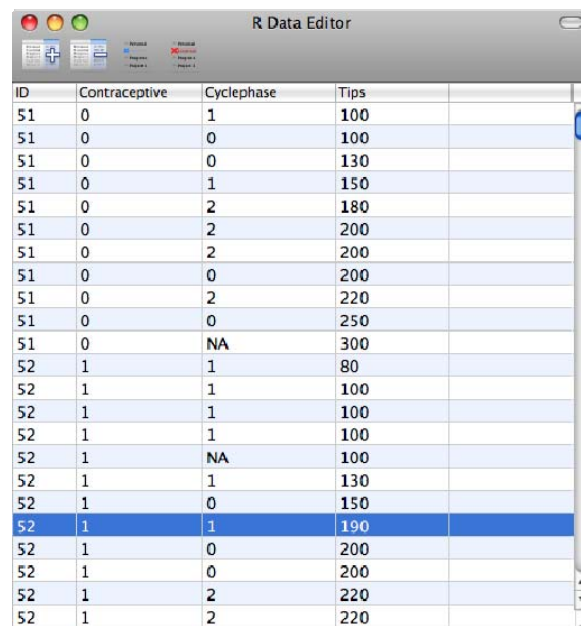
Solution

In the model that Miller et al. fitted, they did not assume that there would be random slopes (i.e. the relationship between each predictor and tips was not assumed to vary within lap dancers). This decision is appropriate for **Contraceptive** because this variable didn’t vary at level 2 (the lap dancer was either taking contraceptives or not, so this could not be set up as a random effect because it doesn’t vary over our level 2 variable of participant). Also, because **Cyclephase** is a categorical variable with three unordered categories we could not expect a linear relationship with tips: we expect tips to vary over categories but the categories themselves have no meaningful order. However, we might expect tips to vary over participants (some lap dancers will naturally get more money than others) and we can factor this variability in by allowing the intercept to be random. As such, we’ll fit a random intercept model to the data.

First we need to get the data. Assuming you have set your working directory to be the folder in which the data are stored, you can use:

```
dancerData = read.delim("Miller et al. (2007).dat", header = TRUE)
```

to create a dataframe called *dancerData*. The data look like this:



ID	Contraceptive	Cyclephase	Tips
S1	0	1	100
S1	0	0	100
S1	0	0	130
S1	0	1	150
S1	0	2	180
S1	0	2	200
S1	0	2	200
S1	0	0	200
S1	0	2	220
S1	0	0	250
S1	0	NA	300
S2	1	1	80
S2	1	1	100
S2	1	1	100
S2	1	1	100
S2	1	NA	100
S2	1	1	130
S2	1	0	150
S2	1	1	190
S2	1	0	200
S2	1	0	200
S2	1	2	220
S2	1	2	220

Note that contraceptive use is stored in a variable that codes 0 = on the pill, 1 = in natural cycle. Cyclephase is codes with numbers 0 = Luteal, 1 = Menstrual, 2 = Fertile. Ideally, we want to code **Cyclephase** in a way that would be most useful for interpretation. **R** will compare each category against the first, so we need the first category to be a meaningful control. The group of interest is the fertile period, so we need this to be coded as the first category (0), to get the contrasts we want. Unfortunately, it's currently coded as the last, so we need to turn this variable into a factor and recode it. This is easily done using the `recode()` function in the `car()` package. If you don't have `car()` installed then use the `install.packages("car")` to install it.

```
library(car)
```

```
Cyclephase_Factor <- factor(car::recode(dancerData$Cyclephase, "2=0;0=2"), levels = 0:2, labels = c("Fertile", "Menstrual", "Luteal"))
```

These commands first initialize the `car()` package. Next we turn **Cyclephase** into a factor called **Cyclephase_Factor** using the `factor()` function. The first part of the function specifies the variable, which in this case is the **Cyclephase** variable in the `dancerData` dataframe (`dancerData$Cyclephase`). However, we have been sneaky and sneaked in a `recode` function at the same time. The `car::recode(dancerData$Cyclephase, "2=0;0=2")` simply means 'recode the `dancerData$Cyclephase` variable so that 2 becomes equal to 0, and 0 becomes equal to 2'. In other words, we're switching the coding of 0 and 2 so that the fertile phase, which was coded as 2, will now be coded as 0, and the luteal phase, which was coded as 0, is now coded as 2. The `car::recode` at the beginning is just because there are more than one function called 'recode' and this disambiguates the situation by telling **R** that we want to use the `recode` function from the `car` package. The rest of the function determines how the factor is created: `levels = 0:2` tells the `factor()` function that the factor will have three levels coded 0 to 2 inclusive (i.e. 0, 1, 2), and `labels = c("Fertile", "Menstrual", "Luteal")` gives these levels names specified in the order given. In other words, we have created a factor called **Cyclephase_Factor** in which 0 = Fertile, 1 = Menstrual, and 2 = Luteal.

I'm also going to recode the **Contraceptive** variable in the same way. We don't need to do this, but it will keep the output consistent with the SPSS version of the book, so we'll do it in case anyone is comparing the two!

```
Contraceptive_Factor <- factor(car::recode(dancerData$Contraceptive, "0=1;1=0"), levels = 0:1, labels = c("In Natural Cycle", "Using Pill"))
```

Having constructed factors for our predictor variables we can start to build the model. First we need to see whether it makes a difference if let the intercepts vary. First, we create a baseline model:

```
intercept <- gls(Tips~1, data = dancerData, method = "ML", na.action = na.exclude)
```

Now we can fit the model with random intercepts. In this example, multiple scores or shifts are nested within each dancer. Therefore, the level 2 variable is the participant (the lap dancer), and this variable is represented by the variable labelled **ID**. To allow intercepts to vary over strippers (**ID**), we therefore specify the random part of the model as `~1|ID`.

```
randomInt<-lme(Tips~1, random = ~1|ID, data = dancerData, method = "ML", na.action=
na.exclude)
anova(intercept, randomInt)
```

The `anova()` function compares these models and yield the following output:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
intercept	1	2	3702.974	3710.355	-1849.487			
randomInt	2	3	3649.519	3660.591	-1821.760	1 vs 2	55.45485	<.0001

This shows that allowing intercepts to vary across dancers significantly improved the model – in other words, intercepts varied significantly, $\chi^2(1) = 55.45$, $p < .0001$.

We can now add our fixed effects of **Cyclephase_Factor**, **Contraceptive_Factor**, and their interaction. Let's add **Cyclephase_Factor** first:

```
cycleModel<-update(randomInt, .~. + Cyclephase_Factor, method = "REML")
```

Note that I have also switched to REML estimation. This is because I needed to use ML to compare the first two models, but the authors used REML in the analysis they report, so now I'm not going to compare models I have switched to be consistent with what they report. We can then update the last model to add **Contraceptive_Factor**:

```
pillModel<-update(cycleModel, .~. + Contraceptive_Factor)
```

Finally, we update the last model to include the interaction term:¹

```
finalModel<-update(pillModel, .~. + Cyclephase_Factor:Contraceptive_Factor)
```

We can get some *F*-tests for the fixed effects using:

```
anova(finalModel)
```

	numDF	denDF	F-value	p-value
(Intercept)	1	233	258.87588	<.0001
Cyclephase_Factor	2	233	48.76266	<.0001
Contraceptive_Factor	1	16	8.25671	0.0110
Cyclephase_Factor:Contraceptive_Factor	2	233	5.31945	0.0055

This output tells us our fixed effects. As you can see, they are all significant. Miller and colleagues reported these results as follows: 'Main effects of cycle phase [$F(2, 236)=27.46$, $p < .001$] and contraception use [$F(1, 17)=6.76$, $p < .05$] were moderated by an interaction between cycle phase and pill use [$F(2, 236)=5.32$, $p < .01$]' (p. 378). Their degrees of freedom and *F*-values differ a bit from ours because they used SPSS rather than R, but the basic results are the same (you can compare with the SPSS output below (basically the *p*-values are the same):

¹ We could have added all of the fixed effects in a single step by using:

```
finalModel<-update(randomInt, .~. + Cyclephase_Factor*Contraceptive_Factor)
```

The `Cyclephase_Factor*Contraceptive_Factor` will add not just the interaction but also lower-order main effects. The reason I didn't do this is simply to keep with the general ethos of building up multilevel models bit-by-bit.

Type III Tests of Fixed Effects^a

Source	Numerator df	Denominator df	F	Sig.
Intercept	1	16.673	198.395	.000
Contraceptive	1	16.673	6.756	.019
Cyclephase	2	235.940	27.461	.000
Contraceptive * Cyclephase	2	235.940	5.319	.005

a. Dependent Variable: Tips earned (US dollars per shift).

Basically this shows that the phase of the dancer's cycle significantly predicted tip income, and this interacted with whether or not the dancer was having natural cycles or was on the contraceptive pill. However, we don't know which groups differed.

We can use the main model parameter estimates to tell us this:

```
summary(finalModel)
```

The output is shown below (for space reasons I've edited down the variable names):

```
Linear mixed-effects model fit by REML
Data: stripperData
      AIC      BIC    logLik
3031.277 3059.416 -1507.638

Random effects:
Formula: ~1 | ID
(Intercept) Residual
StdDev:      59.75883 92.90348

Fixed effects: Tips ~ Cyclephase_Factor + Contraceptive_Factor +
Cyclephase_Factor:Contraceptive_Factor
              Value Std.Error DF   t-value p-value
(Intercept)  356.6538  21.10812 233  16.896525  0.0000
Cyc_Fac[T.Menstrual] -170.8562  17.36623 233  -9.838414  0.0000
Cyc_Fac[T.Luteal]   -100.4089  16.42234 233  -6.114161  0.0000
Cont_Fac[T.Using Pill] -141.6240  35.29920 16  -4.012103  0.0010
Cyc_Fac [T.Menstrual]:Cont_Fac[T.Using Pill]  89.9365  34.14396 233  2.634038  0.0090
Cyc_Fac [T.Luteal]: Cont_Fac[T.Using Pill]  86.0861  30.05306 233  2.864471  0.0046
Correlation:
              (Intr) C_F[T.M
```

Remember that I coded **Cyclephase_Factor** in a way that would be most useful for interpretation, which was to code the group of interest (fertile period) as the first category (0), and the other phases as 1 (Menstrual) and 2 (Luteal). The parameter estimates for this variable, therefore, compare each category against the first category, and because I made the first category the fertile phase this means we get a comparison of the fertile phase against the other two. Therefore, we could say (because the *b* is negative) that tips were significantly higher in the fertile phase than in the luteal phase, $b = -100.41$, $t(233) = -6.11$, $p < .001$, and in the menstrual phase, $b = -170.86$, $t(233) = -9.84$, $p < .001$. The beta, as in regression, tells us the change in tips as we shift from one group to another, so during the fertile phase dancers earned about \$100 more than during the luteal phase, and \$170 more than the menstrual phase.

These effects don't factor in the contraceptive use. To look at this we need to look at the contrasts for the interaction term. The first of these tells us the following: if we worked out the relative difference in tips between the fertile phase and the luteal phase, how much more do those in their natural cycle earn compared to those on contraceptive pills? The answer is about \$86. In other words, there is a combined effect of being in a natural cycle (relative to being on the pill) and being in the fertile phase (relative to the luteal phase) and this is significant, $b = 86.09$, $t(233) = 2.86$, $p < .01$. The second contrast tells us the following: if we worked out the relative difference in tips between the fertile phase and the menstrual phase, how much more do those in their natural cycle earn compared to those on contraceptive pills? The answer is about \$90 (the *b*). In other words, there is a combined effect of being in a natural cycle and being in the fertile phase compared to the menstrual phase and this is significant, $b = 89.94$, $t(233) = 2.63$, $p < .01$.

To conclude then, this study showed that the 'estrus-hidden' hypothesis is wrong: men did find women more attractive (as indexed by how many lap dances they did and therefore how much they earned) during the fertile phase of their cycle compared to the other phases.

Smart Alex's solutions

Task 1

- Using the cosmetic surgery example, run the analysis but also including **BDI**, **age** and **gender** as fixed effect predictors. What differences does including these predictors make?

Let's assume you were starting from scratch. We'll quickly build up the same models in the book chapter but using the *update()* function to speed things up:

```
surgeryData = read.delim("Cosmetic Surgery.dat", header = TRUE)
intercept <- gls(Post_QoL~1, data = surgeryData, method = "ML")
randomIntercept <- lme(Post_QoL~1, data = surgeryData, random = ~1|Clinic, method = "ML")
randomInterceptSurgery <- update(randomIntercept, .~. + Surgery)
randomInterceptSurgeryQoL <- update(randomInterceptSurgery, .~. + Base_QoL)
addRandomSlope <- update(randomInterceptSurgeryQoL, random = ~Surgery|Clinic)
addReason <- update(addRandomSlope, .~. + Reason)
finalModel <- update(addReason, .~. + Reason:Surgery)
```

This gets us to the final model in the book chapter (I have kept the model names consistent with the book chapter so you can check back and get a feel for how I built these models. To create a new model that adds in **BDI**, **Age** and **Gender** we can simply update this final model to include these new predictors. In keeping with the ethos of building multilevel models up a step at a time, we can add each one in separately, creating three new models:

```
BDIModel <- update(finalModel, .~. + BDI)
AgeModel <- update(BDIModel, .~. + Age)
genderModel <- update(AgeModel, .~. + Gender)
```

Or we could skip to the last model and create it by adding in the three new predictors in a single command:

```
genderModel <- update(AgeModel, .~. BDI + Age + Gender)
```

We can compare this new model (*genderModel*) to the final model in the book (*finalModel*) to see whether adding these predictors creates a better-fitting model, and also ask for a summary of the model and confidence intervals:

```
anova(finalModel, genderModel); summary(genderModel); intervals(genderModel)
```

The output that compares the models is as follows:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
finalModel	1	9	1807.045	1839.629	-894.5226			
genderModel	2	12	1749.385	1792.830	-862.6927	1 vs 2	63.65979	<.0001

Adding in these new predictors does result in a significantly better fitting model, $\chi^2(3) = 63.66$, $p < .0001$. Note that we have added three new things, so our degrees of freedom are 3 (you can get this from the output as the change in the degrees of freedom ($12 - 9 = 3$)).

The model summary is as follows:

```
Linear mixed-effects model fit by maximum likelihood
Data: surgeryData
      AIC      BIC    logLik
1749.385 1792.830 -862.6927

Random effects:
Formula: ~Surgery | Clinic
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev   Corr
(Intercept) 4.546213 (Intr)
Surgery      1.437201 -0.829
Residual     5.216628
```

Fixed effects: Post_QoL ~ Surgery + Base_QoL + Reason + BDI + Age + Gender + Surgery:Reason

	Value	Std.Error	DF	t-value	p-value
(Intercept)	29.753678	3.705127	259	8.030408	0.0000
Surgery	-3.995717	1.252947	259	-3.189056	0.0016
Base_QoL	0.225128	0.050481	259	4.459693	0.0000
Reason	1.403844	1.338861	259	1.048536	0.2954
BDI	0.184942	0.045879	259	4.031092	0.0001
Age	0.287955	0.047834	259	6.019925	0.0000
Gender	-1.072723	1.146479	259	-0.935668	0.3503
Surgery:Reason	5.021195	1.482800	259	3.386292	0.0008

Correlation:

	(Intr)	Surgry	Bas_QL	Reason	BDI	Age	Gender
Surgery	-0.167						
Base_QoL	-0.739	-0.089					
Reason	-0.454	0.340	0.072				
BDI	-0.337	0.063	0.059	0.619			
Age	-0.010	-0.052	-0.300	-0.058	-0.429		
Gender	0.034	0.097	-0.018	0.144	0.509	-0.637	
Surgery:Reason	0.117	-0.737	0.030	-0.462	-0.038	-0.033	-0.175

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.33705717	-0.74958976	-0.09782865	0.68673398	2.96495854

Number of Observations: 276
Number of Groups: 10

Age, $b = 0.29$, $t(259) = 6.02$, $p < .001$, and BDI, $b = 0.18$, $t(259) = 4.03$, $p < .001$, significantly predicted quality of life after surgery but gender did not, $b = -1.07$, $t(259) = -0.94$, $p = .35$. The main difference that including these factors has made is that the main effect of **Reason** has become non-significant, and the **Reason** \times **Surgery** interaction has become more significant (its b has changed from 4.22, $p = .013$, to 5.02, $p < .001$).

We could break down this interaction as we did in the chapter by creating variables that select out the physical and cosmetic groups, and then update the model containing only **Surgery** and **Base_QoL** (which was the model called *addRandomSlope*) to include **BDI**, **Age** and **Gender**.

```
physicalSubset<-surgeryData$Reason==1
cosmeticSubset<-surgeryData$Reason==0

physicalModel<-update(addRandomSlope, .~. + BDI + Age + Gender, subset=
physicalSubset)

cosmeticModel<-update(addRandomSlope, .~. + BDI + Age + Gender, subset=
cosmeticSubset)

summary(physicalModel)
summary(cosmeticModel)
```

If you do these analyses you will get the parameter tables below. These tables show a similar pattern to the example in the book. For those operated on only to change their appearance, surgery significantly predicted quality of life after surgery, $b = -3.16$, $t(84) = -2.55$, $p = .01$. Unlike when age, gender and BDI were not included, this effect is now significant. The negative gradient shows that in these people quality of life was lower after surgery compared to the control group. However, for those who had surgery to solve a physical problem surgery did not significantly predict quality of life, $b = 0.67$, $t(163) = 0.57$, $p = .57$. In essence the inclusion of age, gender and BDI has made very little difference in this latter group. However, the slope was positive, indicating that people who had surgery scored higher on quality of life than those on the waiting list (although not significantly so!). The interaction effect, therefore, as in the chapter, reflects the difference in slopes for surgery as a predictor of quality of life in those who had surgery for physical problems (slight positive slope) and those who had surgery purely for vanity (a negative slope).

Surgery to change appearance

Linear mixed-effects model fit by maximum likelihood

```
Data: surgeryData
Subset: cosmeticSubset
      AIC      BIC    logLik
573.6497 599.4993 -276.8248
```

Random effects:

```
Formula: ~Surgery | Clinic
Structure: General positive-definite, Log-Cholesky parametrization
```

```

              StdDev   Corr
(Intercept)  5.766448 (Intr)
Surgery      2.514847 -0.772
Residual     3.495342

Fixed effects: Post_QoL ~ Surgery + Base_QoL + BDI + Age + Gender
              Value Std.Error DF   t-value p-value
(Intercept)  28.394900  4.225896  84   6.719262  0.0000
Surgery      -3.163418  1.240321  84  -2.550484  0.0126
Base_QoL      0.147063  0.055898  84   2.630929  0.0101
BDI           0.472555  0.059681  84   7.918083  0.0000
Age           0.198532  0.060153  84   3.300455  0.0014
Gender       -4.696966  1.523293  84  -3.083429  0.0028

```

Surgery for a physical problem

```

Linear mixed-effects model fit by maximum likelihood
Data: surgeryData
Subset: physicalSubset
      AIC      BIC    logLik
1154.884 1186.702 -567.4421

Random effects:
Formula: ~Surgery | Clinic
Structure: General positive-definite, Log-Cholesky parametrization
              StdDev   Corr
(Intercept)  4.679270 (Intr)
Surgery      1.668345 -0.794
Residual     5.471991

Fixed effects: Post_QoL ~ Surgery + Base_QoL + BDI + Age + Gender
              Value Std.Error DF   t-value p-value
(Intercept)  29.893045  4.411360 163   6.776379  0.0000
Surgery      0.666083  1.162912 163   0.572772  0.5676
Base_QoL      0.265651  0.069800 163   3.805899  0.0002
BDI           0.118640  0.064379 163   1.842838  0.0672
Age           0.274836  0.066637 163   4.124374  0.0001
Gender       -0.460955  1.502015 163  -0.306891  0.7593

```

Task 2

- Using our growth model example in this chapter, analyse the data but include **Gender** as an additional covariate. Does this change your conclusions?

Let's recap how we built up the model in the book chapter. First, we read the data into a dataframe and then restructure it so that it is in the correct format.

```

satisfactionData = read.delim("Honeymoon_Period.dat", header = TRUE)

restructuredData<-reshape(satisfactionData, idvar = c("Person", "Gender"), varying =
c("Satisfaction_Base", "Satisfaction_6_Months", "Satisfaction_12_Months",
"Satisfaction_18_Months"), v.names = "Life_Satisfaction", timevar = "Time", times =
c(0:3), direction = "long")

```

We then create a baseline model, then a model with random intercepts:

```

intercept <-glsl(Life_Satisfaction~1, data = restructuredData, method = "ML", na.action
= na.exclude)

randomIntercept <-lme(Life_Satisfaction ~1, data = restructuredData, random =
~1|Person, method = "ML", na.action = na.exclude, control = list(opt="optim"))

```

We then added **Time** as a fixed factor:

```
timeRI<-update(randomIntercept, ~. + Time)
```

We then added a random effect of **Time** across people:

```
timeRS<-update(timeRI, random = ~Time|Person)
```

We then added an AR(1) covariance structure:

```
ARModel<-update(timeRS, correlation = corAR1(0, form = ~Time|Person))
```

We then added a quadratic trend ...

```
timeQuadratic<-update(ARModel, .~. + I(Time^2))
```

... and a cubic trend:

```
timeCubic <-update(timeQuadratic, .~. + I(Time^3))
```

To add the effect of **Gender** we can simply update this final model to include a fixed effect of gender, then compare it to the previous model, and get a summary of it:

```
genderModel <-update(timeCubic, .~. + Gender)
```

```
anova(timeCubic, genderModel)
```

```
summary(genderModel)
```

```
intervals(genderModel)
```

The output that compares the models is as follows:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
timeCubic	1	9	1816.161	1852.901	-899.0808			
genderModel	2	10	1818.051	1858.874	-899.0257	1 vs 2	0.1101303	0.74

Adding in the effect of **Gender** does not result in a significantly better-fitting model, $\chi^2(1) = 0.11$, $p = .74$. The model summary is as follows:

```
Linear mixed-effects model fit by maximum likelihood
Data: restructuredData
      AIC      BIC    logLik
1818.051 1858.874 -899.0257

Random effects:
Formula: ~Time | Person
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev   Corr
(Intercept) 1.8854932 (Intr)
Time         0.4056477 -0.352
Residual     1.4569475

Correlation Structure: AR(1)
Formula: ~Time | Person
Parameter estimate(s):
      Phi
0.1323070
Fixed effects: Life Satisfaction ~ Time + I(Time^2) + I(Time^3) + Gender
              Value Std.Error DF   t-value p-value
(Intercept)  6.694935  0.2878553 320  23.257989  0.0000
Time         1.546318  0.4778063 320   3.236285  0.0013
I(Time^2)    -1.326026  0.4214584 320  -3.146280  0.0018
I(Time^3)     0.171013  0.0930443 320   1.837975  0.0670
Gender       -0.121360  0.3660524 113  -0.331538  0.7409

Correlation:
      (Intr) Time   I(T^2) I(T^3)
Time      -0.218
I(Time^2)  0.111 -0.951
I(Time^3) -0.080  0.896 -0.987
Gender    -0.630  0.003 -0.005  0.006

Standardized Within-Group Residuals:
      Min       Q1       Med       Q3      Max
-2.57843022 -0.55055690 -0.03672578  0.50911575  2.77705706

Number of Observations: 438
Number of Groups: 115
```

The fixed effects and the parameter estimates tell us that the linear, $b = 1.54$, $t(320) = 3.24$, $p < .01$, and quadratic, $b = -1.33$, $t(320) = -3.15$, $p < .01$, trends both significantly described the pattern of the data over time; however, the cubic trend, $b = 0.17$, $t(320) = 1.84$, $p > .05$ does not. These results are basically the same as in the chapter. Gender itself is also not significant in this table (note the p -value is the same as for the log-likelihood test), $b = -0.12$, $t(113) = -0.33$, $p = .74$.

Approximate 95% confidence intervals

```
Fixed effects:
      lower      est.      upper
(Intercept) 6.13184889 6.6949351 7.2580213
Time        0.61165990 1.5463175 2.4809751
I(Time^2)   -2.15045923 -1.3260262 -0.5015931
I(Time^3)   -0.01099486  0.1710130  0.3530209
Gender      -0.84242491 -0.1213602  0.5997044
attr(,"label")
```

```
[1] "Fixed effects:"

Random Effects:
Level: Person
              lower      est.      upper
sd((Intercept)) 1.4834804 1.8854932 2.39644858
sd(Time)         0.1689522 0.4056477 0.97394441
cor((Intercept),Time) -0.6774066 -0.3523010 0.08794113

Correlation structure:
              lower      est.      upper
Phi -0.1913577 0.1323070 0.4300194
attr(,"label")
[1] "Correlation structure:"

Within-group standard error:
              lower      est.      upper
1.168776 1.456947 1.816170

F(1, 113.02) = 0.11, p > .05.
```

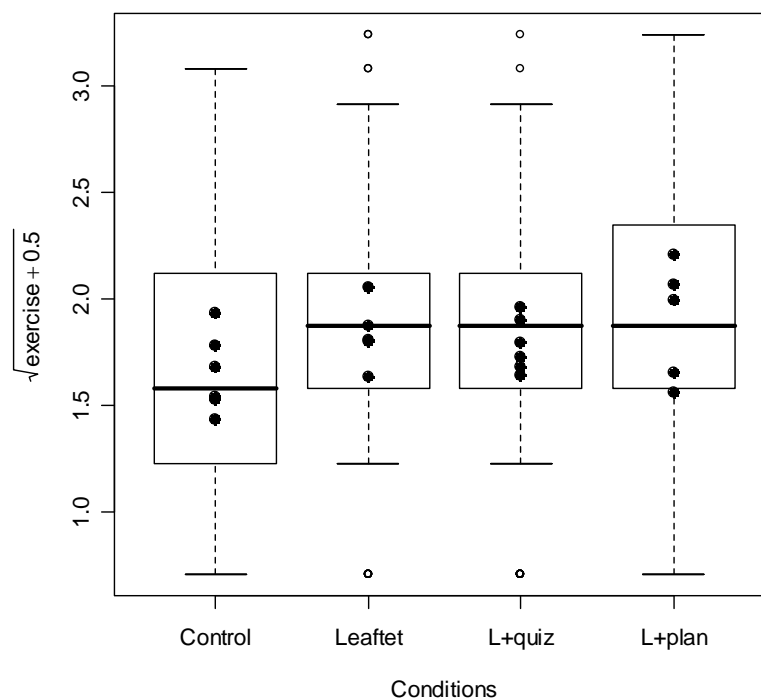
The final part of the output tells us about the random parameters in the model. First of all, the standard deviation of the random intercepts was 1.89 (95% CI: 1.48, 2.40). The confidence interval does not cross zero, which suggests that we were correct to assume that life satisfaction at baseline varied significantly across people. Also, the people's slopes varied significantly, 0.41 (0.17, 0.97). This suggests also that the change in life satisfaction over time varied significantly across people too. Finally, the correlation between the slopes and intercepts (−0.35) suggests that as intercepts increased, the slope decreased, but the confidence interval suggests that this pattern is not significant because it crosses zero (−0.68, 0.09).

These results confirm what we already know from the chapter. The trend in the data is best described by a second-order polynomial, or a quadratic trend. This reflects the initial increase in life satisfaction 6 months after finding a new partner, but a subsequent reduction in life satisfaction at 12 and 18 months after the start of the relationship. The parameter estimates tell us much the same thing. As such our conclusions have been unaffected by including gender.

Task 3

- Getting kids to exercise (Hill, Abraham, & Wright, 2007): The purpose of this research was to examine whether providing children with a leaflet based on the 'theory of planned behaviour' increases children's exercise. There were four different interventions (**Intervention**): a control group, a leaflet, a leaflet and quiz, and a leaflet and plan. A total of 503 children from 22 different classrooms were sampled (**Classroom**). It was not practical to have children in the same classrooms in different conditions, therefore the 22 classrooms were randomly assigned to the four different conditions. Children were asked 'On average over the last three weeks, I have exercised energetically for at least 30 minutes _____ times per week' after the intervention (**Post_Exercise**). Run a multilevel model analysis on these data (**Hill et al. (2007).dat**) to see whether the intervention affected the children's exercise levels (the hierarchy in the data is: children within classrooms within interventions).

Here is a graph of the data; the big dots are means for the schools, the boxplots are standard ignoring the structure.



Let's assume that you have set up your working directory to be the folder in which the data file is stored. We can read these data into a dataframe called *exerciseData* as follows:

```
exerciseData = read.delim("Hill et al. (2007).dat", header = TRUE)
```

The data file looks like the image: note that the intervention group is stored as text rather than numbers (control, leaflet, leaflet + quiz, leaflet + plan). R should convert this variable to a factor when it reads in the data and handle it intelligently. At least, we can hope so.

To do the analysis, we can create a baseline model that contains only the intercept as a predictor of post-intervention exercise (note we have no missing data, so I have omitted the *na.action* option):

```
intercept<-gls(Post_Exercise~1, data = exerciseData, method = "ML")
```

Then add in a random intercept that varies over classrooms (**Classroom**):

```
randomInt <-lme(Post_Exercise~1, data = exerciseData, random = ~1|Classroom, method = "ML")
```

The final model updates the random intercept model to include intervention as a predictor:

```
intervention<-update(randomInt, .~. + Intervention)
```

We can compare these models with:

```
anova(intercept, randomInt, intervention)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
intercept	1	2	853.0918	861.5330	-424.5459			
randomInt	2	3	836.3542	849.0159	-415.1771	1 vs 2	18.737664	<.0001
intervention	3	6	836.8585	862.1820	-412.4292	2 vs 3	5.495703	0.1389

Adding in the random effect of intercepts across classrooms results in a significantly better-fitting model, $\chi^2(1) = 18.74$, $p < .001$. However, adding in the effect of intervention does not, $\chi^2(3) = 5.50$, $p = .14$. Note that with the addition of intervention we get three new degrees of freedom (6 degrees of freedom instead of 3). This might seem odd given we have added only one predictor, but this reflects the fact that **Intervention** has been split into three dummy variables comparing each group against the first category (the control). As such, we have added three parameters to the model, not one.

We can get a model summary using:

```
summary(intervention)
```

Intervention	Classroom	Pre_Exercise	Post_Exercise
Control	1	2.54951	2.54951
Control	1	2.345208	2.345208
Control	1	0.7071068	0.7071068
Control	1	0.7071068	0.7071068
Control	1	1.224745	0.7071068
Control	1	1.870829	2.121320
Control	1	2.121320	2.121320
Control	1	1.581139	1.581139
Control	1	0.7071068	0.7071068
Control	1	1.581139	1.581139
Control	1	1.581139	1.224745
Control	1	1.581139	2.345208
Control	1	1.581139	2.54951
Control	1	1.581139	0.7071068
Control	1	2.738613	1.870829
Control	1	2.121320	2.121320
Control	1	1.581139	1.581139
Control	1	0.7071068	1.224745
Control	1	0.7071068	0.7071068
Control	1	1.224745	1.581139
Control	1	2.121320	2.121320
Control	1	0.7071068	0.7071068
Control	5	2.54951	2.54951

The model summary is as follows:

Linear mixed-effects model fit by REML

Data: exerciseData
 AIC BIC logLik
 849.9616 875.2372 -418.9808

Random effects:

Formula: ~1 | Classroom
 (Intercept) Residual
 StdDev: 0.1542001 0.5392267

Fixed effects: Post_Exercise ~ Intervention

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.6479107	0.07897683	481	20.865749	0.0000
Intervention[T.Leaflet]	0.1874555	0.11527284	18	1.626189	0.1213
Intervention[T.Leaflet + Plan]	0.2493624	0.11613323	18	2.147209	0.0456
Intervention[T.Leaflet + Quiz]	0.1351455	0.11152106	18	1.211838	0.2412

Correlation:

	(Intr)	I[T.L]	I[T.+P]
Intervention[T.Leaflet]	-0.685		
Intervention[T.Leaflet + Plan]	-0.680	0.466	
Intervention[T.Leaflet + Quiz]	-0.708	0.485	0.482

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-2.60230928	-0.52527291	0.02427543	0.63070950	2.80192525

Number of Observations: 503

Number of Groups: 22

The fixed effects give the information in which most of you will be interested. Interestingly, although we know that intervention did not significantly improve the fit of the model, $\chi^2(3) = 5.50$, $p = .14$, when this effect is broken down we find that although the leaflet condition, $b = 0.19$, $t(18) = 1.63$, $p >$

.05, and the leaflet and quiz condition, $b = 0.14$, $t(18) = 1.21$, $p > .05$, did not differ from the control group, the leaflet and plan group did, $b = 0.25$, $t(18) = 2.15$, $p < .05$.

The result from these data could be that the condition failed to affect exercise. However, there is a lot of individual variability in the amount of exercise people get. A better approach would be to take into account the amount of self-reported exercise prior to the study as a covariate.

Task 4

- Repeat the above analysis but include the pre-intervention exercise scores (**Pre_Exercise**) as a covariate. What difference does this make to the results?

This task is a simple extension of the previous one. Therefore, we can simply update the model called *intervention* to include **Pre_Exercise**:

```
finalIntervention<-update(intervention, .~. + Pre_Exercise)
anova(intervention, finalIntervention)
summary(finalIntervention)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
intervention	1	6	836.8585	862.1820	-412.4292			
finalIntervention	2	7	389.6581	419.2022	-187.8290	1 vs 2	449.2004	<.0001

Adding in the effect of **Pre_Exercise** significantly improves the fit of the model (i.e., it is a significant predictor of post-intervention exercise), $\chi^2(1) = 449.20$, $p < .0001$. The model is:

```
Linear mixed-effects model fit by maximum likelihood
Data: exerciseData
      AIC      BIC    logLik
389.6581 419.2022 -187.8290

Random effects:
Formula: ~1 | Classroom
      (Intercept)  Residual
StdDev:  0.04170216 0.3493499

Fixed effects: Post_Exercise ~ Intervention + Pre_Exercise
              Value Std.Error DF   t-value p-value
(Intercept)    0.4424141 0.05698607 480    7.763548 0.0000
Intervention[T.Leaflet] 0.1584760 0.05079900  18    3.119668 0.0059
Intervention[T.Leaflet + Plan] 0.2184495 0.05161469  18    4.232312 0.0005
Intervention[T.Leaflet + Quiz] 0.2070905 0.04999701  18    4.142058 0.0006
Pre_Exercise    0.7154123 0.02650430 480   26.992306 0.0000
Correlation:
              (Intr) I[T.L] I[T.+P I[T.+Q
Intervention[T.Leaflet] -0.415
Intervention[T.Leaflet + Plan] -0.409 0.480
Intervention[T.Leaflet + Quiz] -0.482 0.493 0.486
Pre_Exercise          -0.783 -0.024 -0.024 0.053

Standardized Within-Group Residuals:
      Min       Q1       Med       Q3       Max
-3.7344350 -0.6164039  0.1411349  0.4840052  3.4070963

Number of Observations: 503
Number of Groups: 22
```

Note that, unlike before, now we are factoring in the baseline level of exercise ($b = 0.72$, $t(480) = 26.99$, $p < .0001$), all of the interventions have a significant effect compared to the control group: leaflet, $b = 0.16$, $t(18) = 3.12$, $p < .01$; leaflet and plan, $b = 0.22$, $t(18) = 4.23$, $p < .001$; and leaflet and quiz, $b = 0.21$, $t(18) = 4.14$, $p < .001$.