Exercise 18 - Basics of linear mixed models

Zoltan Kekecs

19 May 2021

Table of Contents

Abstract 1
Data management and descriptive statistics 1
Load bullying data 1
Check the dataset 2
Basics of linear mixed models 2
Exploring clustering in the data 2
Mixed models 5
Two types of effects in mixed models (types of predictors) 5
Two types of random effects 6
Building the models 9
Assessing model fit of mixed models and deciding which model to use 13
Determining significance of the model as a whole 17
What to report 17

Abstract

In this exercise you will learn how linear mixed models (aka multilevel models, mixed effects models) work and how to implement these models in SPSS. You will also learn how to make a decision about which random effect term to use, and what to report of the results of mixed models.

Data management and descriptive statistics

Load bullying data

In this exercise we will work with simulated data about bullying. Let's say that we are interested in predicting how much body size can predict vulnerability to bullying among

4th grade primary school children. Vulnerability to bullying in this study is quantified by the number of lunch sandwiches taken from the child during the period of one month based on self report. The predictor of interest in this study is weight.

The researchers hypothesize that the smaller the child, the more sandwitches will be taken from him or her.

Participants come from different classes in the same school which is denoted in the variable 'class'.

Variables:

- sandwich_taken - This is the measure of vulnerability to bullying: the number of lunch sandwiches taken from the child by the bullies over a period of a month.

- weight - weight in kilograms

- class - factor variable indicating the school class the child belongs to in primary school.

The datasets can be downloaded from this link:

https://github.com/kekecsz/SIMM32/blob/master/2021/Lab_4/Data_bully_int.sav

https://github.com/kekecsz/SIMM32/blob/master/2021/Lab_4/Data_bully_slope.sav

Check the dataset

First, we are going to work with the data containing a random intercept (see the meaning below), which is called Data_bully_int.sav.

As always, you should start by checking the dataset for coding errors or data that does not make sense, by eyeballing the data through the data view tool, checking descriptive statistics and through data visualization.

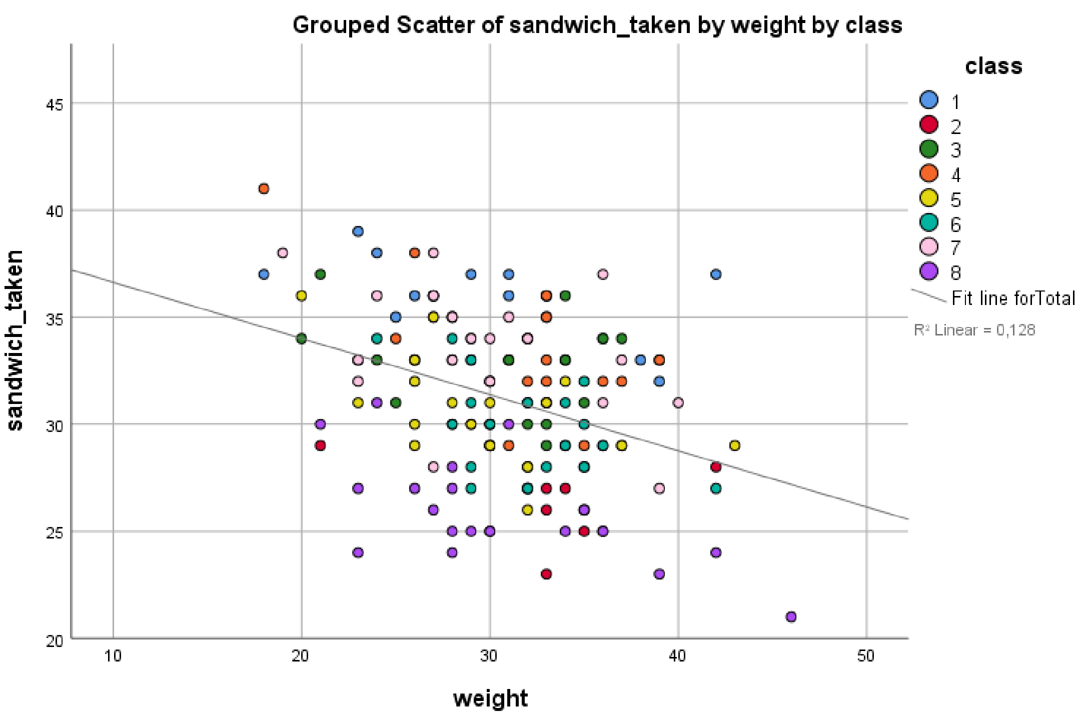
Basics of linear mixed models

Exploring clustering in the data

Let's plot the relationship for the simple regression model of sandwitch_taken predicted by weight on a scatterplot. It seems that there is a clear negative relationship if weight and the number of sandwitches taken from the children, howeverm the variability seems pretty large.

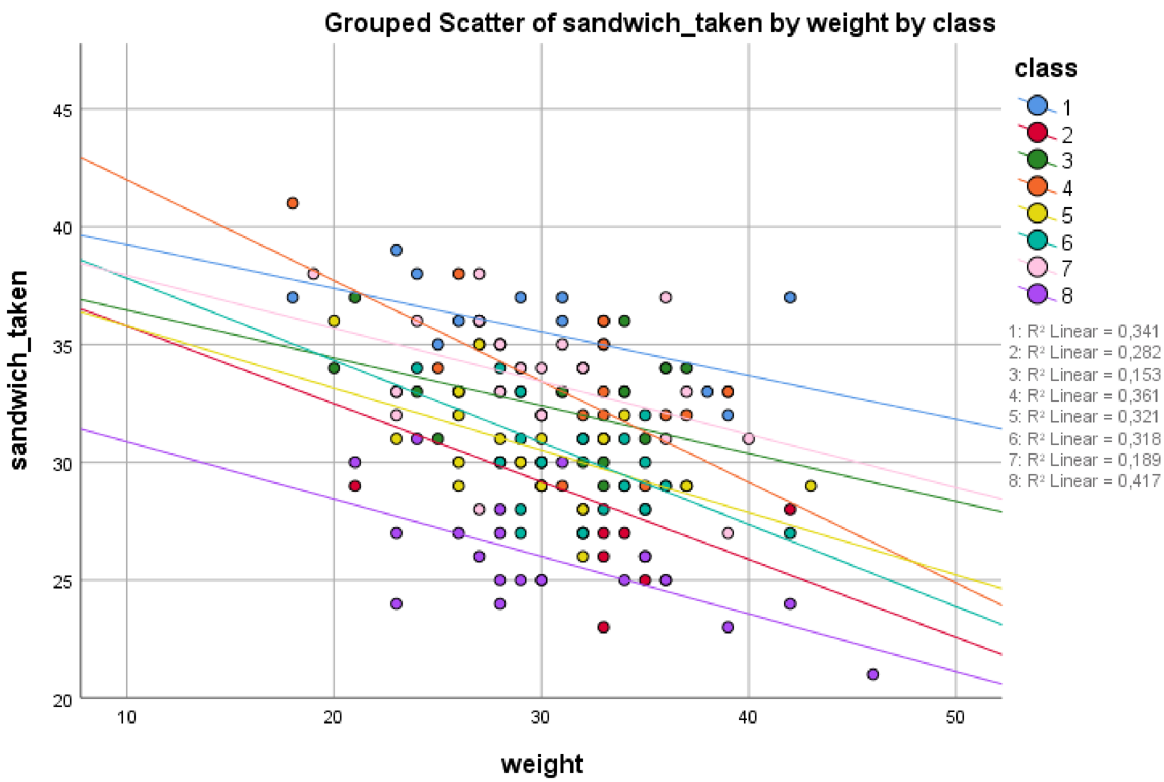
If we look at the color of the dots showing class membership, we may notice that children coming from the same class are grouped together on the scatterplot. This indicates that there is some "clustering" in the data, so observations might not be completely independent from each other.

```
GGRAPH

  /GRAPHDATASET NAME="graphdataset" VARIABLES=weight sandwich_taken class
MISSING=LISTWISE

    REPORTMISSING=NO

  /GRAPHSPEC SOURCE=INLINE

  /FITLINE TOTAL=YES SUBGROUP=NO.

BEGIN GPL

  SOURCE: s=userSource(id("graphdataset"))

  DATA: weight=col(source(s), name("weight"))

  DATA: sandwich_taken=col(source(s), name("sandwich_taken"))

  DATA: class=col(source(s), name("class"), unit.category())

  GUIDE: axis(dim(1), label("weight"))

  GUIDE: axis(dim(2), label("sandwich_taken"))

  GUIDE: legend(aesthetic(aesthetic.color.interior), label("class"))

  GUIDE: text.title(label("Grouped Scatter of sandwich_taken by weight by class"))

  ELEMENT: point(position(weight*sandwich_taken), color.interior(class))

END GPL.
```



Let's see if class membership can explain some of this variability. We can plot the regression lines for each class separately. This seems to be able to explain some of the variability in the data, bringing the regression lines closer to the actual observations. So it seems that it would be worthwile to take into account class membership of the participant when assessing their vulnerability to bullying to get good predictions.

```
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=weight sandwich_taken
class_rec MISSING=LISTWISE
    REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE
  /FITLINE TOTAL=NO SUBGROUP=NO.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
  DATA: weight=col(source(s), name("weight"))
  DATA: sandwich_taken=col(source(s), name("sandwich_taken"))
  DATA: class_rec=col(source(s), name("class_rec"), unit.category())
  GUIDE: axis(dim(1), label("weight"))
  GUIDE: axis(dim(2), label("sandwich_taken"))
  GUIDE: legend(aesthetic(aesthetic.color.interior), label("class_rec"))
  GUIDE: text.title(label("Grouped Scatter of sandwich_taken by weight by
class_rec"))
  ELEMENT: point(position(weight*sandwich_taken), color.interior(class_rec))
  ELEMENT: line(position(smooth.linear(weight*sandwich_taken)),
split(class_rec))
END GPL.
```



Mixed models

Right now we are only interested in whether or not weight influences bullying vulnerability, and if so, to what extent. Class membership is secondary to our interests, and even if we would be able to establish the particular effect of any of the classes in this school, in other schools this information would be useless, since there would be different classes.

We use random effect predictors for which we have many levels in real life, but we only have information about a small subset of those levels in our dataset, and knowing the effects of these particular levels, like the effect of class 1 in school X, would not be very useful when we use the regression equation on new data, because the new data will most likely not come from class 1 in school X. So this is more of a nuisance variable for us, we can't do much with knowing its effect, but still we need to take into account that data is clustered according to these levels to correctly estimate the precision

Using linear mixed models we can take into account that data is clustered according to class membership, without having to enter class as a classical predictor in our model. Instead, we can enter it into the model as a "random effect" predictor.

Two types of effects in mixed models (types of predictors)

Predictors can be entered into mixed effect models as having two types of effects: the predictor can either have a "fixed effect" on the outcome or a "random effect" on the outcome.

Fixed effects predictors - These are the good old predictors that we are familiar with from the regular linear models. We would like to estimate this effect so that we can use it in future predictions.

Random effect predictors - These are the "nuisance variables", the categorical variables that have many possible levels in the wild but we only have a few of these levels observed in our sample. We do not want to estimate the effects of these predictors on the outcome, and don't want these in our regression equation, because we will not be able to use the coefficients for the observed levels for prediction in the new data. But we cannot simply disregard these variables, because they influence the outcome, and thus, because the observations are related to each other based on this variable, this affects how precise our estimation of the model's effectiveness is and how precise the model coefficient's estimates are. So we need to factor the effect of these variables into our models to get a clear picture about the precision of our estimates.
The logic is to assume that the groups or clusters represented by this nuisance variables have a "random effect" on the outcome variable, and the effect of the groups are drawn from a normal distribution. This means that if we took enough of these groups, their effect would cancel out, since some would have a positive effect, while others would have a negative effect on the outcome, so it is not necessary to know exactly what the effect of each possible group is on the outcome, all we need to know is that there is this a variable that creates some "noise" in our data, and the amount of noise it generates. If we know this, we can factor this into the precision of our other estimates.

Models that contain both fixed and random effect predictors are called **mixed models**. So this is where the name comes from.

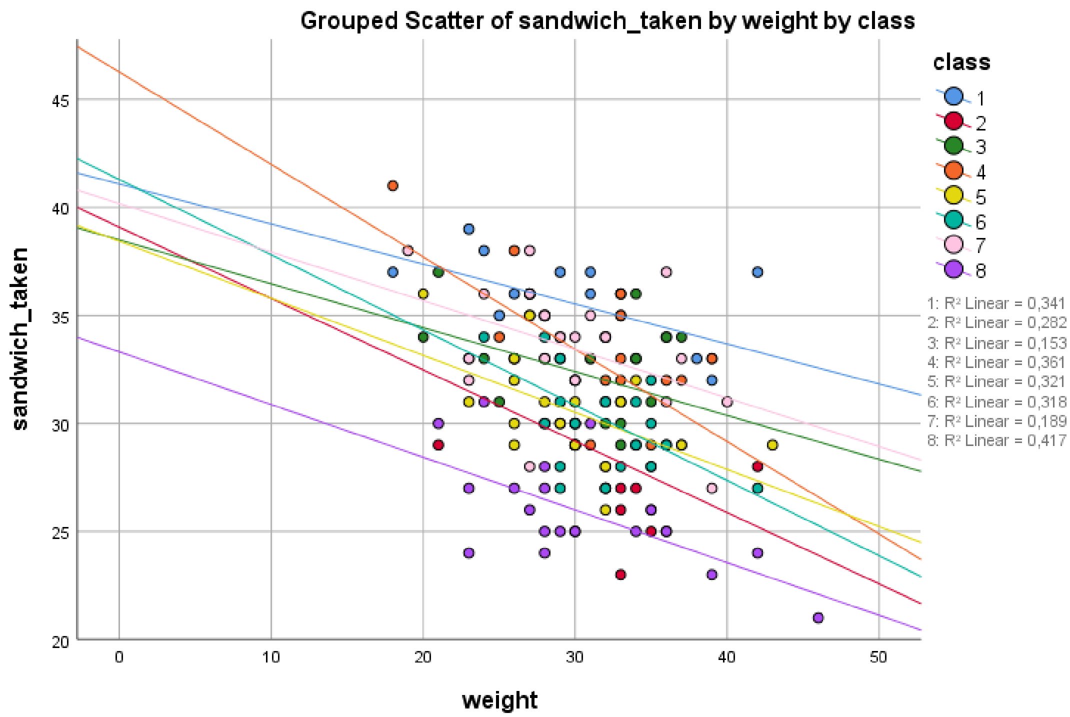
Two types of random effects

There is generally two ways in which a random effect predictor can influence the outcome variable:

Random intercept, but no random slope: it is possible that predictor has a direct effect on the outcome, that is, the clusters within this predictor are only different in how high or low they are on the value of the outcome variable on average, but the effect of the fixed effect predictors are the same in all of the clusters. This is what we see in the dataset named data_bully_int. You can observe that in some classes children lose more sandwiches on average, while in other classes children lose less sandwiches. Also notice that weight affects number of sanwiches taken by rougly the same amount in all classes in data_bully_int.

If we build different regression models for each class and visualize them on a plot, the scatterplot shows that the regression lines cross the y axis at different places (different intercepts), but the regression lines are almost parallel to each other (similar slopes for the regression lines), indicating that the effect of weight is similar in the classes.

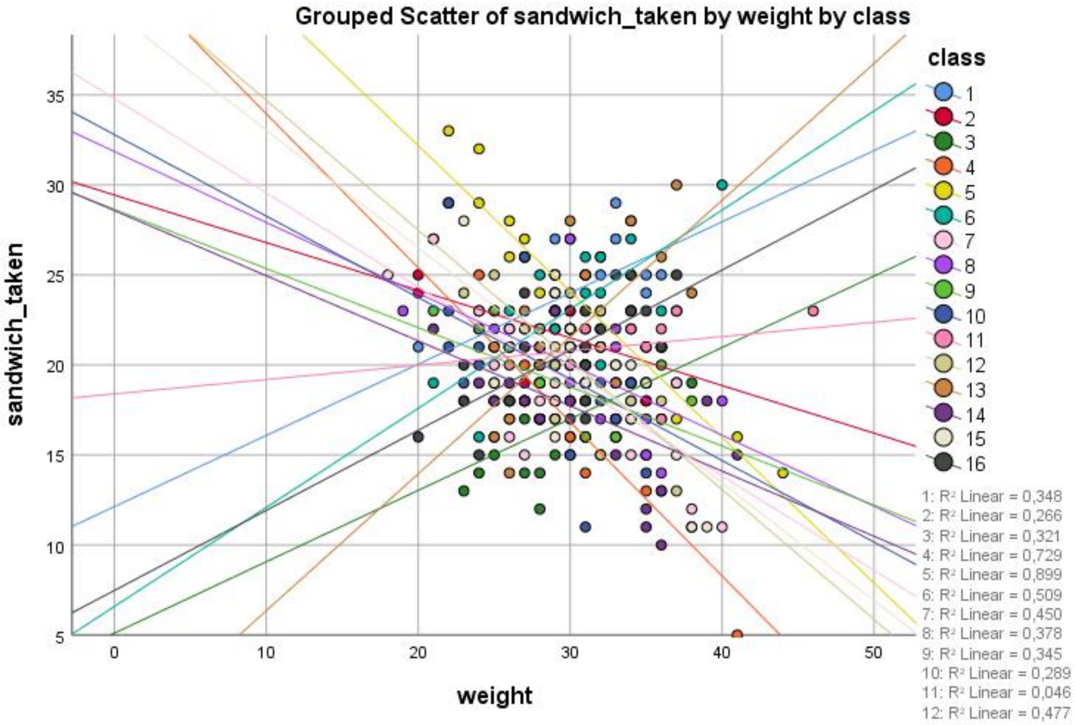
```
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=weight sandwich_taken
class MISSING=LISTWISE
    REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE
  /FITLINE TOTAL=NO SUBGROUP=YES.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
  DATA: weight=col(source(s), name("weight"))
  DATA: sandwich_taken=col(source(s), name("sandwich_taken"))
  DATA: class=col(source(s), name("class"), unit.category())
  GUIDE: axis(dim(1), label("weight"))
  GUIDE: axis(dim(2), label("sandwich_taken"))
  GUIDE: legend(aesthetic(aesthetic.color.interior), label("class"))
  GUIDE: text.title(label("Grouped Scatter of sandwich_taken by weight by class"))
  SCALE: linear(dim(1), min(0))
  ELEMENT: point(position(weight*sandwich_taken), color.interior(class))
END GPL.
```



random intercept, AND random slope: Now let's look at the other dataset called Data_bully_slope.sav. This dataset is also simulated, and it comes from the same scenario as the data_bully_int dataset, but the data looks a bit different. When we plot the regression lines for the classes separately, we find that not only the mean value of the stolen sandwitches is different across classes, but the effect of weight also seems to be different in the different classes.

For example in Class 2 (the yellow dots), weigth has a strong negative effect on the number of sandwiches taken, for class 7 (pink dots) weight seems to make almost no difference, while for class 6 (light blue) there seems to be a positive relationship between weight and sandwiches taken.

On the scatterplot, you can easily spot this by noticing that both the intercepts and the slope of the regression lines are different across classes.



An even clearer picture of what is the effect like in each group can be seen if the same scatterplot is produced for each class separately. You can do this in the **Data>Split file**… procedure, by selecting the **Organize output by groups** option, and entering class in the "group based on" box. Now if you run the scatterplot again, you will find that each class will have a separate scatterplot. To make the plots comparable, it is helpful to specify that the plots should contain the origin, 0 value on both the y and the x axis. This can be set in the chart builder in the element properties by selecting the axis and unticking the "Automatic" box for the minimum value, and entering 0 manually in the textbox.

Important: when you split your file using the Split file procedure, all your analyses will be done separately for each level of the group you use for splitting. You should revert this once you are done with the plotting by going back to the **Data>Split file**… menu and selecting **"Analyze all cases, do not create groups"** option!

Building the models

Let's see how we can use this knowledge about clustering in the dataset to get more accurate picture about the influence of the predictive value of the predictors on the outcome, and to ultimately make better predictions. We will use the data_bully_slope dataset here.

Now we are going to fit three different models, a simple fixed effect model with no random effects, a **random intercept model**, and a **random slope model**. Remember that based on the above plots, we suspect that the random effect predictor, class, has an effect on both the mean of the outcome (intercept), and the effect of the fixed effect predictor (slope) in the data_bully_slope dataset. So in reality we would probably only fit the random slope model.

The other models are just here to show you how they differ in prediction effectiveness and how to formulate them.

First, let's fit the simple linear regression model as we have learned in the previous exercises. Notice that this model only contains a single *fixed effect* predictor, weight, so we will save this regression model to a model called mod_fixed.

simple regression (only fixed effect term, no random effect):

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS CI(95) R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT sandwich_taken
  /METHOD=ENTER weight.
```

Model Summary[a]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,692[b]	,478	,449	2,300

a. class = 16

b. Predictors: (Constant), weight

ANOVA[a,b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	87,329	1	87,329	16,508	,001[c]
	Residual	95,221	18	5,290		
	Total	182,550	19			

a. class = 16

b. Dependent Variable: sandwich_taken

c. Predictors: (Constant), weight

Coefficients[a,b]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	7,475	3,283		2,277	,035
	weight	,445	,110	,692	4,063	,001

a. class = 16

b. Dependent Variable: sandwich_taken

Random intercept model:

We have to use a different analysis type to fit a mixed effect model (including both fixed and random effects). This is in **Analyze > Mixed models > Linear**.

On the first dialog box, we would enter 'class' **in the subjects box**, click next,

In the next dialog box, we would specify the model: sandwiches_taken will be the dependent variable, and weight will be a covariate.

You will also want to specify some other things in the menus:

Fixed…: Put the predictor weight in the Model box

Random…: In the top of the random effects box, "check include intercept"; under covariance type you should enter: Variance components (this is the default); and in the Subject groupings enter class into the Combinations box. (DO NOT enter weight into the model box, since that would create a random slope model.)

Statistics…: Check Parameter estimation

If all this is set, we can run the model.

This means that we allow the model to fit a separate regression line to each of the clustering levels (in this case, classes), but we restrict it so that all of these regression lines have to have the same slope.

We would do this if we suspected that the clustering variable (random effect predictor) would have no influence on the effect of the fixed effect predictor. So based on what we saw on the plots above, this would be a good fit for the data_bully_int dataset. But here we fit this to the data_bully_slope dataset, so we can compare the effectiveness of random slope and random intercept models.

The "Estimates of Fixed Effects" table in the output provides us with the model coefficients and the confidence intervals.

```
MIXED sandwich_taken WITH weight
  /CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100) MXSTEP(10) SCORING(1)
    SINGULAR(0.000000000001) HCONVERGE(0, ABSOLUTE) LCONVERGE(0, ABSOLUTE) PCONVERGE(0.000001, ABSOLUTE)
  /FIXED=weight | SSTYPE(3)
  /METHOD=REML
  /PRINT=SOLUTION
  /RANDOM=INTERCEPT | SUBJECT(class) COVTYPE(VC).
```

Estimates of Fixed Effects[a]

Parameter	Estimate	Std. Error	df	t	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Intercept	27,155272	1,372192	236,225	19,790	,000	24,451975	29,858569
weight	-,230474	,041501	304,906	-5,553	,000	-,312139	-,148808

a. Dependent Variable: sandwich_taken.

random slope model (allowing BOTH random intercept and random slope):

The model can be fit just like in the case of the random intercept model, but in the Random… menu you would **put weight in the Model box**, and covariance type should be set to: **Unstructured**

Fixed…: Put the predictor weight in the Model box

Random…: In the top of the random effects box, “check include intercept”; under covariance type you should enter: **Unstructured**; **in the Model box include weight**; and in the Subject groupings enter class into the Combinations box.

Statistics…: Check Parameter estimation

This means that we allow the model to fit a separate regression line to each of the clustering levels (in this case, classes), and we do not restrict the slope or the intercept of this regression line (other than the line needs to be linear).

We do this because we suspect that the clustering variable (random effect predictor) influences both the mean value of the outcome and the effect of the fixed effect predictors.

For some reason in my current version of SPSS, the “Unstructured” covariance type is not carried out even if we select it in the menu, so we need to correct this in the syntax. You have to paste the command into the syntax when you are done with the above settings, and

```
MIXED sandwich_taken WITH weight
  /CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100)
MXSTEP(10) SCORING(1)
    SINGULAR(0.000000000001) HCONVERGE(0, ABSOLUTE) LCONVERGE(0,
ABSOLUTE) PCONVERGE(0.000001, ABSOLUTE)
  /FIXED=weight | SSTYPE(3)
  /METHOD=REML
  /PRINT=SOLUTION
  /RANDOM=INTERCEPT weight | SUBJECT(class) COVTYPE(UN).
```

Estimates of Fixed Effects[a]

Parameter	Estimate	Std. Error	df	t	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Intercept	25,494684	3,834857	14,878	6,648	,000	17,315048	33,674321
weight	-,173378	,132070	14,838	-1,313	,209	-,455148	,108391

a. Dependent Variable: sandwich_taken.

Assessing model fit of mixed models and deciding which model to use

As always, when selecting model components, you should make decisions based on theoretical considerations and prior research results. So in this case, whether it makes sense theoretically for class membership to influence the slope of the effect of weight as well or not, or whether previous exploratory studies have shown that modeling a random slope roduces much better model fit than a simple random intercept model.

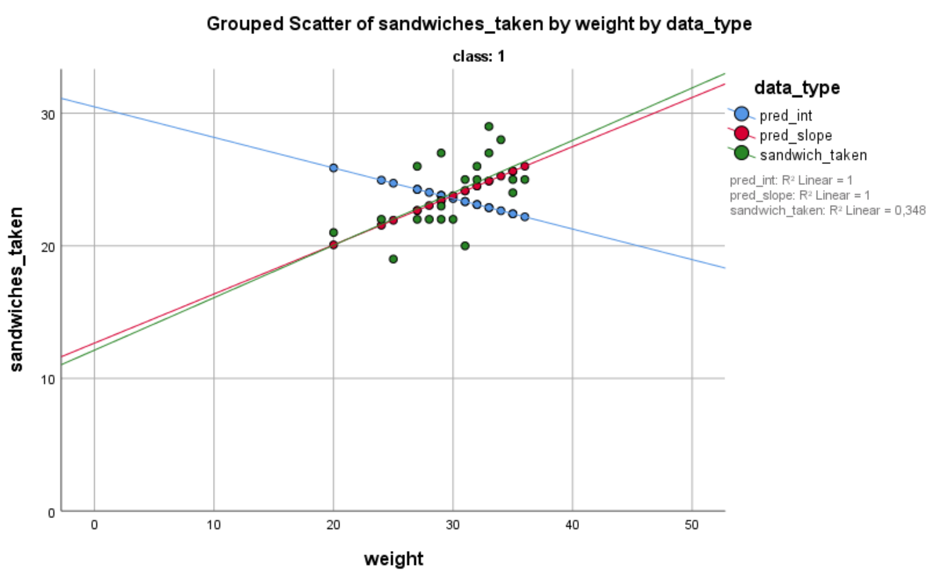
In some case however, we are exploring the topic without strong prior research or theoretical cues. In these cases it may be appropriate to select between random slope and random intercept models based on model fit. But this decision needs to be clearly documented in the publication. The best if such decisions are pre-registered before data collection.

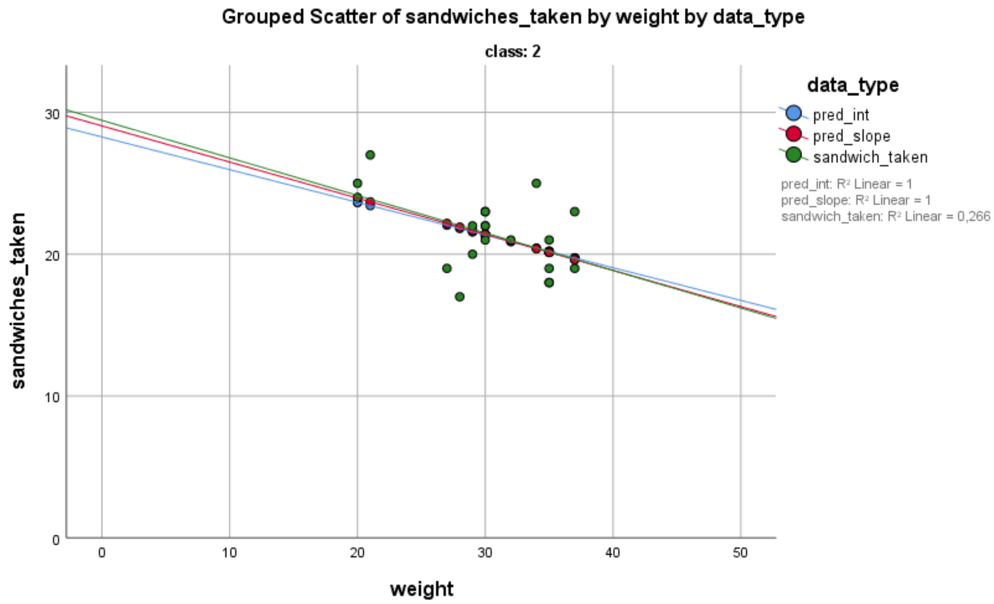
Visualization

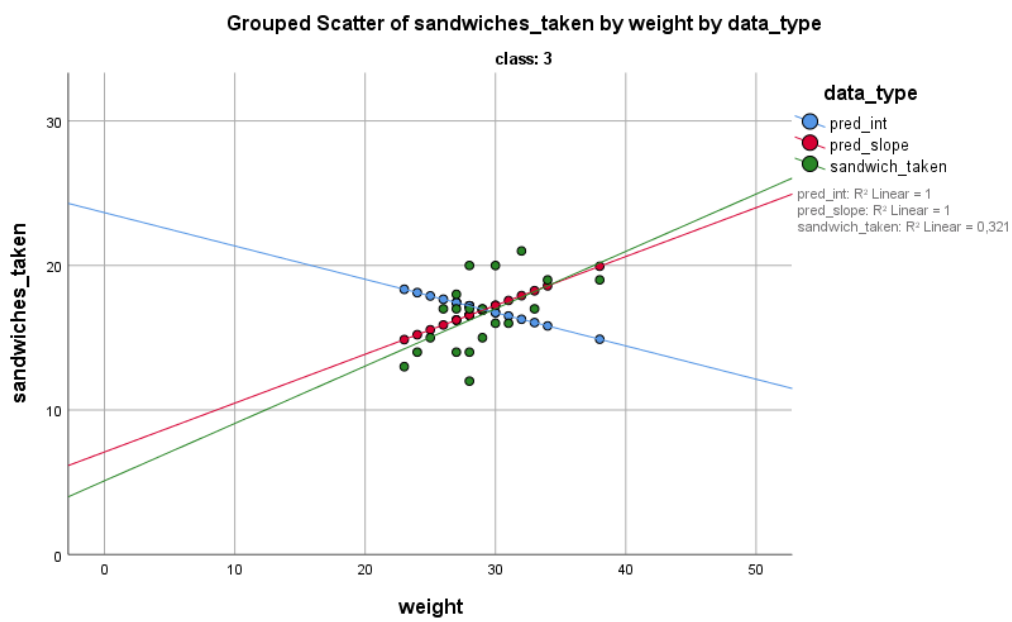
Visualization plays an important role in assessing the model fit of mixed effect models and potentially guiding decisions about whether to use random slope or random intercept models.

We can plot the regression lines of the two models by saving the predictions of the models into a variable. We can follow the following steps:

- Saving the predictions of the models into new variable (be sure to check the predicted values checkbox in the "Predicted values and residuals" box instead of the "predicted values" box, so that you will use the information from the groups (clusters) as well, instead of just using the predictions based on the fixed effects coefficients).
- Restructuring the dataset so that all of the predictions and the actual observed values are in the same column one after the other, and that the corresponding class, and weight information is copied over as well. (You can do this manually or using the **Data > Restructure** function.
  During this process you will have to create an index variable (for example called "data_type"), which will tell you which lines contain the actual observed sandwiches taken vales, and which contain the predictions from which models. Even if you use a numerical variable to make this distinction, it is best if you define the vale labels in the variable view so that it is obvious which data point corresponds to which prediction or the observed data.
- Split data by Class using the **Data > Split** data function (you will have to allow SPSS to sort the cases by this variable)
- Use the chart builder to build either a grouped scatterplot or a grouped line plot, where the y axis contains the sandwiches taken values (predicted and observed), the x axis contains weight, and the color will be defined by data_type (the index variable). While setting this up, it is helpful to set the minimum and maximum values on the axes (as described above), and also to ask for a fit line through the subgroups. These can be set in the Element properties tab in the Chart builder.







This way, for each class, you will get a plot depicting the actual observed values, and the predicted values by the random intercept and the random slope model. Now you can look at these plots to see which model's predictions match the observed data better. The random slope model will always be somewhat better, but if the difference is not substantial, you might decide to use the random intercept model, to restrict the flexibility of the model and avoid overfitting.

When comparing the plots, we can see that for many of the classes the random intercept model is not good enough, often the relationship between weight and sandwich taken is positive in the classes, and the random intercept model's prediction does not fit these classes at all.

Comparing model fit via model fit indices

If we would compare the prediction error of the 3 models we just built (if we would ask for the residuals to be saved in the Save menu), we would see that the RSS is largest for the fixed effect model and the smallest for the random slope model. But this is no surprise, since we allow for increasingly more flexibility in the model to fit our data in the random intercept, and the random slope models. So relying on RSS alone in our original dataset when comparing prediction efficiency would be misleading (although RSS would still be useful if we used a training-set, test-set approach we saw in the Exercise on Overfitting).

Instead, we can use model fit indices that are sensitive to the number of predictors, such as AIC.

Note that in the case of the random effect models, it is more appropriate to use a model fit index which is corrected for overparametrized models. SPSS provides Schwartz's Bayesian Information Criterion (BIC) and Bozdogan's Consistent AIC (CAIC) which can be used for model selection purpose. Both of these have been shown to perform well when the random effects are uncorrelated. However, their performance is poorer and lead to bad decisions in model comparison when the random effects are correlated. This is only an issue when you have multiple random effects, such as a random intercept and a slope. In a simple random intercept model, there is only one random effect, the intercept, nothing else to correlate with, so we can use the AIC and the CAIC in these cases.

When the correlation of random effects is suspected (in a random intercept and slope model where we use an Unstructure covrience type), the use of regular AIC is advised in SPSS. In other software there are better model fit indices, for example the conditional AIC (cAIC, not to be confused with consistent AIC here refered to as CAIC). More details about the comparative effectiveness of different model fit indices and formulas to compute them can be found in the following publication: (Vallejo, Tuero-Herrero, Núñez & Rosário, 2014).

If you are not sure whether the random effect are correlated or not in the random slope model, you can get information on the correlation of the random effects by asking for **Correlation of parameter estimates** in the **Statistics menu** of the linear mixed model builder dialog box. In the output, you can look at the "Correlation matrix of the estimates of the fixed effects".

Reference: Vallejo, G., Tuero-Herrero, E., Núñez, J. C., & Rosário, P. (2014). Performance evaluation of recent information criteria for selecting multilevel models in behavioral and social sciences. *International Journal of Clinical and Health Psychology*, *14*(1), 48-57.

In the case of the Data_bully_slope, the AIC of the random slope model is smaller by more than 2, indicating that **the random slope model seems to be a better fit**.

Determining significance of the model as a whole

To see if the models are better than the null model without any fixed effect predictors (only the random effect of intercept), you can build the **null model** the same way as the other models, except **in the Fixed…: menu you need to take out all predictors from the Model box** (remove weight from there if it is in there). Compare the AIC (or BIC, CAIC as appropriate) of the null model with the model with the fixed effect predictors. Use the regular criteria: smaller AIC is better, and if the difference between the AIC is greater than 2, there is a significant difference between the predictive power of the models.

```
MIXED sandwich_taken WITH weight
 /CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100)
MXSTEP(10) SCORING(1)
   SINGULAR(0.000000000001) HCONVERGE(0, ABSOLUTE) LCONVERGE(0,
ABSOLUTE) PCONVERGE(0.000001, ABSOLUTE)
 /FIXED=| SSTYPE(3)
 /METHOD=REML
 /PRINT=SOLUTION
 /RANDOM=INTERCEPT | SUBJECT(class) COVTYPE(VC).
```

What to report

We have to report the same information about linear mixed models as for fixed-effect-only linear models seen in the previous exercises.

You could describe the following in your **methods section** about the model formulation:

"In order to determine the influence of weight on vulnerability to bullying, we used a linear mixed model. In the model the outcome variable was the number of sandwitches taken, and we used a single fixed effect predictor, weight. Because data was clustered in school classes, we included the random effect of class in the model. No prior data or theory was available about how the random effect of class might manifest, so we built two separate models. In one model we only allowed for a random intercept of classes, while in the other model we allowed for both random intercept and the random slope of the effect of weight across different classes. As pre-registered in our experimental protocol, we compared the model fit of the random intercept and slope models using the AIC model fit index, and we made a choice between these two models based on this index."

Model R squared

We need to use a special type of the R squared called the marginal R squared in the case of mixed effects models that shows the proportion of variance explained by the fixed factor(s) alone, when not taking into account the random effect terms.

Reference: Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. Methods in Ecology and

Evolution, 4(2), 133-142. and Johnson, P. C. (2014). Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models. Methods in Ecology and Evolution, 5(9), 944-946.

Marginal R squared (variance explained by the fixed effects in the model) can be manually computed in SPSS using the formulae in Nakagawa & Schielzeth (2013). When using a Gaussian model, the formula is:

R^2m = Vf / (Vf + Vr + Ve).

Conditional R^2 (variance explained by both the fixed and random effects in the model) can be computed with this formula:

R^2c = (Vf + Vr) / (Vf + Vr + Ve)

Source: https://ecologyforacrowdedplanet.wordpress.com/2013/08/27/r-squared-in-mixed-models-the-easy-way/

In the above formulae, Vf is the fixed effects variance, which can be calculated by saving the **fixed effect predicted values** based on the model (ask for this in the **Save menu**, make sure to ask for the predicted values in the **"Fixed predicted values" box** instead of the "Predicted values and residuals" box!), and calculating their **variance** (for example by using the **Analyze** > **Descriptive Statistics** > **Descriptives** command, and in the **options** asking for **variance**.

Vr is the sum of the random effects variances, and Ve is the residual variance. This can be found in the **Estimates of Covariance Parameters table** in the output of the linear mixed model.

R^2 of the random intercept model

Estimates of Covariance Parameters[a]

Parameter		Estimate	Std. Error	
Residual		11,678184	,948788	Ve
Intercept [subject = class]	Variance	4,512842	1,862085	Vr

a. Dependent Variable: sandwich_taken.

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation	Variance	
Fixed Predicted Values	320	16,55	23,01	20,2094	1,07576	1,157	Vf
Valid N (listwise)	320						

R^2m = 1.16 / (1.16 + 4.51 + 11.68) = 0.067

R^2c = (1.16 + 4.51) / (1.16 + 4.51 + 11.68) = 0.33

R^2 of the random slope model

The calculation of Vr for random slope models in SPSS is not this straightforward, so I would recommend that you compute the marginal and conditional R^2 of random slope models in another statistical software, such as R in you know how to use these software. Here is the R code for building the random intercept and the random slope models in R and to compute the marginal and conditional R^2 for them. However, in most cases this would requires a deep understanding of how to build the same models in R as in SPSS.

https://github.com/kekecsz/SIMM32/blob/master/2021/Lab_4/calculating%20random%20effect%20marginal%20r%20sq.R

Here is an imperfect alternative that you can use in SPSS for approximation of the marginal and conditional R^2 for random slope models:

Calculating Vf and Ve works the same way as above. However, the Estimates of Covariance Parameters table will list multiple entries for the random effects, and it is not clear how to compute the Vr from there. So instead, we will approximate Vr by calculating the total variance (Vtotal), and subtracting Vf and Ve from it.

Vr = Vtotal – (Vf + Ve)

The total variance can be computed by getting the variance of the outcome variable (dependent variable), in our case, this is sandwich_taken. You can get the variance of this variable through the **Analyze** > **Descriptive Statistics** > **Descriptives** procedure, and in the **options** asking for **variance.**

Estimates of Covariance Parameters[a]

Parameter		Estimate	Std. Error	
Residual		6,643526	,553528	Ve
Intercept + weight [subject = class]	UN (1,1)	219,854012	86,259242	
	UN (2,1)	-7,522553	2,958735	
	UN (2,2)	,262353	,102446	

a. Dependent Variable: sandwich_taken.

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation	Variance	
sandwich_taken	320	5	33	20,21	4,106	16,862	Vtot
Fixed Predicted Values	320	17,52	22,37	20,2695	,80926	,655	Vf
Valid N (listwise)	320						

Vr = 16.86 – (0.66+ 6.64) = 9.56

R^2m = 0.66/ (0.66 + 9.56+ 11.68) = 0.03

R^2c = (0.66+ 9.56) / (0.66+ 9.56+ 6.64) = 0.61

In the results section, you should report the linear mixed models analysis as follows:

“The random slope model produced a better model fit according to the AIC (AIC intercept = 1735.01, AIC slope = 1609.00). Which favors the random slope model, so we chose the random slope model. Thus, we present the results of the random slope model in the following. (The results of the random intercept model are listed in the supplement.)

The random slope linear mixed model was not significantly better than the null model (AIC null = 1608.50, AIC slope = 1609.00). The fixed effect predictors explained 3% of the variance of sandwich taken, while the fixed and random effect predictors combined explained 61% of the variance (R^2m = 0.03, R^2c = 0.61)."

Statistics of the predictors

You will also have to report statistics related to the predictors, this is usually done in a table format, because most often we have multiple predictors (even though in this example we only have one). You can get the information about the important results related to the predictors by asking for Parameter estimates in the Statistics menu. The model coefficients and confidence intervals then can be found in the estimated fixed effects table in the output.

Table 1. Regression coefficients

	b	95% CI lb	95% CI ub	p-value
Intercept	25,49	17.31	33.67	>0,001
weight	-0.17	-0.46	0.11	0.209

Note that the use and interpretation of p-values for linear mixed models is controversial at the moment, so observe the trend in your particular sub-field and decide whether you want to use them or not. Confidence intervals give you information about statistical significance, so it is not necessary to provide p-values.

Standardized betas cannot be extracted in SPSS for mixed linear models, so these can be omitted from the report (or can be computed in another statistical software).