MSMR: Multilevel Models Scripts

Table of Contents

Week 1: Introduction to Multilevel Modelling

- Hello and welcome to Multivariate Statistics and Methodology with R. My name is Dan Mirman and I'll be teaching the Multilevel Modelling module, which is the first 5 weeks of the semester
    - Multilevel modelling is really just an extension of linear regression, which you learned about last semester in univariate statistics. But multilevel models allow you to capture nested or hierarchical data structures
- What do I mean by nested or hierarchical data structures?
    - Well, for example these would be clustered observations like measurements of multiple children that all come from the same family or classroom or neighbourhood
    - Or these could be repeated measurements of the same individual, like multiple questions on an exam or a within-subject experimental manipulation
    - A specific and important case of repeated measures is longitudinal data where the repeated measurements are spread out over time
  - As you can see, nested or hierarchical data are very common across different sub-disciplines of psychological science (they're actually very common in the behavioural sciences in general, as well as in various branches of biology and ecology). This makes multilevel modelling a very useful statistical method
- Here's one quick example of why this is useful: modelling gradual change
  - You can see from the graph that word learning was faster for high TP words than for low TP words
  - But when I tried a t-test on overall accuracy, it was only marginal.
  - A repeated measures ANOVA showed a main effect of Block, marginal effect of TP (that's the same thing as the t-test), and the interaction was not significant
  - Could do separate t-tests for each block, and they're significant only in block 4 (and marginal in block 5), but that's not a very compelling result
    - What we really need here is a model of the learning curve trajectories, which we can get from multilevel models
- Nested data have two key features that multilevel models can capture
    - Nested data are not independent. Taking a simple longitudinal example, a child that taller-than-average at time t, is likely to be taller-than-average at your next measurement time t+1
    - This non-independence is related to individual differences – that's a tall kid. So it's not just noise or a statistical nuisance; it's actually related to the phenomenon you're studying
    - Nesting can also happen at multiple levels: if you're measuring kids from the same family or hospital, then those height measurements have an additional layer of non-independence
  - The second key feature is that the clustered observations can be related by a continuous variable, like time.
    - When that happens, you want to model it as a continuous variable (for example, not as discrete measurement waves, but as measurements that were separated by a specific duration)
    - You might also be interested in the trajectories or shapes of change over the continuous course of that variable. In developmental or longitudinal contexts these are sometimes called "growth curves"
- So, multilevel models allow you to
    - correctly model the non-independence of observations

        - to simultaneously estimate group-level and individual-level effects
        - and to model trajectories of change
    - a quick note about nomenclature: “multilevel modelling” is part of a family of very closely related statistical methods that just have slight differences in terminology and implications about the data
        - in this course, I’ll mostly use “multilevel models” or I might slip up and say multilevel regression
        - they are also called hierarchical linear models or mixed effects models (also mixed effects regression and linear mixed models)
        - in longitudinal contexts with non-linear data they are also called growth curve models or growth curve analysis
        - I’m saying this because you’ll sometimes see these other terms and I want you to know that they are referring to the same kind of statistical method, just with a slightly different name
- Let’s start from a brief review of linear regression
    - Here you’ve got a relationship between outcome variable Y and predictor Time. Beta_0 is the “intercept”, which is the value of Y at Time=0. Beta_1 is the “slope”, which is the estimated change in Y per unit of Time
    - If we want to talk about individual observations Y of participant i at time j, we can write it like this, with the residual error epsilon_ij
    - In multilevel model terms, we would refer to beta_0 and beta_1 as “fixed effects”
    - And the error term as a “random effect”
- In multilevel models, we can think of that linear regression as a level-1 model
    - And the level-2 is a model of the level-1 parameters. That is to say, a model of beta_0i – the intercept for participant i, which is the population mean intercept (gamma_00) and that participant’s deviation from the population mean intercept (zeta_0i)
    - You could also imagine an effect of condition C on the intercept (gamma_0C)
    - And an analogous model for the level-1 parameter beta_1
- The fixed effects part of multilevel models is pretty similar to regular regression, the magic is in the random effects or residual errors
        - Because those zeta values are specific to a particular individual (or other kind of cluster, like family or school or whatever)
        - They are that individual’s unexplained deviation from the intercept and slope, so they reflect some aspect of their individual differences. We’ll talk more about this as we go on through the module
        - For now, I want to get that general idea and to know that these random effects require a lot of data to estimate.
        - In a sense, random effects are “the hard part” of multilevel modelling both conceptually (this is where you need to describe the nested structure of your data) and computationally (need lots of data to estimate).
- As a quick summary
        - Fixed effects are typically the things you’re studying, the reproducible properties of the world, like differences between nouns and verbs, effects of WM load or age, etc. The model will estimate unique, essentially unconstrained coefficients for each fixed effect condition
    - Random effects are the randomly sampled observational units over which you intend to generalise

        - So these might be particular nouns and verbs, particular individuals that happened to participate in your study, etc.
        - They reflect unexplained variance and that variance is assumed to follow a normal distribution with a mean of 0
- One more technical thing you need to know before we fit our first multilevel model: maximum likelihood estimation
        - This is a procedure for finding parameter estimates that maximize the likelihood of observing the actual data
        - For simple regression, ordinary least squares produces the MLE parameters by just solving an equation
        - This is not possible for multilevel models, so they use an iterative algorithm to gradually converge to MLE parameters. The algorithm is good, but it's not guaranteed to converge to MLE parameters, and we'll talk later about how to deal with convergence problems
    - For now, you need to know that the relevant goodness of fit measure is log-likelihood (often abbreviated LL)
        - Unlike R^2, log-likelihood is not inherently meaningful (it's not proportion of variance or anything like that)
        - But, changes or differences in LL are good measures of improvement in model fit.
        - In fact, -2 times the change in LL is distributed as chi-square with degrees of freedom corresponding to the number of added (or removed) parameters.
        - So it's easy to get a p-value for a comparison of two multilevel models, though the models have to have the same structure with parameters only added (or removed)
- Ok, that's the theoretical stuff, now let's talk about how to actually fit the models in R
        - You'll need the lme4 package
        - The key function is lmer() and it takes a formula, the data, and some options. Broadly similar to the lm() function for basic regression
        - Then you'll evaluate the model or models by comparing them, checking their parameter estimates, plotting model fits, and so on
        - Based on those evaluations, you will often want to improve or adjust the model in various ways and again do comparisons or evaluations.
        - So your analysis will often involve fitting multiple models for comparison or to get the random effects right
- Let's look at a concrete example
        - These are visual search response times and just from looking at the data we can guess that there will be two main effects: it takes more time to find the target when there are more distractors (effect of set size) and stroke survivors (aphasic group) are slower than the control group. It's not immediately clear whether the group difference gets bigger when there are more distractors – that would be a group-by-set size interaction
- Now let's fit the models using the lmer() function from the lme4 package
        - We can start with a null model: there's just one fixed effect -- an overall intercept for the RT (that's what the "1" stands for); but it has a more complex random effect structure – there are by-participant "random intercepts" (participants have different baseline RT) and "random slopes" (participants are differentially slower as the number of distractors increases.
        - The null model isn't very interesting, but it will serve as a baseline comparison for the next model: where we add a fixed effect of set size

        - The “1” is implied, so we don’t need to explicitly specify it unless there are no other terms in the formula. Also, if I’m feeling lazy, I’ll use F instead of writing out FALSE
        - Notice that the random effects stayed the same, so the only difference between the “null” model and this model is the addition of a single fixed effect
    - We can further build up the model by adding a fixed effect of Dx (that’s the group variable and will correspond to a baseline difference in RT) and the Dx-by-set size interaction
- Now that we’ve fit the models, we can compare them. The anova() function will do the comparison – this is a little confusing because the test is not an ANOVA, it’s a model comparison using change in log-likelihood and the chi-square distribution, as you can see from the output
- Here are the same model comparisons in a cleaner table. You can see the log likelihood values (logLik), the “deviance” is -2*logLik, and the Chisq column shows the change in the mode fit – that’s the chi squared statistic and the Df column has the number of added parameters. And from that you get the p-value.
        - So we can see that, compared to the null model, adding set size substantially improves model fit: response times are affected by number of distractors
    - Then adding effect of Diagnosis on intercept (vs.0) significantly improves model fit: stroke survivors respond more slowly than control participants do
    - But adding the interaction of set size and Diagnosis, i.e., effect of Diagnosis on slope (vs.1), does not significantly improve model fit: stroke survivors are not more affected by distractors than control participants are
- You can also use the summary() function to inspect the model
- You’ll notice that the fixed effects don’t have p-values
        - First, let’s just think about what those p-values would be: they are one-sample t-tests of whether the fixed effect estimate is different from 0, with the t-statistic equal to the estimate divided by the standard error of the estimate
    - The problem is that the degrees of freedom for that t-test are not simple to determine
        - The fixed effects are certainly free parameters, but the random effects are estimated under constraints (normal distribution with mean of 0)
        - However, the df *can* be estimated. The two most common estimations are Kenward-Roger and Satterthwaite and they are implemented in a few different packages
- One of the easiest to use is lmerTest: you just load the package and fit your model the same way, then the summary will contain Satterthwaite-approximated df and p-values
    - Here’s an example using the full model of the visual search data [scroll down to see the p-values in the summary]
- An important step in model evaluation is plotting the model fit
        - One way to do that is using the fitted() function. This works very nicely for overlaying observed data and model fits
- Another way is to use the effects package to get model estimates along with SE and confidence intervals.
- Then you can plot those values – this is particularly useful when you have complex models or missing data
- Some general advice
        - This semester you will be learning statistical methods that don't have "cookbook" recipes. You'll need to actively engage with the data and research question in order

to come up with a good model for answering the question, then to defend/explain that model.

- Practice is absolutely critical to learning how to do this. You can't learn it just from the lectures; you have to try it with real data. You will make mistakes, run into problems, etc. Identifying the mistakes and solving those problems is how you'll master this material.
    - We have made all of the example data sets and code available to you for exactly this reason.
- Come to our live Q&A sessions, do the lab exercises. If you're not sure, *try something* then try to figure out whether it worked or not. Ask questions when you're stuck -- we're here to help you learn, but it will only work if you engage in *active, hands-on learning.*

Week 2: Logistic Multilevel Modelling

- Last semester you learned about using logistic regression for binary outcomes
- A quick review: when there's a single binary outcome, it's intuitive that logistic regression is the right option. A common mistake is that this is still true if you have a bunch of binary trials and aggregate the results.
    - For example, if you're talking about accuracy or fixation proportion over a set of trials, 85% correct or 30% target fixation sound like continuous variables but they're not.
    - You can tell they're not because (1) they are bounded – proportions are never less than 0.0 or higher than 1.0, and (2) they have a very specific non-uniform variance pattern
- This can have real consequences for analysis. Here's an example study of a treatment effect – which group benefitted the most from treatment?
    - If you do a linear regression, it looks like the biggest benefits were in the moderate and severe groups, and the mild group had a smaller benefit.
        - But notice that the mild group is close to the ceiling. Linear regression doesn't know that, but logistic regression does and knows that variance is smaller near the ends.
    - In a logistic regression, the benefit for the mild and moderate group is approximately the same, and the severe group is the one showing a substantially larger benefit of treatment
- Ok, so logistic regression is a model of the binomial process that generated your binary data
    - To fit that model, it's not enough to know that overall accuracy was, say, 90%, you need to specify whether that was 9 out of 10 trials or 90 out of 100
    - You can specify this as a vector of individual 1's and 0's, in a more compact way as the number (or count) of 1's and 0's
    - The outcome that is being modelled is log-odds, also known as the logit, here is the formula for it, which you can see is different from the formula for proportions
    - When the probability is 50%, the log odds is 0. When the probability is less than 50, the log odds is negative; when the probability is higher, the log odds is positive. So log odds are not bounded at 0 and 1 the way proportions are
    - But there's a bit of a problem when proportions are exactly 0 or 1: the logit is undefined for those values (Inf), which makes it hard for logistic models to fit data with such extreme values. This is something to keep in mind if you're trying to model very rare outcomes – a different kind of model might be required.
- Let's look at an example: a word learning study comparing control participants and participants with aphasia
- There's a learning phase, then a test, then a follow-up 1 week later
- Let's just look at the test data for now – the test that was immediately after training and the follow-up 1 week later
    - We can ask whether the patients were less successful than controls were at learning these new words? (Lower test performance)
    - Did recall decrease from immediate test to 1-week follow-up?
    - Was retention (recall decrease) different for the two groups?
    - One might be tempted to use a 2x2 ANOVA (2 groups x 2 test phases)
        - Take a moment to think about what would be right about that approach and what would be wrong

    - What it gets right: group as a between-participant variable, Phase as a within-participant variable (MLM is a more flexible version of repeated measures ANOVA)
        - Phase-by-group interaction to test group differences in retention
    - What it gets wrong: ANOVA would treat Accuracy as a continuous linear variable, but it is an aggregated binary variable
- The good news is that fitting a logistic multilevel model is very similar to fitting a linear MLM, there are just three differences
        - You need to use the glmer() function instead of lmer() -- *generalised* linear mixed effects regression
        - Your outcome variable needs to be either a binary vector of 1's and 0's or two columns with paired counts of 1's and 0's – that's what we'll use here
        - And you need to add "family=binomial" as an option so glmer() knows which generalised linear model you're using
        - Here's what the model code looks like:
            - Use "cbind" (column bind) to make the outcome pair of columns. These values are each participants number of correct responses and number of errors in each test phase.
            - The fixed effects are test Phase and participant Group (the asterisk is a compact notation meaning both main effects and the interaction between them)
            - The random effects are by-participant random intercepts (implicit) and slopes – that means random by-participant variability in performance (intercept) and retention (slope)
            - The data – only the test phase data are included
            - And the family option
    - Note: logistic MLMs are slower to fit and are prone to convergence problems
        - This may require simplifying random effect structures (we'll talk more about that later in module)
        - Don't panic if you get convergence warnings or this singular fit message. These are not errors, the algorithm did produce a model, but it is telling you to carefully check your model and be cautious about interpreting the estimated parameters
- Looks like there is a significant effect of Phase -- a decrease in performance from the immediate phase to the 1-week follow-up
        - A significant effect of group – patients perform worse than controls
        - And no interaction
        - Note that these parameter estimates correspond to *simple* effects, not *main* effects: that test phase parameter is estimate for the control group and the group effect is estimated for the follow-up test phase
- When interpreting parameter estimates for categorical variables, it is important to keep in mind how the contrasts are coded. In R, the default is "treatment" coding, which produces simple effects as I just described.
        - This can be very sensible for some situations, like a treatment study: you get estimates for the control group (for example, is there natural recovery or a placebo effect or something like that) and baseline difference between control and treatment conditions (did your randomisation work properly), then the interaction gives you the differential effect of treatment

    - But in many of our studies, we want “main” effects. To get those, you need to use sum coding and you can specify that in the model code
        - These parameter estimates now correspond to the main effects of test Phase (across both groups) and Group (across both test phases)
- Plotting the model fits from logistic models can be a bit tricky, but conveniently the fitted() function returns proportions, so you can do something like this – the violins show the distributions of the observed data and the points show the model-estimated means
- I mentioned earlier that the computational demands of logistic models may require simplifying random effects. There are two key issues regarding getting the random effects structure right
        - When you have within-subject variables, omitting their random effect tends to inflate false positive rates for the corresponding fixed effect. So if you want to make inferences about a particular (within-subject) fixed effect, you’ll want to make sure you include the corresponding random effect
    - However, random effects require lots of data to estimate so it’s easy to over-parameterise a model with too many random effects. When you have convergence problems, there may be other fixed effect estimates that (nearly) as good as the ones you got, which is a problem if you want to make inferences based on the specific estimates you got.
    - So, as a general strategy, I recommend starting with a maximal random effect structure.
        - If it converges well, great.
        - If you have convergence problems, check the model summary random effect variance-covariance matrix and look for values that are very small or unrealistic – these are good candidates for removing to simplify the structure
        - You can also compare the fixed effect estimates (and SE) under different random effect structures – ideally, they should stay about the same. If they are substantially different, then you’ve got a fragile model and you should be *very* cautious about interpreting the results (and maybe consider a different statistical approach).
        - We’ll keep revisiting issues related to random effect structures each week, from slightly different angles and with increasing sophistication

Week 2: Longitudinal Data Analysis using Multilevel Modelling

- Longitudinal data are a natural application domain for MLM
    - Longitudinal measurements are *nested* within subjects (by definition)
    - Longitudinal measurements are related by a continuous variable, spacing can be uneven across participants, and data can be missing -- these are problems rmANOVA
    - Trajectories of longitudinal change can be nonlinear (we'll get to that next week)
  - We've already seen some examples of this:
    - the weight maintenance data from the Week 1 lab
    - the visual search example wasn't longitudinal, but the idea was the same
- Let's consider another example where we can think about longitudinal issues some more
    - These are Public Health England data on various mental health indicators. For this example, let's focus on county-level percentage of adults who are physically active at recommended levels
- The indicator ID for percentage of physically active adults is 90275
- Here's a plot of the data by region of England from 2012 to 2015
    - We can ask whether the baseline rates differ and whether the slopes of the change differed during this time window
- To answer the first question, we need to check what we mean by "baseline"
    - The intercept coefficient could answer this question
  - But those will be estimated at Year=0 and the question is not about whether adults were physically active the year Jesus was born.
    - We need to adjust the time variable so that 2012 corresponds to time 0 (and we can select just the physical activity values while we're at it)
    - Now Time is a variable just like Year, but going from 0 to 3 instead of 2012 to 2015
- Now we can fit the models
    - The base model just has an overall effect of time, and by-county random intercepts and slopes
    - Then we can add baseline differences between regions
    - And the full model will have an effect of time, differences between regions, and region-by-time interaction (that is, slope differences between regions)
    - Notice that, as usual, the random effects remain the same across all models, we're only changing the fixed effects
- The model comparison reveals that there were significant baseline (2012) differences between regions in terms of % of physically active adults, but not in the slope of change over the next 3 years
    - When doing the model comparisons, the degrees of freedom are a good quick check: df=8 here, which makes sense because there are 9 regions. So when we add a region effect (or the region-by-time interaction), one region becomes the reference and the model estimates 8 additional coefficients for each of the other regions
- We can plot the model-estimated trajectories using the effects package, like we've done with previous examples
- Another option is to plot the parameter estimates themselves. I think this approach gets used more in epidemiology and biostats than in psychology, but it can be quite useful
    - Side note: it took me a while to work out the data wrangling for pulling the estimates from the model and setting them up to make this plot
    - I spend at least half of my analysis time doing data wrangling and it's often closer to 80%. That's not a major part of this course, but it will be once you're analysing your own data.

Week 3: Longitudinal Data Analysis using Multilevel Modelling – Nonlinear Change

- Last week we talked about using multilevel models for longitudinal data analysis, now let's talk about how to deal with non-linear change over time
- All the good things about multilevel models are still relevant, now we're just focusing on modelling non-linear trajectories
    - This application of multilevel models is sometimes called "growth curve analysis"
- Here's an example that's a little more complicated than the ones we've looked at so far: these are data from an eye-tracking experiment where participants had to pick the picture that matched a spoken word.
        - They were faster to look at targets for high frequency (more familiar) words than for low frequency (less familiar) words
        - You can see there's a sort of S-shaped curve here, so how are we going to model this non-linear trajectory?
- To start, we have to choose a "functional form" – a mathematical function to describe that shape. To do that, we need to consider three factors:
        - The function must be adequate to the data – it must be able to take that shape
        - The function needs to have a property called "dynamic consistency"
        - And we need to think about what kind of prediction we want to do
    - It turns out that polynomials offer a pretty good solution to this problem, especially orthogonal polynomials. That's the approach I'll focus on in this lecture, but first let's talk about those three factors in a bit more detail
- The function must be able to take that shape – if your data have this kind of U-shape, a straight line isn't going to do it.
        - A quick way to check how well your function is fitting the data is to plot the residual error against the fitted values – the residuals should be about evenly distributed around 0 (how the red dots are).
        - If you see consistent deviations (as in the black dots), that means your function is missing some consistent aspects of the data
    - You can generate that plot quickly using the augment() function from the broom package (actually, for multilevel models, you need the broom.mixed package)
- Now let's talk about dynamic consistency. It will help to understand that there are two kinds of non-linearity
        - First is non-linear variables or predictors: for example time-squared – that's non-linear (quadratic) time, but notice that that the regression equation has the same form as a typical linear regression – the beta coefficients that are being estimated are still linear
        - Things are different when the parameters (the values that the model is trying to estimate) are themselves non-linearly related to the outcome, as in this equation
    - Dynamic consistency is when the model of the average (that's the group-level model) is equal to the average of the individual models
        - In multilevel models, the random effects (which correspond to individual-level models) have a mean of 0, so the average of those individual-level models will necessarily have 0 deviation from the group-level fixed effect estimates. In other words, multilevel models are always dynamically consistent

    - But this isn't always true once you go outside the multilevel modelling framework. If you fit a complex non-linear function to individual participants' data, then average those models together, you're not necessarily going to get a model of the group data
- Here's an example of what I mean:
        - the blue dots are the group-level data and the blue line is a model of those data – it fits nicely
        - the grey lines are individual participant models, they also fit well
        - but if I average the parameters of those individual participant models and plot it, I get the red line – that's not the group-level model and doesn't fit the group-level data
        - this is what lack of dynamic consistency looks like
    - and it's a big problem for statistical inference: if I compare the individual-level parameters from the left panel to the ones in the right panel (using something like a t-test for example), I'll be testing whether the *red lines* are different but those lines don't correspond to anything interesting
    - so if you fit non-linear functions to individual participants and try to compare the central tendencies of their parameter estimates, you'll be making non-sensical inferences
        - it is hypothetically possible to implement non-linear functions within a multilevel modelling framework, though I am not aware of any good packages for doing this
- It's maybe useful to step back and think about why we even use statistical models and how they are related to making predictions
        - One of the simplest statistical models is the mean and standard deviation
        - The point of them is to provide a compact quantitative description of the data
        - They're not making any new predictions and they're not falsifiable – they just describe the data. A mean can't be "wrong", though it can be useful or not so useful depending on the data
        - A good statistical model provides a useful quantitative description of the data
    - Useful for what? That's the role of the theoretical models. Theoretical models are how you make new predictions and they can be wrong. In fact, a theory being wrong – being falsified by the data – is an important step in scientific discovery.
        - Theoretical models can be quantitative, as in mathematical psychology or computational cognitive modelling, but they're still doing a different thing from a statistical model
        - The statistical model and the theoretical model need to be connected – the statistical model should provide information for testing the theoretical model – but they are not the same thing. And that's a good thing because the data can inform different theoretical models
- Polynomials are a pretty good solution to these different constraints
        - They can model any smooth curve shape (this is closely related to Taylor's theorem)
        - They are dynamically consistent
        - Their main weakness is that they are bad at capturing asymptotic behaviour. In general, this isn't too bad because we usually care about the changing part of the data, not the asymptotic part. It helps if you can avoid modelling long flat sections (the asymptotes). And it is important not to try to extrapolate from polynomial curve fits.
    - If you're going to use polynomials, you'll need to choose a polynomial order – quadratic, cubic, fourth-order, whatever

        - With some experience, you can just look at the curve shape and have a pretty good guess about the right order
        - You can also take a statistical approach: include terms that improve the model fit
        - Alternatively you can use a theoretical approach: your predictions will typically be about particular kinds of effects so you can focus on polynomial terms that correspond to those effects
- One more thing about polynomials: the terms of a natural polynomial are usually correlated and on different scales – this means that you'll have collinear predictors and can have estimation problems (left figure)
        - Orthogonal polynomials are just a transformation that centres and scales them, so they are uncorrelated and on the same scale
- You do need to be careful about interpreting orthogonal polynomials: the intercept corresponds to the overall average rather than the y-intercept
- The linear term corresponds to the linear slope, with the pivot point in the centre
        - The quadratic term corresponds to the steepness of the rise and fall rate, which is useful for modelling U-shapes
        - The cubic and quartic terms also correspond to steepness around inflection points, though it gets hard to interpret those higher-order terms
- A quick reminder about random effects: we've talked about random intercepts and slopes, now that we're dealing with polynomials we're going to have random "slopes" for those polynomial terms
    - And following the strategy I described before, we'll start with a maximal random effect structure and simplify it if we run into convergence problems
- Ok, so now we have some ideas about how to model the data in our eye-tracking data example
- To start, let's create a third-order polynomial – the code_poly() function is a helper function for doing this. You give it your data frame, identify the predictor variable that needs to have polynomial versions, and the polynomial order you need. By default, it will make a graph that shows you how your raw predictor is related to the polynomials it created. (This function defaults to using orthogonal polynomials, but you can specify natural polynomials if you want those)
- You can see this added three variables ploy1, poly2, and poly3 to the data frame
- I've skipped ahead to fitting the full model, but you can build up the Condition effects if you want
        - The fixed effects are the three polynomial time terms and their interactions with Condition
        - Then we have random effects of Subject with random "slopes" for each of the polynomial terms. This allows individual subject variability in each of the curve shape components
        - And we have analogous random effects for subject-by-condition, since condition is within-subject (I'll come back to why I specified them this way in a minute)
        - Here are the fixed effects: you can see there is a significant Condition intercept effect and quadratic effect
- And you can plot the model fit – an important step when fitting complex models to make sure the fit is good and the inferences you want to make are supported by your model
- Now let's talk about the random effects. We specified two sets of random effects: subject-level and subject-by-condition level, for each time term. You can see values there: these would be the zeta values from the equations we talked about before
- In the model summary, what we see is the variance (standard deviation here) and covariance of these random effects – these values are being estimated and the individual unit-level random effects on the previous slide are constrained to have this variance-covariance structure with a mean of 0

    - This is why degrees of freedom in multilevel models are tricky to define – you have these inter-related parameters to estimate so they're not exactly "free"
- There's another aspect of the random effects I want to discuss. In the full model before, I used two random effects (subject-level and subject-by-condition).
        - Another approach would be to have random by-condition slopes of Condition; putting Condition on the left of the pipe. This would be a more straightforward extension of the random effect structures we used in the first two weeks
        - You can see it would produce virtually the same result (somewhat higher p-values, but the same result – significant intercept and quadratic effect of Condition)
- Using this approach has some advantages – it allows more flexible variance and covariance estimates
- But at the cost of more random effect parameter estimates. With two conditions (high vs. low) and three time terms, the difference is not too bad. But if you have more conditions (like a 2x2 design) and more time terms, the difference can be huge
        - In general, I think it's worth trying this left-side random effect structure. In practice, I often have data sets that can't support that level of complexity (flexibility) in the random effect structure and the other approach offers a reasonable way to simplify it without losing the most important elements
- When you run into convergence problems, polynomial models offer you several different options for simplifying the random effect structure
    - One option is to remove a higher-order polynomial random effect. For example, if I'm not too interested in the cubic term, I might be willing to make the simplifying assumption that subjects do not vary in that term
    - Another option is to remove the random effect correlations. This can be done manually by writing out each effect or by using a double-pipe notation
- This is also a good time to talk a little more about what the random effects are doing
        - When we treat participants as random effects, this captures the typical assumption that the participants are a random sample from some population and we want to generalise to that population
        - This graph (created by Tristan Mahr) shows three ways of thinking about individual variation:
            - "complete pooling" is one model for everyone. That's what you get from OLS regression and all of the residual error is just treated as the same kind of error
            - "no pooling" is each person has their own completely independent model. I've heard this called statistical "amnesia" because each participant's data is fit without any memory of the other participants. Now it's true that each participant is different from the others, but they're not completely unrelated – they come from the same population after all
            - "partial pooling" is what you get in multilevel models. Each participant has their own random effect, so they are different from others, but those random effects are constrained to come from one distribution (that is, participants are sampled from some single population)
- This produces an effect called "shrinkage": the individual participants' estimates "shrink" toward the group mean. This is a very useful property because each individual's performance is a mix of general population-level properties, their individual unique properties, and noise. We usually care about the population-level and individual-level stuff, but we can't know exactly how much noise is

contributing. But, we can use the performance of everyone else in the sample to *estimate* how much noise might be happening and what might be legit individual differences. This is exactly what partial pooling or shrinkage accomplishes
        - I recommend reading Tristan Mahr's and/or Michael Clark's explanations of these concepts to understand them better
- Keep in mind that the benefits of partial pooling and shrinkage depend on the assumption that the participants are sampled from some population and you want to generalise. That's not always the case.
    - For example, in neurological or neuropsychological case studies, you're trying to characterise the performance of each participant and make claims about that specific participant, not some population of participants.
    - In those cases, it can make more sense to treat participants as fixed effects
- Key points: this lecture covered two broad issues
    - Modelling non-linear change: I described some general considerations and focused on using polynomials. Polynomials have some very nice mathematical properties, but you have to be careful with interpretation and extrapolation when you use them
    - Random effect structure: as a general rule, you want to start with a maximal random effect structure and simplify it if needed.
        - Three simplification strategies: Putting categorical variables on the right side of the pipe, removing random correlations, and removing higher-order terms
    - The discussion of random effects also brought up the key concepts of partial pooling and shrinkage, which are important for understanding how multilevel models distinguish group-level effects, individual-level differences, and noise.

Week 4: Three-level Nesting

- This week we will focus on more complex random effect structures. In this lecture, I'll show you how to handle three levels of nesting
- So far, the data we've analysed had two levels of nesting: there was a group level and an individual participant level, with multiple observations for each participant
    - It's also possible to have more levels of nesting or other nesting structures, all of which can be captured by specifying the random effect structure
- Let's consider three-levels of nesting: this is a hypothetical treatment study where each therapist administers a control or treatment condition to multiple subjects who are tested at multiple time points.
    - So, working from the bottom up, we have observations at different time points nested within subjects, and subjects nested within therapists. The group (treatment vs. control) is between-subjects (and between-therapists)
- Here's another example (with simulated data): imagine we have a new computer-based active learning method and we want to know if it improves scores on a math test
    - We measured students' math scores and proportion of time spent using the computer-based active learning software
    - This program was implemented in 3 schools, 8-12 classrooms per school, and 12-24 students per class
    - So students are nested within a classroom (those student data are not independent) and classrooms are nested within school (those classroom data are not independent)
- The data are stored as a CSV file and here I'm doing a little data wrangling when I read them in:
    - Creating a unique class ID (so classes don't get confused across schools)
    - Centering time spent on the computer relative to the overall mean
    - Centering class size relative to the overall mean and the school mean
    - Recall: the point of the centering is so that we'll be estimating meaningful intercept parameters rather than estimating for class sizes of 0 or no time spent on the computer
- Once the data are read in, we can make some exploratory plots to get a sense of what is going on
    - Looks like math scores are better in smaller classes (left)
    - And for students with a higher active learning proportion (right)
- Two more things we need to consider:
    - Left: Classes differ in size, and this is unequal across schools: school 1 tends to have smaller classes and school 3 tends to have larger classes
    - Right: Students differ in proportion of time spent on active learning, but each class has a big range and the ranges seem fairly similar across classes
- Let's start with a really simple model: math score as function of active learning time and class size, with random intercepts by class (that is, classrooms have vary in their average math score)
    - Both main effects are significant and the interaction is marginal
    - Keep an eye on the effect of class size and its interaction with active learning time
- If we look at the model fits, we can see offset parallel lines for the classrooms – they had random intercepts but not random slopes for the effect of active learning
    - To allow that variability, we need to add random slopes
- In this model, the fixed effects are the same, but we've added by-class random slopes of active learning time

        - Adding those random slopes substantially increased the standard errors and reduced estimated degrees of freedom for Active Time fixed effects. The main effect was very strong, and it still is, but the interaction is not significant any more
    - I said before that omitting random slopes makes the corresponding fixed effects anti-conservative: this is an example of that pattern
- When we plot the model fits we can now see random class-level variation in the relationship between math scores and active learning (that is, the lines have different slopes)
- This 2-level model assumes that classrooms are independent, but this is not quite true. Classrooms are nested within schools and there may be school-level differences in math scores, effects of active learning, class size, etc.
    - If we split up the data by school, we can see that there really might be school-level differences and those aren't anywhere in the 2-level model
- There's a specific issue here to be concerned about: school and class size are confounded. What appears to be a class size effect, might actually be differences between schools.
    - A class size effect would be interesting – that's something we might try to generalise to other classes.
    - But it's hard to know what differences between schools mean, especially when we only have 3 schools and don't know anything else about them
    - One way to deal with this is to use school-mean centered class sizes and control for overall school-level differences
- This confound is related to an important statistical paradox called Simpson's Paradox: you can get an overall effect even when each of the sub-groups shows *the opposite* effect. This happens when there is a confounding variable. A famous example is a study of gender bias in admissions at UC Berkeley in the 1970's
    - In multilevel modelling contexts we need to be particularly careful about sub-group (that is, cluster-level) differences that might be confounded with the variables of interest
- Coming back to our model, here's how we can capture three levels of nesting: we use the school-mean-centered class size and we have both school-level and class-level random effects
- And now the class size effect is no longer significant (not even close). Looks like school differences were masquerading as class size differences – a Simpson's paradox type of effect
- There is one more thing to worry about: there were only 3 schools, so how is the model going to estimate those school-level random effects
    - Looks like it had some trouble: the by-school intercept-slope correlation is -1.00, which is almost certainly untrue
    - We can use the double-pipe notation to remove that mis-estimated correlation: you can see we still have school-level random slopes and random intercepts, but not the correlation between them
- That didn't make a big difference for the fixed effects – class size still not significant, interaction still marginal. If it *had* made a big difference, I'd be worried about the overall model, but this way I'm feeling pretty confident about these results
- To sum up the key points from this lecture
    - The nested structures we've talked about can extend to three (or more) levels and this can be captured by the random effects structure
    - This is particularly important when sub-group (cluster-level) differences might be confounded with variables of interest (Simpson's Paradox).

- We saw a concrete example of how omitting random slopes tends to make models anti-conservative (inflates rates of false positives). My general recommendation is to start with a "full" or "maximal" random effect structure and reduce as needed
- We saw an example of that when we had only a few observational units (only 3 schools) and estimating the random effects is very difficult. This can produce unreliable fixed effect estimates. Over-parameterised random effect structures should be simplified and we saw how to identify a (probably) mis-estimated random effect and remove it from the model.

Week 4: Crossed Random Effects

- This week we are focusing on more advanced nesting structures and how to capture them with random effects. In this lecture I will cover crossed random effects
- In pretty much all of the examples you've considered so far, there was variability between subjects and you wanted to evaluate the reliability of some effect in the context of that variability
    - We can ask an analogous question about variability between test items. This often comes up in laboratory studies where participants solve some problems or answer some questions.
        - We could average across those test items to get a subject mean, but that inter-item variability might be important
        - The inferences we want to make are usually intended to generalise to other items of the same general type so we have the same problem of quantifying the effect relative to inter-item variability
    - Historically, this was done by conducting separate by-subjects and by-items analyses, which were called "F1" and "F2" tests. And journals sometimes even required this (a long time ago, I had a paper rejected because a key result was statistically significant by subjects but only marginal by items).
    - Multilevel models offer a better solution to this problem
- Here's an example using simulated data for an intervention intended to improve problem solving ability. 120 participants were randomly assigned to either the Treatment or Control condition, then solved 30 problems (16 hard ones and 14 easy ones)
        - The outcome we care about is the response time (RT) for a correct solution. Note that there are missing data because if the participant failed to solve the problem, there is no RT
- If we were going to do a traditional kind of analysis, we'd take these data and calculate by-subject means for each problem type
    - Looks like everyone solves Easy problems faster than Hard ones, and
        - The Treatment group seems faster at problem solving, esp. for Hard problems
- Then we'd run a by-subject repeated measures ANOVA, which we can do with the afex package
    - Looks like there is an overall problem difficulty effect, and no effect(s) of the intervention.
    - But hang on: not all word problems are the same and we're going to make inferences about solving problems of this type, not just about solving these particular problems
- So we can do the analogous by-items thing: calculate item means
- And run a by-items repeated measures ANOVA – in this analysis all of the effects are significant. Why the difference?
- Problem type (difficulty) had a large effect, and that effect was significant in both analyses
        - The effect of Condition (the intervention) and its interaction with problem type were small.
    - Condition is between-subjects but within-items, so the between-subjects variability is strong in the by-subjects analysis but gets averaged away in the by-items analysis.
        - This makes the by-items analysis look overly strong (subject variability is missing) and the by-subjects analysis look overly weak (items consistency is missing).
        - This kind of thing comes up a lot because we can try to improve power by designing studies to be within-subjects or within-items, but it's often impossible to do both.
    - Also, the idea that effects should be significant in separate by-items and by-subjects analyses (aka F1 and F2) is generally thought to be overly conservative.
- Multilevel models offer an alternative approach because they can simultaneously model random variability at subject and item levels, as well as the group-level effects that we are interested in.

        - The key observation is that data are nested (or clustered) both within-subjects (each subject solved a set of problems) and within-items (each problem was solved by a set of subjects). These are called “crossed random effects”
        - In the model specification we have by-subject random effects (with random slopes of problem type, since that was within-subjects) and by-item random effects (with random slopes for Condition, since that was within-items)
- When we do the analysis this way, each of the effects is significant
        - Though notice that they are not as strong as in the by-items ANOVA (which was overly strong) nor as weak as in the by-subjects ANOVA (which was overly weak)
        - This analysis is capturing both the variability and the consistency across these different levels of nesting.
- It can be tricky to plot the results of a crossed random effects analysis because either by-subject or by-item averaging can mis-represent the data. This is a case where the effects package is very useful – you can get the model-estimated means and confidence intervals
- and plot those: here I’ve constructed something like a box plot with a marker for the mean, thick lines for +/- 1 standard error, and thin lines for the 95% confidence intervals
- This week we covered some more complex random effect structures: 3-level nesting and crossed random effects. Although the random effects are more complicated, the other aspects of multilevel models that you learned in prior lectures also hold here:
        - p-value estimation was done using Satterthwaite method; model comparisons would've been a good alternative
        - start with a full or “maximal” random effect structure and can simplify if model doesn't converge by removing correlations and random "slopes"
        - be aware of how your categorical variables as coded; can conduct pairwise comparisons using a single model
        - use logistic regression for binary outcomes

Week 5: Individual Differences, part 1

- As psychological scientists, we're interested in the fundamental principles of how the mind works. Often, that means thinking about a hypothetical "average" mind, but individual differences can provide additional, unique insights into the mechanisms or principles that we're studying
    - For example, you might find that a treatment works better than some control or placebo. But it invariably works better for some people than others – understanding why that is tells us something about how it works and the condition it's treating
        - Or we might have a problem solving task where people solve the easy problems faster than hard problems. Understanding why some participants find the easy problems *a lot easier* than the hard problems tells us something about the problem solving mechanisms
    - Methods like t-test and ANOVA are all about that central tendency and treat individual differences as noise, which reinforces this idea that we're interested in the behaviour of a hypothetical average individual
        - Multilevel models provide two ways to quantify and analyse individual differences, which will be our topic this week
- The first (and more straight-forward) way is for dealing with individual differences that are "external" to your study. Essentially, these are just additional properties of your participants that you measure in addition to whatever you're studying
        - Here's an example: let's say you're interested in tolerance for deviant behaviour, which you study by asking participants whether it is ok for someone their age to break various rules (cheat on tests, use drugs, steal things, etc.).
        - In addition, you've assessed some other variables: exposure to this kind of deviant behaviour and gender
- You can see here that as kids get older their average tolerance for this kind of behaviour increases, and this seems to be more so for males than for females
- As a starting point, we can ask group-level questions: does tolerance increase with age and is it modulated by gender?
        - In the model we have fixed effects of age, gender, and their interaction
        - And random by-subject intercepts (participants can have different baseline tolerance levels) and slopes (participants can have different rates of increased tolerance)
        - And we've sum-coded gender to get a main effect of age
        - The results shows a significant effect of Age, and no effects of gender
- Now an individual differences question: is this modulated by exposure to deviant behaviour?
        - I don't know the exposure scale here, but the range was about 0.8 to 2, so let's center it to make the other estimates easier to interpret
    - Now we can fit a model that is like the one we had before, but with the addition of this centred exposure variable
- In this model we see significant main effects of age and exposure
        - And significant interactions with exposure: Age-by-Exposure, Gender-by-Exposure, and a three-way interaction of Age-by-Gender-by-Exposure
- I can usually guess what a two-way interaction will look like, and in any case, that's relatively easy to plot and see what it looks like. A three-way interaction is trickier to plot, especially this one, which is a relationship among four variables (Tolerance for deviant behavior, Exposure to deviant behavior, Age, and Gender), and three of those are continuous variables.

        - To make it easier to visualise, we can split exposure into levels as if it were an ordinal variable
    - We can make a median split like this
- Then plot it like this, including both the observed data and the model fits: looks like adolescents with higher Exposure to deviant behaviour tend to have increased Tolerance for deviant behaviour as they get older, and this is stronger for males than for females.
- Median splits are ok, but often I find that a tertile split – so you have a "High", "Medium", and "Low" group – provides more information while still being relatively easy to interpret
        - Here's some code for dividing the exposure variable into three equal groups
- And here's a plot of that: looks like that interaction is really being driven by the high-exposure males
        - A word of warning: if you use this sort of visualisation strategy, reviewers and readers may get confused about whether your model used continuous or categorical predictors, so you'll need to be extra clear about this in your write-up.
- That was a little digression into data visualisation and interpreting interactions
        - The key points for this lecture were that individual differences provide deeper insights into group-level phenomena and you can assess them by adding them to the multilevel model

Week 5: Individual Differences, part 2

- The first part of this week's lecture was about "external" individual differences that could just be added as fixed effects to a model. What happens if you're interested in quantifying individual differences that were internal to your study?
- This might happen if you don't have an external measure, or you need to extract individual differences for another analysis (this comes up in my work if I want to assess neural correlates of some individual difference)
    - Random effects provide a way to do this, here's a simple example
    - Imagine you have two participants (A and B) in two conditions (0 and 1). The dashed lines indicate the condition means, those would be the fixed effects
    - The zetas show the individual deviations from those means, those are the random effects
    - Now for participant A, we can we can take the difference between the random effects: that's 1 minus negative 1, which is positive 2
    - For participant B, the same subtraction gives us negative 1 minus 1, which is negative 2
    - So this quite nicely captures the fact that participant A had a larger-than-average effect of condition while participant B had a smaller-than-average effect of condition
    - Notice that, because random effects are deviations from the average, these individual differences are relative to the average effect size, so they'll have positive (larger) and negative (smaller) values even if everyone shows an overall positive (or negative) effect
- That was a very simple example, let's look at how it would work in an actual study. In this simulated data set, we're looking at the effect of a school mental health intervention on educational achievement (math scores)
    - Condition is whether the kids were assigned to receive mental health services or not
    - SDQ is a mental health screening which was given to the treatment group – lower scores are better (less mental health difficulties)
    - Math is their score on a standardised math test
- Here are the data: looks like all kids math scores improved and maybe the treatment group improved faster
    - First, we can ask the group-level question: did the math scores improve more for the treatment group than the control group
    - If there's a group-level effect, then we're going to infer that mental health intervention facilitates academic achievement. If that's true, then the kids who had the most mental health benefit should have the most improvement in math scores -- that's an individual differences question that could provide further support for our inference
- Let's start with the group-level question
    - Like before, we can adjust the time variable to have a sensible intercept
    - Start with a base model that just has an overall effect of time and random by-subject intercepts and slopes
    - Add a baseline effect of treatment condition
    - Then fit the full model with time-by-condition interaction
- When we compare the models, we see there's no significant effect of condition at baseline (the randomisation worked), but there's an effect of slope: the groups started out about the same and the treatment group's math scores improved more rapidly

- The parameter estimates tell basically the same story: no significant effect of condition at baseline and a statistically significant time-by-condition interaction
    - We're taking this to mean that mental health intervention facilitated math learning. As I said, if that's true, then the kids with the most mental health benefit should show the biggest math score improvements
- That's the second question -- individual differences. We can start by just making a plot of individual differences in mental health change. Looks like there was a lot of variability: some people responded really well (big decreases in difficulties on SDQ), some people didn't respond well (increased difficulties according to SDQ).
    - To answer the individual differences question, we want to quantify individual differences in SDQ slopes and math learning slopes, then see if they are related
- Here's the analysis strategy:
    - First, we build separate models of change in SDQ and change in Math scores
    - Then we use the random effects estimated in those models to quantify individual differences in those slopes
    - Then we test the correlation between those individual differences
- Ok, here are the models: pretty simple ones with just the outcome as a function of time, with random by-subject intercepts and slopes. Remember that only the treatment group got the SDQ mental health assessment, so we're fitting these models for just the treatment group's data
- Now we extract the random effects
    - The get_ranef() function is a little helper function I wrote that will extract a random effect from a model and clean it up a bit so it's easier to analyse
    - The merge function combines them into one data frame, aligning the data by ID and giving the math and sdq random effects sensible variable names
    - You can see in the summary there's an ID variable, each participant's random intercepts and slopes for math scores, and random intercepts and slopes for SDQ
- Now we can just see if those random slopes are correlated: indeed they are, r = -0.77. That's a strong negative correlation indicating steeper rise in math scores was associated with steeper decrease in mental health difficulties
- Why bother with this random effects business when we could've just fit models for each individual participant and gotten their slopes that way?
    - An individual's performance (on the math test, on the SDQ) is their actual level plus some noise.
    - Individual models (no pooling) don't make that distinction, so you have a noisy estimate of individual differences.
    - Multilevel models reduce the noise component by using the mean and variance of the rest of the group – they *shrink* (or partially pool) the individual estimates toward the group mean based on the distribution of the other individual estimates. This produces a better estimate of true individual differences.
    - See also: Stein's Paradox
- Key points
    - The broad conceptual point from this week is that individual differences provide additional insights into phenomena of interest. You can use them as further tests of a hypothesis
    - When you have a group-level phenomenon or model, you can use the random effects from that model to quantify individual differences
    - Partial pooling / shrinkage improves individual difference estimates