Introduction to Multilevel Modelling using R

UCL ICH STATISTICS COURSES

George Michaelides - g.michaelides@bbk.ac.uk
Department Of Organizational Psychology - Birkbeck, University of London

About this course

- Session 1: Introduction to multilevel regression
- Session 2: Estimating Multilevel Models in R
- Session 3: Longitudinal models
- Session 4: Aggregation Testing: Agreement and Reliability
- Session 5: Advanced Models

Session 1: Introduction to Multilevel Regression

What is Multilevel regression

- An extension to linear regression
- Allows for the analysis of data that are structured in more than one level

Multilevel by any other name...

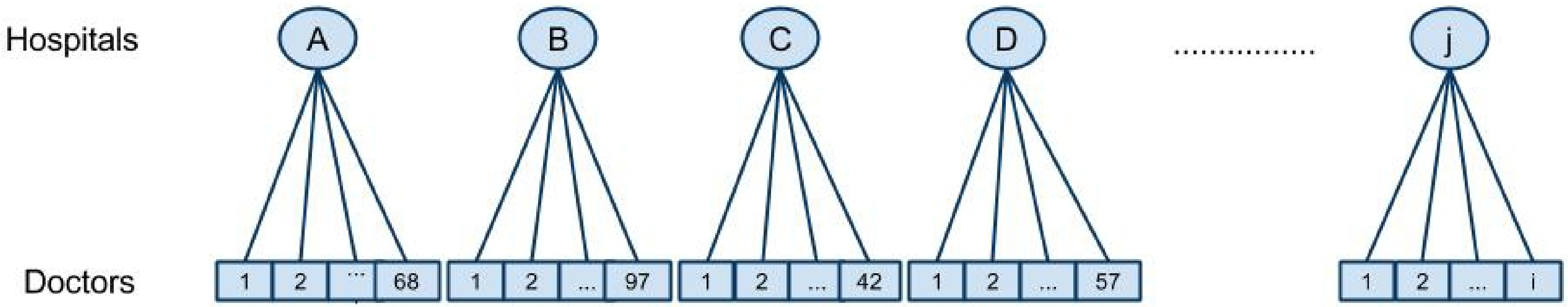
- Multilevel models
- Hierarchical Linear Models (HLM)
- Linear Mixed Models (LMM)
- Mixed Effects Models
- Random Coefficient Models

…or by any other software…

- R (multilevel, nlme, lme4)
- SAS (PROC MIXED)
- Stata
- S-Plus
- MPlus
- SPSS (Linear Mixed Models)
- HLM
- MlWin

Why? Analysis of structured data

- Data can be hierarchically nested
  - Employees in departments, departments in organizations, and so forth
  - Students in classrooms, classrooms in schools
  - Doctors in hospitals



Why is this important

Assumptions of regression

- Linear regression could violate assumption of independence of errors
- Using ANOVA or dummy variables with many groups reduces power and parsimony
- Performing multiple regressions could be difficult to interpret

Model context

- Social context is important even if not part of our theory
- Accounting for the nested nature of data we explore and control for the possibility that social context matters
- By including higher order variables in a model we can testing whether some contextual characteristics matter

Why is this important

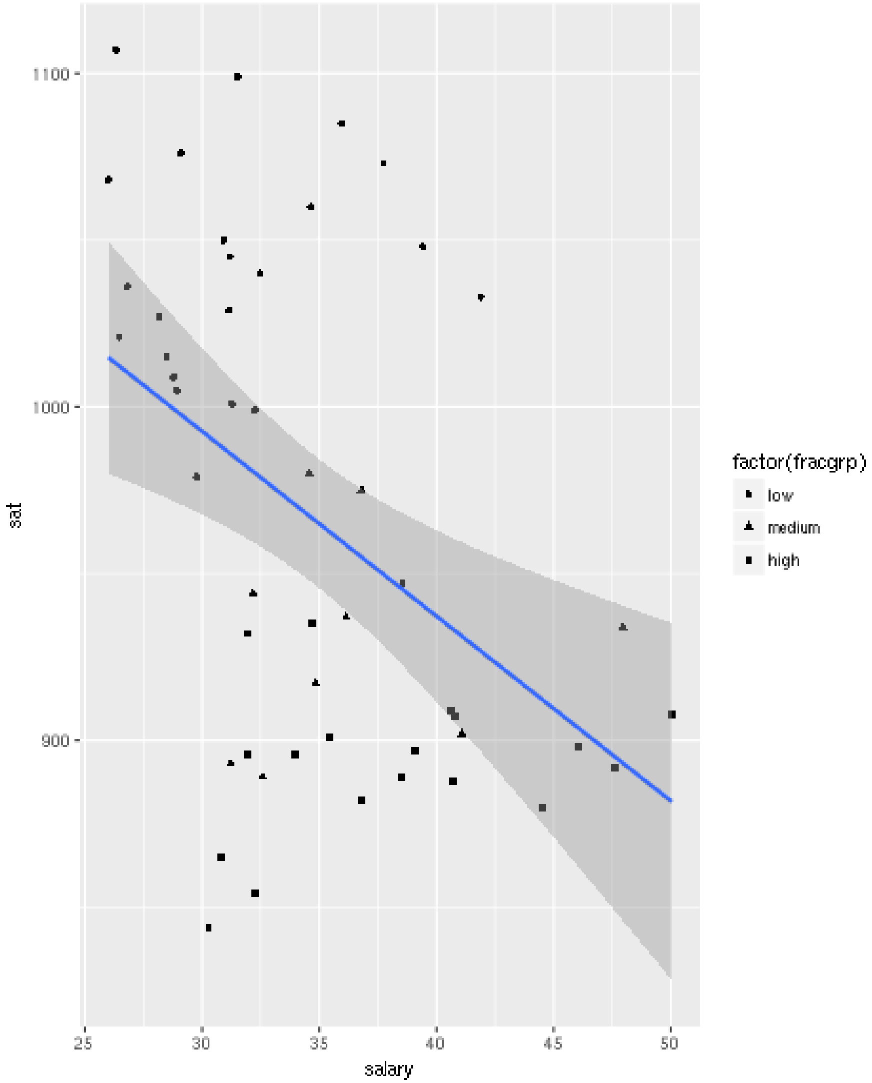
Model Time

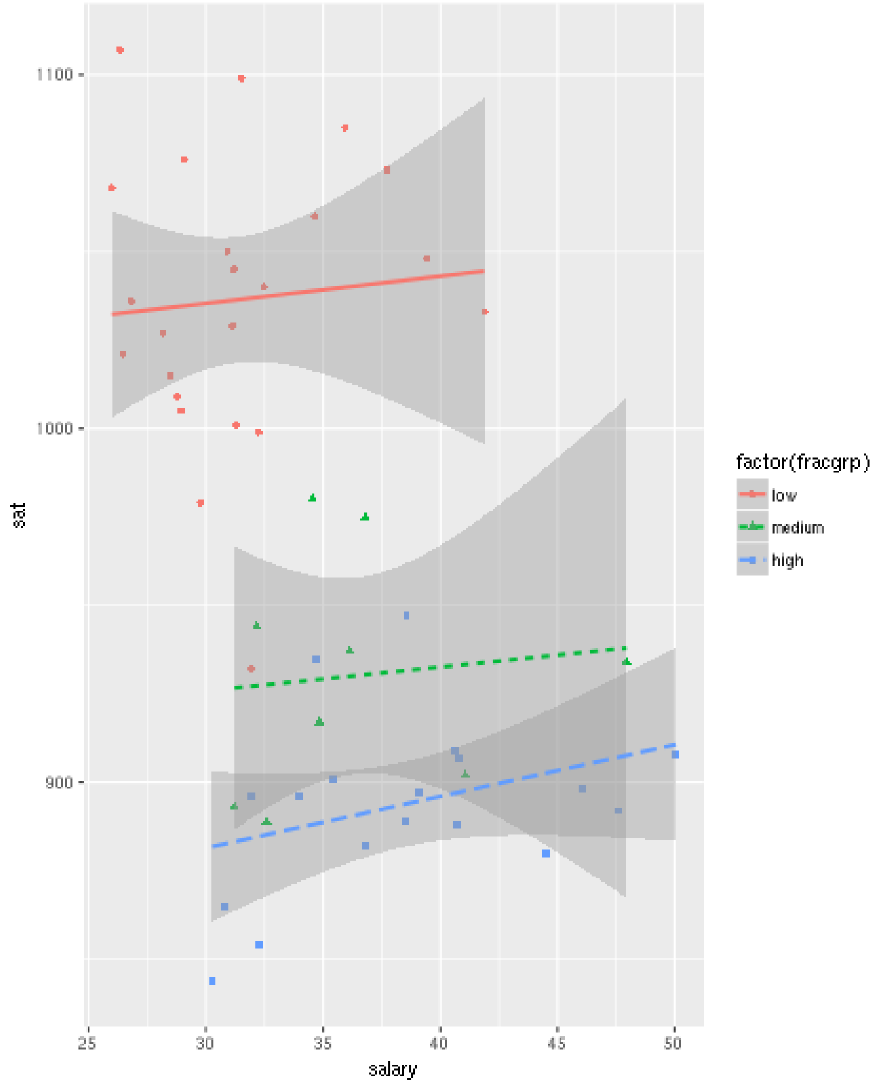
- Dynamic models can be seen as a variation of contextual models
- Repeated measures are a lower level and individuals is the higher level variable
- This allows modelling how individual characteristics relate to different starting values and subsequently can affect changes over time

Match theory and data

- Ecological fallacy
  - "…drawing inferences at the individual level based on group-level data" Diez-Roux (1998)
- Atomistic fallacy
  - "…drawing inferences at the group level based on individual-level data"
- Simpson's paradox suggests that
  - erroneous conclusions can be drawn if grouped data, drawn from heterogeneous populations are collapsed and analysed as if from a single population

Simpson's paradox





Why is this important?

Correctly specify models

- Disaggregation of data to lower level is problematic
- Regression approach results in
  - biases because of correlated errors (Violation of independence)
  - danger of the ecological fallacy
  - over-optimistic estimates of significance
- ANOVA approach
  - may lead to too many parameters (reducing power and parsimony)
  - fixed effects ignore random variability associated with group level characteristics

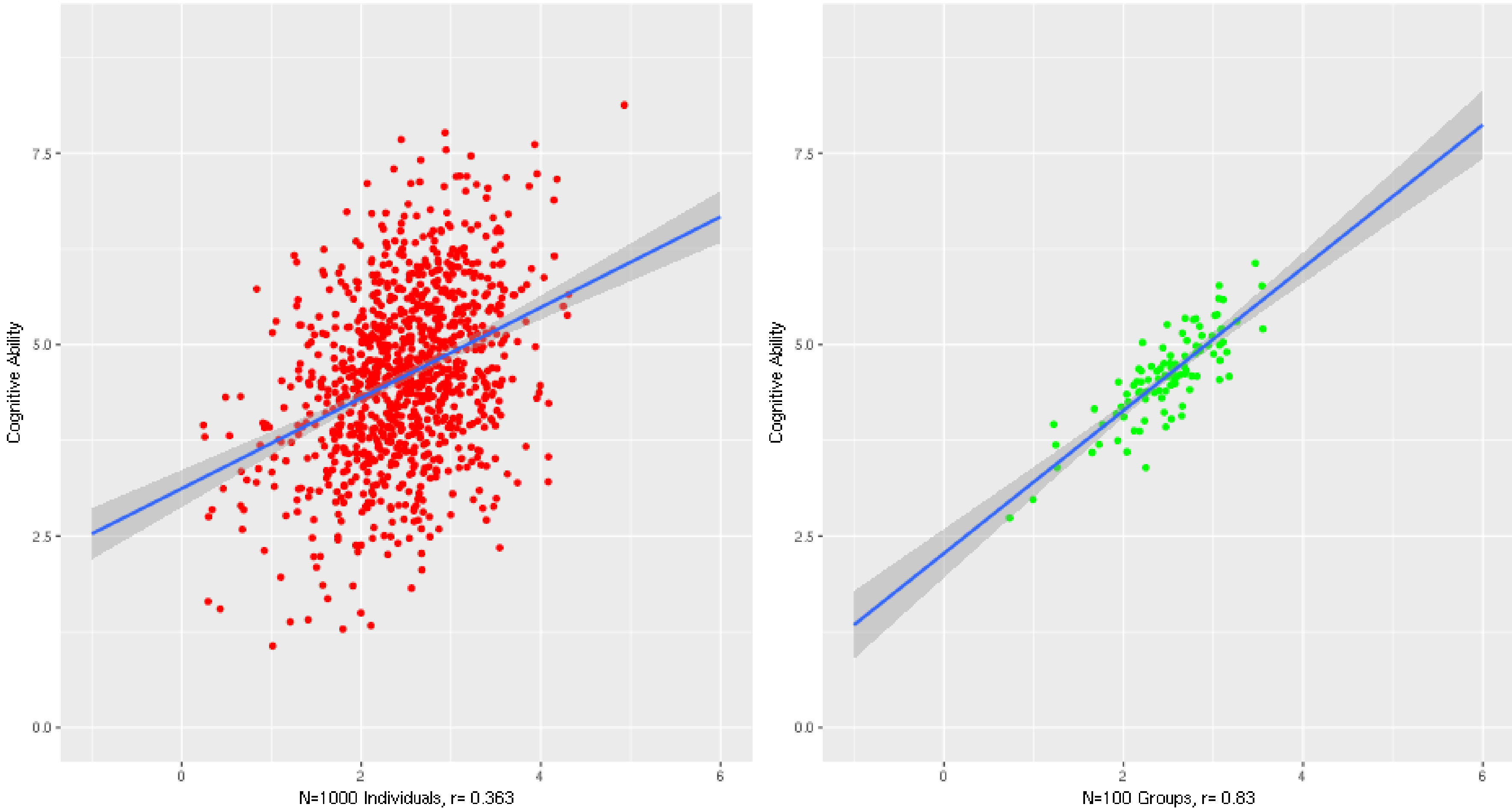
Why is this important?

Correctly specify models

- Aggregation of data to higher level for regression is also problematic fewer units of analysis resulting in loss of statistical power and
  - too low estimates of standard error
  - too-narrow confidence limits
  - prevents valid analysis of cross-level interactions
- Multilevel regression can handle variables at different levels
  - Can also use aggregated data as group level independent variables
  - But dependent variable must be at the lowest level

Why is this important?

Correctly specify models



How does it work?

- Multilevel models partition and analyse the variance of the DV by the hierarchy of the data
- Modelling variance at multiple levels allows variables to explain variance at their theoretical level
- In effect parameters (coefficients) of the model are given a probability model with its own parameters
- Consider the example of 1000 doctors from 10 hospitals
    - We want to model burnout as a function of working hours and hospital resources
    - If we perform 10 linear regressions we can estimate the regression coefficients
    - We can then use another linear regression with sample 10 to explain those coefficients using hospital resources as a predictor
- Multilevel regression combines those two steps

R for multilevel regression

For multilevel analyses we will discuss three libraries:

- “multilevel” : designed to be used by organizational and social psychologists
  - Includes estimates of within-group agreement, RGR, basic routines for estimating reliability
- “nlme”: used to fit and compare Gaussian linear and nonlinear mixed-effects models
- “lme4”: used to fit linear and generalized linear mixed-effects models
- nlme and lme4 use a slightly different syntax - but it is easy to translate one into the other
- For this workshop we will be using RStudio (but you can use any other IDE or text editor)
- Other packages used are ggplot2
- For this presentation I used knitr and slidify

Session 2: Estimating Multilevel Models in R

Estimating Multilevel Models

Linear Regression Equation:

$$Y_i = \beta_0 + \beta_1 X_i + r_i$$

- Assumptions for residuals:
- Normally distributed with mean=0
- Homoscadicity – residuals are constant across different levels of the IV
- Independent from each other (uncorrelated)

Estimating Multilevel Models – Systems of equations

- Indexing for group membership allows to specify varying coefficient (i.e. a different coefficient for each group):

$$Y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + r_{ij}$$

If we do not want the intercepts and slopes to vary according to group membership then

$$Y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + r_{ij}$$

$$\beta_{0j} = \gamma_{00}$$

$$\beta_{1j} = \gamma_{10}$$

...which is the equivalent of the linear regression

Steps for multilevel modelling

Rather than testing the final hypothesised model, we test the model incrementaly

- Step 1: Examine if a varying intercept imrpoves the model
- Step 2: Add level-1 and level-2 predictors
- Step 3: Examine if varying slope for level-1 predictors improves the model (optional)
- Step 4: Add level-2 predictors to explain slopes variation: i.e. cross-level interactions (optional)

Step 1 - Varying intercept

- For the first step we need to examine if a random intercept model has a better fit than a fixed intercept model
- Thus we need to fit both:
  - Fit a generalised least squares model without predictors (using REML)
  - Fit a varying intercept model without predictors (using REML)
  - Compare the two models
- Multilevel models are estimated using restricted maximum likelihood (REML) so we need to fit a linear model (with fixed effects) estimated with REML (in order to compare them)
- We can do that with the gls function (in the nlme package)
- The syntax for gls is the same as for lm

The Null Model (unconditional)

- In order to create a random coefficient model either the intercept or the slope must be allowed to vary (and we don't even need level-1 or level-2 predictors)

$$Y_{ij} = \beta_{0j} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + u_{0j}$$

- Where u0j is the error term for γ00
- This is essentially one-way random effects ANOVA
- We use this model as a baseline for comparison

Step 1: Fitting a lme model

- lme command is the key function in the nlme package
- In its most basic format for fitting a random intercept model with no predictors the lme syntax will be:

```
lme(y~1, random=~1|Grouping_Variable, data=dataset)
```

- To get the results from either model we can use the summary function
- To compare the two models we can extract the model log likelihood using the logLik function
- Multiplying by -2 their difference we get the -2LL (-2 log likelihood) which has a chi-square distribution with degrees of freedom (DF) equal to the difference of the DF of the two models
- We can also compare the two models with the anova function (which will give as the -2LL and whether it is significant)
- We can also compare two models using the AIC

Step 1: Lets try it

- Load the “eduation” dataset and explore the data

```
load(“UCLMultilevel.Rdata”)
ls()
names(education)
head(education)
```

- Fit a linear regression with

```
summary(lm(perf~gender+tage+tgender+hours+partic+pcohesion, data=education)
```

Step 1: Lets try it

- Fit the gls and lme models without predictors and get the summary of the results

require(nlme)

```
gls.null <- gls(perf ~1, data=education)
lme.null <- lme(perf ~1, random=~1|grpid, data=education)
summary(gls.null)
summary(lme.null)
```

Step 1: Lets try it

- Get the -2LL and the difference of the models

```
logLik(gls.null); logLik(lme.null)
1-pchisq(-2*I(logLik(gls.null)[1]-logLik(lme.null)[1]), 1)
```

- or simply compare them with the anova function

```
anova(gls.null, lme.null)
```

Step 2: Adding predictors

- Up to now we showed that allowing a random intercept improves the model
- This is signified by the level-2 error
- Adding level-1 predictors to the model is no different than adding predictors to a multiple linear regression

$$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + u_j$$

$$\beta_{1j} = \gamma_{10}$$

Step 2: Adding level 2 predictors

- Once the intercept is allowed to vary (through the addition of the second level error) we can add level-2 predictors
- Adding a level-2 predictor is to add a predictor to explain the variation of the intercept
- Thus, we are hypothesising that the intercept varies because of some other level 2 variable rather than just because of error

$$Y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j$$

$$\beta_{1j} = \gamma_{10}$$

- If we substitute the beta coefficients in the level-1 equation with the level-2 equations we get

$$Y_{ij} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j + \gamma 10 X_{ij} + r_{ij}$$

Step 2: level-1 and level-2 predictors

- Now the equation looks like a multiple linear regression with level-1 and level-2 predictors and two error terms
- In lme we can add level-1 or level-2 predictors by adding them to the “fixed” part of the equation
- Testing the effect of either level-1 or level-2 is done through a t-test
- Based on the t-test we can calculate the p-value given the degrees of freedom
- However although lme reports the degrees of freedom there are some issues associated with calculating them (we will revisit this issue later on)

Step 2: level-1 and level-2 predictors

- Another way to evaluate significance of predictors is to compare models with and without the predictors (using the anova function)
- But we cannot compare these models if they are fitter with Restricted Maximum Likelihood and we will need to refit the model with Maximum Likelihood
- We can do this by specifying in our model that method="ML"

```
lme(y ~ x1 + x2 + x3, random=~1|Grouping_Variable, data=dataset, method="ML")
```

- We can use the drop1 command to test all of the predictors iteratively

```
drop1(fullmodel, test="Chisq")
```

Step 2: level-1 and level-2 predictors

- In the education data frame we only have two group level predictors: teacher age (tage) and teacher gender (tgender)
- However we can aggregate, hours, partic, and pcohesion to the classroom level to create level-2 predictors
- Adding level-2 predictors is NOT necessary for testing a multilevel model
- Once we fit a model we can extract the variance components

```
VarCorr (lme.object)
```

- By examining the variance components of the models with and without the predictors we can estimate a pseudo \$R2\$ (using the same method as for linear regression)

\$\$ R2=1-\frac{Variance-With-Predictors}{Variance-Without-Predictors} \$\$

Step 2: Let's try it

- Fit a model with gender as a level-1 predictor

```
lme.m1 <- lme(perf~gender, random=~1|grpid, data=education)
summary(lme.m1)
```

- Get the variance components of the models and calculate variance explained

```
VarCorr(lme.null)
VarCorr(lme.m1)
```

- Add the rest of the predictors

```
lme.m2<-lme(perf~gender+hours+partic+pcohesion, random=~1|grpid, data=education)
summary(lme.m2)
```

Step 2: Let's try it

- Refit the model with method="ML" and iteratively compare models with and without each predictor

```
lme.m2ml <- lme(perf~gender+hours+partic+pcohesion, random=~1|grpid, method="ML",data=educatic
drop1(lme.m2ml, test="Chisq")
```

- Create group level variables for hours, partic and pcohesion and add them to the model

```
temp <- aggregate(cbind(education$hours, education$partic, education$pcohesion),  by=list(educ
names(temp)<-c("grpid", "hours.grp", "partic.grp", "pcohesion.grp")
education<-merge(education, temp1, by="grpid", all=T)
lme.m3<-lme(perf~gender+tage+tgender+hours+partic+ pcohesion+hours.grp +
            partic.grp + pcohesion.grp, random=~1|grpid, data=education)
summary(lme.m3)
```

Step 3: Random slopes

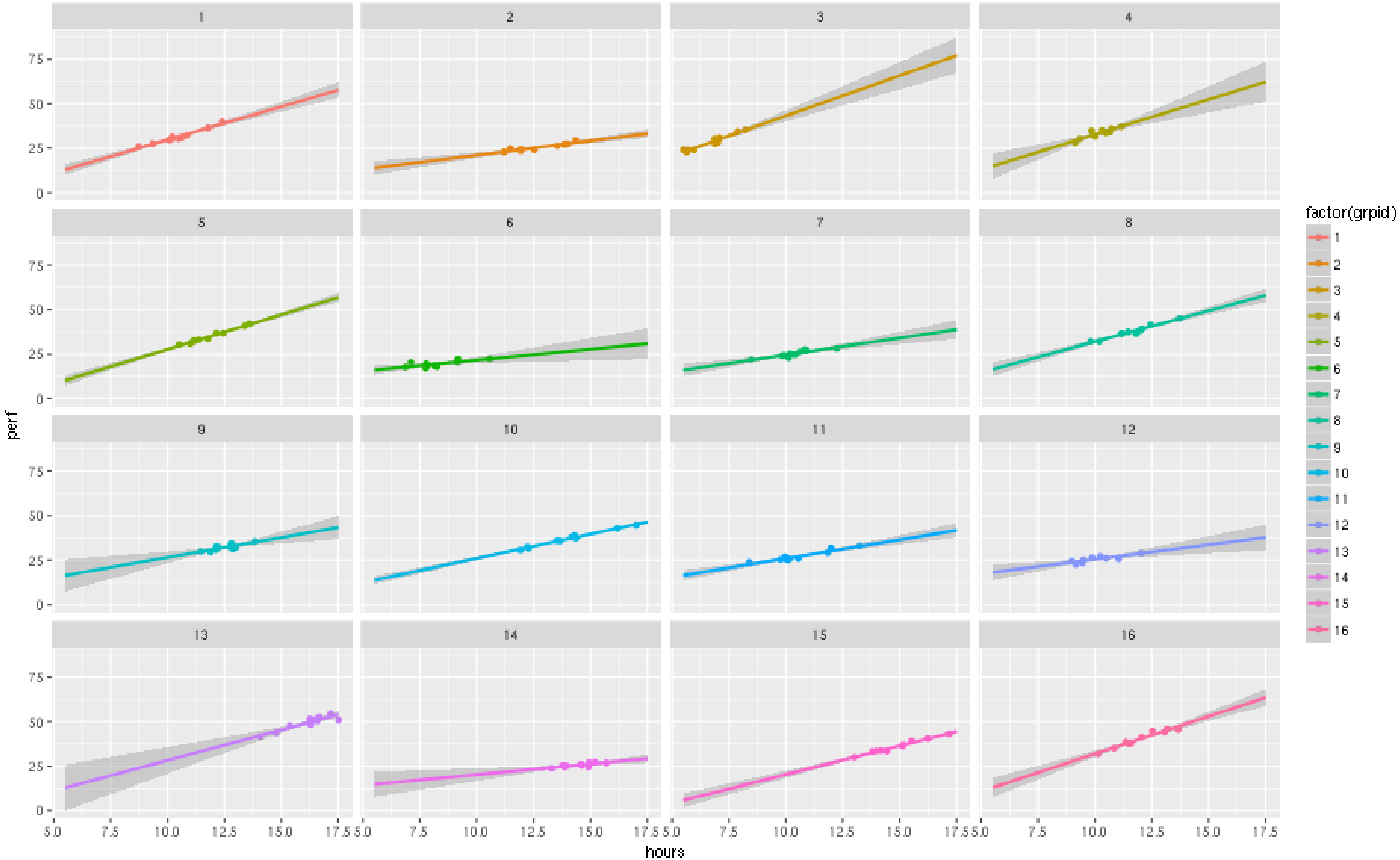
- This is an optional step
- It allows to test the hypothesis that the effect of a level-1 predictor is different in each group
- In the same way we allowed for variation of the intercept we can allow a slope of one (or more) of the level-1 predictors to vary

$$Y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j$$

$$\beta_{1j} = \gamma_{10} + w_j$$

Step 3: Random slopes



Step 3: Random slopes

- In lme we can do this by adding the variable whose slope we want to allow to vary to the random part of the model

```
lme( y ~ x1 + x2 + x3, random=~1+x1|grpid, data=mydata)
```

- It is also possible to remove the varying intercept and only allow for a varying slope

```
lme( y ~ x1 + x2 + x3, random=~-1+x1|grpid, data=mydata)
```

- To evaluate if allowing a varying slope for hours improves our model we can use the anova command to compare the model with a random slope to the model with the fixed slope (we do this REML)

Step 3: Let's try it

- Plot regression relationships between one level-1 variable and the dependent variable for each group to examine visually if there is slope variability

```
library(ggplot2)
ggplot(education[1:160,], aes(x=hours, y=perf, color=factor(grpid))) +
       geom_point() + stat_smooth(method="lm", fullrange=T)
       + facet_wrap(~grpid)
```

- Then we can fit the model and compare with the fixed slope model

```
lme.m4<-lme(perf~gender+tage+tgender+hours+partic+ pcohesion+
        hours.grp + partic.grp + pcohesion.grp,
        random=~hours|grpid, data=education)
anova(lme.m3, lme.m4)
summary(lme.m4)
```

Step 3: Let's try it

- We can also extract and plot the varying intercept and slope

```
ranef(lme.m4)
plot(ranef(lme.m4))
```

- We can reorder the varying intercept to get a nicer looking plot

```
plot(ranef(lme.m4)[order(ranef(lme.m4)[[1]]),])
```

Step 4: Cross-level interactions

- This is also an optional step and it is only meaningful if we have a varying slope in our model
- It allows to test hypotheses about why the effect of a level-1 variable varies according to a level-2 predictor
- Essentially we are using level-2 predictors to explain variability of the slopes

$$Y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11} Z_{1j} + w_j$$

Step 4: Cross-level interactions

- From the previous equation if we substitute the level-2 equations into the level 1 equation we get

$$Y_{ij} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j + (\gamma_{10} + \gamma_{11} Z_{1j} + w_j) X_{ij} + r_{ij}$$

- Which simplifies to

$$Y_{ij} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j + \gamma_{10} X_{ij} + \gamma_{11} Z_{1j} X_{ij} + w_j X_{ij} + r_{ij}$$

- Note that the level-2 predictor is now multiplied with the level 1 predictor whos slope was allowed to vary
- This shows that when we use level-2 predictors to model slope variability at level 1 we are essentially testing an interaction between variables at two different levels

Step 4: Let's try it

- Simply add an interaction effect between the variable with the varying slope and the level-2 variable to the fixed part of the model
- Note: The level-1 variable should also be specified in the random part of the model

```
lme.m5<-lme(perf~ gender+tage+tgender+hours+ partic + pcohesion+
          hours.grp + partic.grp + pcohesion.grp +
          hours*hours.grp+ hours*partic.grp + hours*pcohesion.grp,
          random=~hours|grpid, data=education)
summary(lme.m5)
```

- We can use Maximum Likelihood to test if the interactions improved the model fit

```
lme.m5ml<-lme(perf~ gender+tage+tgender+hours+ partic + pcohesion+
          hours.grp + partic.grp + pcohesion.grp +
          hours*hours.grp+ hours*partic.grp + hours*pcohesion.grp,
          random=~hours|grpid, method="ML" data=education)
drop1(lme.m5ml, test="Chisq")
```

Session 3: Longitudinal Data

Longitudinal data

- Longitudinal data are naturally multilevel
- We have observations over time nested within a higher level unit such as individuals or organizations
- The advantage of multilevel models for longitudinal data is that cases with missing responses from one time period don’t have to be excluded
- Similarly it allows for modelling time data collected from different time points
- Data need to be arranged at the lower level, with a row of data for each time-point within the higher level unit

Longitudinal data

- The steps we follow towards testing multilevel models with longitudinal data is very similar to the steps followed for cross-sectional data
- First we specify the individual (or the higher level unit) as our grouping variable in the model and test a random intercept model
- BUT before entering any predictors we test the “growth model”
- This involves the time variable and (a) we test for it’s fixed effect and (b) for the variation of this effect between the subject (as defined by the higher level variable)
    - i.e. we test random intercept model with time as the only predictor and subsequently we test the same model but with time as a random effect
- The slope of time represents the trajectory of change of the dependent variable
- By testing it as a random effect we test if this trajectory varies between individuals
- We can also model the time variable as a non-linear trajectory
    - e.g. using splines or polynomial models

Longitudinal data

- Although specifying the effect of the time variable as a varying slope we can model time, sometimes longitudinal data present serial correlation
- For instance we expect that health at time 1 will be more closely related to health at time 2 rather than health at time 3 (or with any other measurement in the future)
- To deal with this issue we can apply the appropriate correlation matrix (in this case an autoregressive correlation matrix) for the situation to model within-subject correlations of repeated measures
- With lme we can add the parameter correlation=corAR1()

```
lme(y~time, random=~time|subject, correlation=corAR1(), data=mydataset)
```

- We can then test whether doing so adds value to our model by comparing it a model without the autoregressive correlation matrix

Longitudinal data

- One of the more interesting hypotheses with longitudinal data is that we can test the cross-level interaction between time and a predictor measured at the higher level
- Since time is modelled as a varying effect, a cross-level interaction would mean that the higher level predictor explains the varying coefficients (slopes) for time

Steps for longitudinal data analysis

- Step 1: Examine if a varying intercept imrpoves the model
- Step 2a: Add time as a fixed effect
- Step 2b: Specify time as a random effect and compare with model from step 2a
- Step 2c: Test for serial correlation and compare with model from step 2b
- Step 3: Add level-1 and level-2 predictors
- Step 4: Examine if varying slope for level-1 predictors improves the model (optional)
- Step 5: Add level-2 predictors to explain slopes variation: i.e. cross-level interactions (optional)
    - it is particularly interesting to test cross-level interactions with time
    - since the slope for time represents the rate of change of the trajectory of the dependent variable, a significant interaction would suggest that the higher level predictor explains the steepness of the trajectory or the rate of change of the dependent variable

Longitudinal data: Let’s try it

- Load the “migraine.Rdata” data and explore it

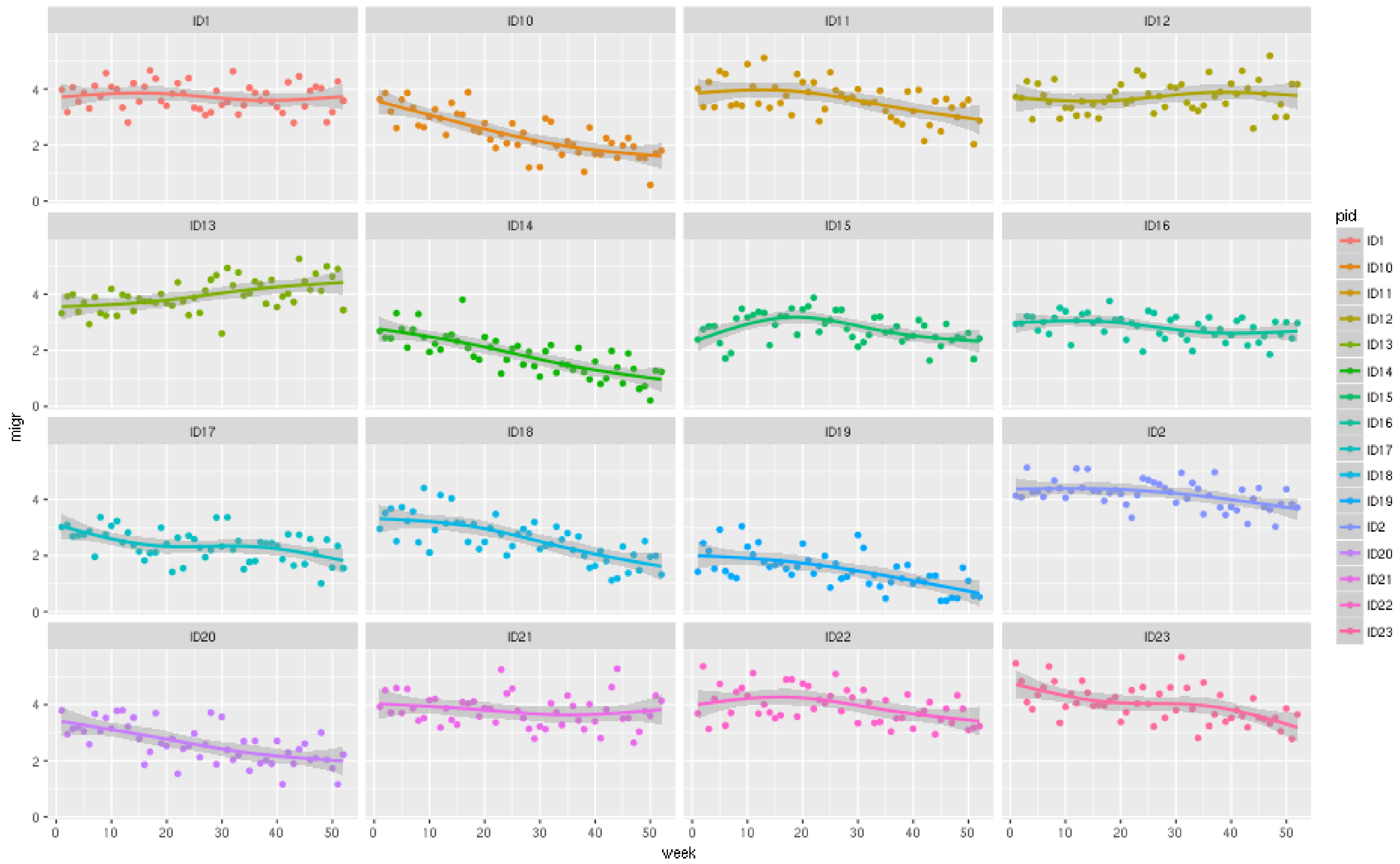
```
require(nlme) ; require(multilevel)
load(“UCLMultilevel.Rdata”)
ls()
names(migraine)
head(migraine)
```

- We can plot some the relationship between week (our time variable) and migr (our dependent variable) to see how it looks like

```
require(ggplot2)
require(splines)
migraine<-migraine[order(as.numeric(migraine$pid)),]
ggplot(migraine[1:832,], aes(x=week, y=migr, colour=pid)) + geom_point() +
        stat_smooth(method="lm", formula=y~ns(x,3)) + facet_wrap(~pid)

# If you do not have the splines package you can change the formula to y~poly(x,3, raw=TRUE)
```

Migrain data



Longitudinal data: Let's try it

- Step 1: Fit the gls and null model and compare them

```
gls.null<-gls(migr~1, data=migraine)
lme.null<-lme(migr~1, random=~1|pid, data=migraine)
anova(gls.null, lme.null)
```

- Step 2a: Add the time variable as a predictor
- Note it is preferable for the time variable to start from 0
- So transform the week variable so that week 1 is 0 in the data set

```
migraine$week<-migraine$week-1
lme.m1<-lme(migr~week, random=~1|pid, data=migraine)
summary(lme.m1)
```

Longitudinal data: Let's try it

- We can also try for a quadratic model (or any other shape for the trajectory of the dependent variable)

```
lme.m1b<-lme(migr~week+I(week^2), random=~1|pid, data=migraine)
summary(lme.m1b)

lme.m1c<-lme(migr~ns(week,3), random=~1|pid, data=migraine)
summary(lme.m1c)
```

Longitudinal data: Let's try it

- Step 2b: Fit the random effect for week and compare the model to a fixed effect model

```
lme.m2<-lme(migr~week, random=~week|pid, data=migraine)
anova(lme.m1, lme.m2)
summary(lme.m2)
```

- Step 2c: Specify an autoregressive correlation matrix and test if it adds to the model fit

```
lme.m3<-lme(migr~week, random=~week|pid, correlation=corAR1(), data=migraine)
anova(lme.m2, lme.m3)
```

Longitudinal data: Let’s try it

- Step 3: Add predictors to the model

```
lme.m4<-lme(migr~age+week+treat, random=~week|pid, data=migraine)
summary(lme.m4)
```

- For this dataset we can skip step 4 because we don't care for additional random effects
- Step 5: The final step would be to see if we there is a cross level interaction between week of treatmens and number of treatments
    - i.e to explain the steepness of the trajectory of migrains (or the rate of change in migrains)

```
lme.m5<-lme(migr~age+week*treat, random=~week|pid, data=migraine)
summary(lme.m5)
```

Session 4: Aggregation Testing: Agreement and Reliability

Aggregation testing

- Often we want to to aggregate lower level variables to create group level variables
  - e.g. measuring culture using a questionnaire
- Before we aggregate such variables we need to examine if it is meaningful to do so
- There are two types of tests:
  - tests that evaluate agreement
  - tests that evaluate reliability
- What we need depends of the nature of the variables to be aggregated (aggregation model)

Typology of aggregation models (Chan, 1998)

- Additive (no concern about agreement)
- Direct Consensus
  - Referent of item is individual-level
  - Need to establish within-group agreement
  - Assume Functional Isomorphism (e.g. job satisfaction)
- Referent-Shift Consensus Models
  - Referent of item shifted to group
  - Within-group agreement still important
  - Isomorphism not assumed (e.g. cohesion)
- Dispersion Models
  - Model within-group variance

Tools for aggregation testing

- Two tools used to support aggregation models
- Reliability-based measures (useful to justify all models)
  - ICC(1) and ICC(2) (Intra-class Correlation Coefficient)
  - Eta-squared ($\eta2$)
- Agreement-based measures (useful in isomorphic models such as Chan's consensus models)
  - r_{WG}
  - $r_{WG(j)}$
  - $r_{*WG(j)}$
  - adm

Reliability approaches

- Measure of the proportional consistency of variance among raters
- ICC(1) - How much of the variability in individual responses can be predicted by group membership?
- ICC(2) - tests the reliability of the group means
- Eta-Squared ($\eta2$) - Represents the ratio of between group Sum of Squares to total Sum of Squares
- As group sizes increase, eta-squared values asymptotically approach ICC(1) values. When group sizes are small, eta-squared gives a very inflated estimate of ICC(1)
- Can be calculated from an one-way ANOVA

Agreement approaches

- Agreement reflects the degree to which individuals provide essentially the same rating
- Most commonly used index are
  - r_{WG} (for single items)
  - $r_{WG(j)}$ (for scales)
- Compares observed group variances to an expected random variance
- To calculate the expected random variance from a uniform distribution

$$ erv=\frac{A2-1}{12} $$

- where A is the number of response options
- More that .70 as a heuristic
- $r_{WG(j)}$ corrects for inflation of reliability with multiple items (Lindell et al, 1999)
- ADM is an alternative measure based on the average deviation from the mean

Functions for aggregation testing

- Functions for aggregation testing can be found in the multilevel library
- For reliability
  - ICC1
  - ICC2
- For Agreement:
  - rwg, rwg.j, rwg.j.lindell, ad.m
  - rwg.sim, rwg.j.sim, ad.m.sim
- To get help on these functions use help() or ?. e.g.:

```
help(rwg)
?rwg
```

Reliability testing in R

- We can try it with the cohesion variable in the klein2000 dataset in the “multilevel” library
- To do this load the library and the dataset
- Fit an ANOVA model with GRPID as the the independent variable and COHES as the dependent
- Assign the model to an object (e.g. coh.aov)
- Get a summary of the ANOVA results to estimate eta-square
- Use the ICC1 and ICC2 functions with the ANOVA object you created

```
library(multilevel)
data (klein2000)
coh.aov <- aov(COHES~GRPID, data=klein2000)
summary(coh.aov)
ICC1(coh.aov)
ICC2(coh.aov)
```

Agreement testing

- Evaluate agreement using the function rwg, rwg.j, rwj.lindell, and ad.m
- The R syntax for these functions is

```
rwg(x, grpid, ranvar=2)
rwg.j(x, grpid, ranvar=2)
rwg.j.lindell(x, grpid, ranvar=2)
ad.m(x, grpid, type="mean")
```

- For rwg, x is a vector and for rwg.j and rwg.j.lindell is a matrix of the variables to be analysed
- For ad.m, x can be either a vector or a matrix
- grpid is the group ID variable and ranvar is the expected random variance

Agreement testing

- Load the lq2002 dataset from the “multilevel” library
- Calculate the r_{WG} and adm scores for the LEAD01 variable for each group in the COMPID variable
- Calculate the $r_{WG(j)}$ for all the LEAD variables for each group in the COMPID variable
- Do the same for the $r_{*WG(j)}$ and adm scores

```
data(lq2002)
rwg(lq2002$LEAD01, lq2002$COMPID, ranvar=2)
rwg.j(lq2002[3:13], lq2002$COMPID, ranvar=2)
rwg.j.lindell(lq2002[3:13], lq2002$COMPID, ranvar=2)
ad.m(lq2002[3:13], lq2002$COMPID, type="mean")
```

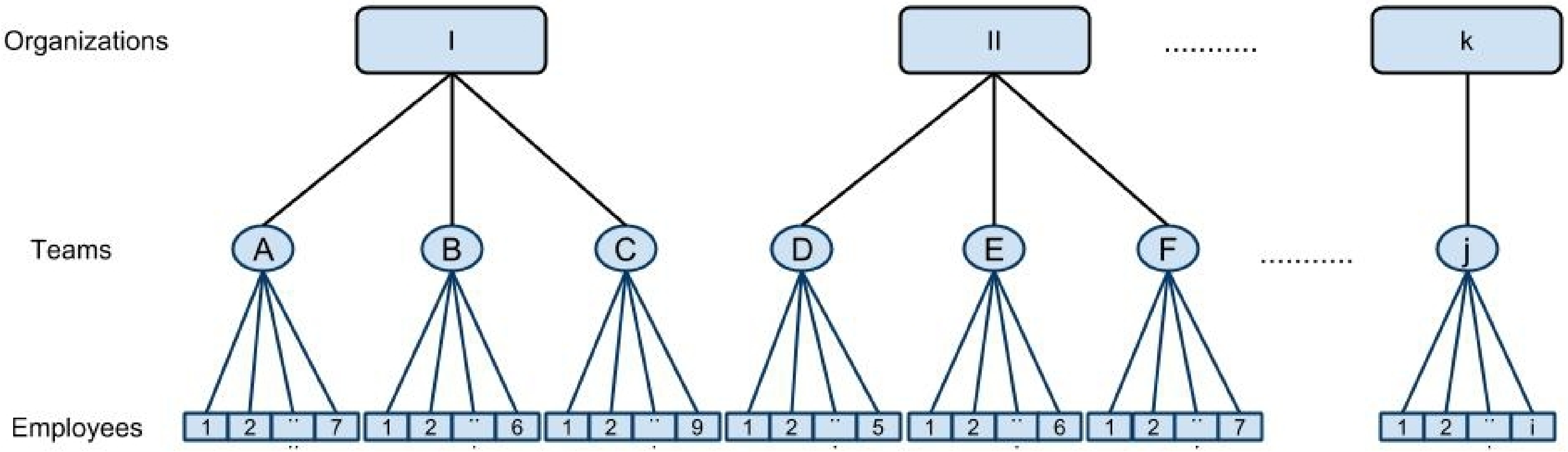
Session 5: Advanced models with lme4

Advanced Models

- We can extend multilevel models in a lot of different ways. Some examples:
    - More than 2 levels
    - Multilevel Generalised Linear Models
    - Multilevel Structural Equation Models
- lme4 is an R package for generalised mixed effects models
- It was written as a replacement of nlme
- It is very efficient when it comes to fitting models with more complex structures
- It is faster and has a better syntax than nlme
- But some models are still easier to fit with nlme
- It has a better syntax for specifing models with three or more levels (can also be done with nlme)
- Can easily specify cross-classified models

Advanced Models: three level models

- Responses at one level are nested within another which is nested within a third level
- e.g. Employees nested within departments nested within organizations



Advanced Models: three levels

- We can also write the equations for this structure
- For example for a random intercept model with three levels

$$Y_{ijk} = \beta_{0jk} + \beta_{1jk}X_{ijk} + r_{ijk}$$

$$\beta_{0jk} = \gamma_{00k} + \gamma_{01k}Z_{1jk} + u_jk$$

$$\gamma_{00k} = \delta_{00} + \delta_{01}Z_{2k} + W_k$$

For

1…i individuals

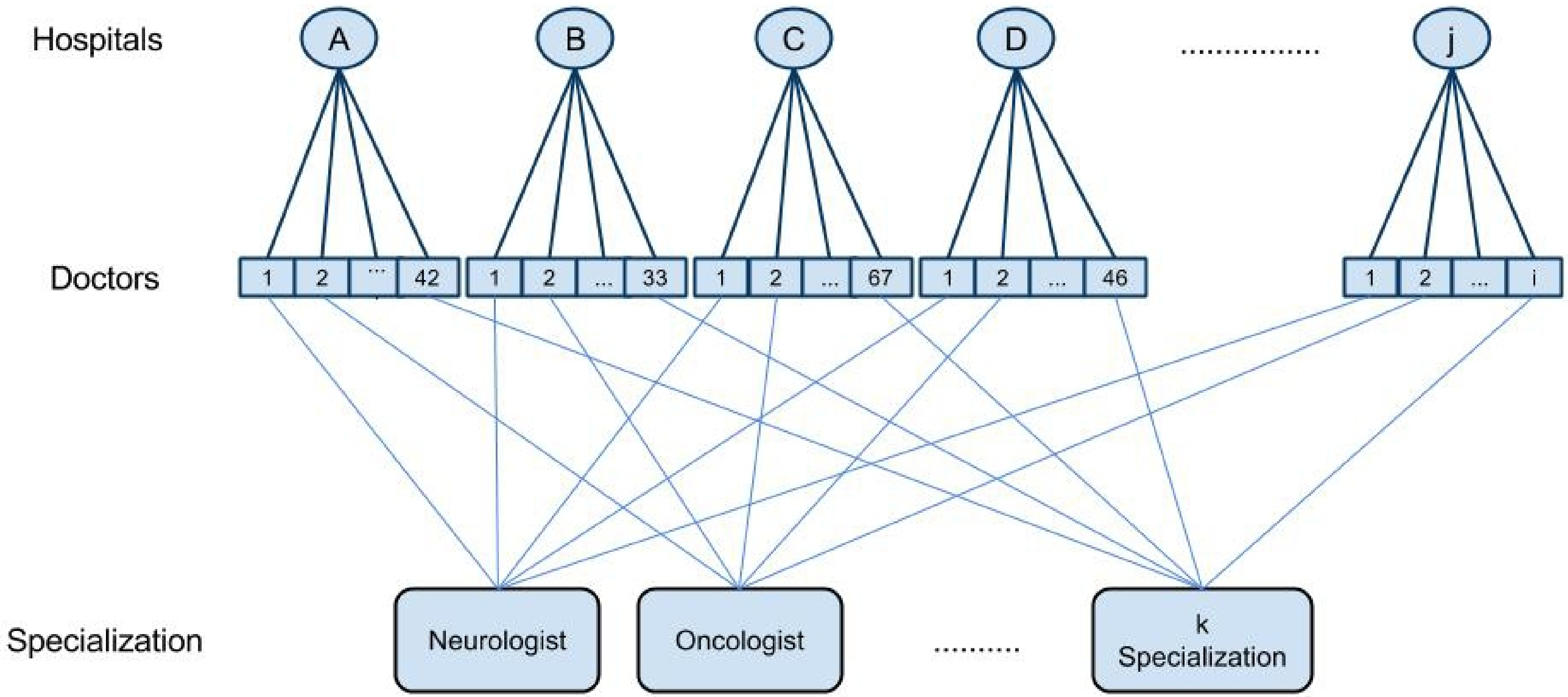
1…j workplaces

1…k organizations

- We can further extend this model by specifing random slopes for variables at either the first or second level

Advanced Models: cross classified

- Responses at one level are nested within more than one overlapping higher level
- E.g. doctors within hospitals and doctors within specializations
- E.g. Children within classrooms and children within neighbourhoods



Advanced Models: cross classified

- We can write the equations for this model by adding another random intercept to the first level

$$Y_{ijk} = \beta_{0j} + \beta_{1k} + \beta_{2jk} X_{ijk} + r_{ijk}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + u_j$$

$$\beta_{1k} = \gamma_{10} + \gamma_{11} Z_{2k} + W_k$$

For

1…i individuals

1…j hospitals

1…k specializations

- Similarly we can extend this by specifing random slopes and allowing each of the slopes to vary either per hospital or per specialization (or the interaction of the two)

Advanced Models: lme4 syntax

- lmer is the main function of lme4
- lme4 uses a slightly different syntax than nlme
- Random effects are specified as part of the equation by putting them in parentheses
- For example the following two models are equivalent

```
model1<- lme(y~x, random=~1|group, data=mydata)
model1<-lmer(y~x +(1|group), data=mydata)
```

- This allows to specify some correlation structures very easily e.g. when we want no correlation between the varying intercept and a varying slope

```
model1<-lmer(y~x +(1|group)+(0+x|group), data=mydata)
```

Advanced Models: lme4 syntax

- It also allows to specify multiple different groupings for more complex structures in the same way
- A three level model

```
lmer(y~x +(1|level3/level2), data=mydata)
```

- A crossed classified model

```
lmer(y~x +(1|level2) +(1|level3), data=mydata)
```

- Note that for a three level nested structure both specifications will produce the same results
- However for a crossed classified dataset the (1|level3/level2) specification will fit a nested model where the lower level is the cross-tabulation of the two crossed levels
- Note that lme4 does not allow to specify correlation structures directly (e.g. for autoregressive structures)

Advanced Models: Let’s try it

- Load the “orgs” dataset (in UCLMultilevel)
- Simulated data on the job satisfaction of employees
- Each employee (n=4792) is working in a workplace (n=250) or department and each workplace is nested within an organization (n=10). In addition employees are also nested in different occupations (n=6)

```
load(“UCLMultilevel.Rdata”)
names(orgs)
head(orgs)
unique(orgs$org); unique(orgs$wp); unique(orgs$jobs)
```

Advanced Models: Let's try it

- Load lme4 and fit the null models (with wp as grouping variable) with the lmer and gls commands
- Get the Log Likelihood for the two models and test if it is significant (note that we cannot use anova to compare gls and lmer models)

```
library(lme4)
gls.m0<-gls(js~1, data=orgs)
lmer.m0<-lmer(js~1+(1|wp), data=orgs)
anova(gls.m0, lmer.m0)
summary(lmer.m0)
logLik(gls.m0)
logLik(lmer.m0)
1-pchisq(-2*(logLik(gls.m0)[1]-logLik(lmer.m0)[1]),1)
AIC(gls.m0)
AIC(lmer.m0)
```

Advanced Models: Let's try it

- Add the level-1 predictors to the model

```
lmer.m1<-lmer(js~age+gender+demands+control+(1|wp), data=orgs)
summary(lmer.m1)
```

- Add group level predictors

```
lmer.m2<-lmer(js~age+gender+demands+control+ support.wp+(1|wp), data=orgs)
summary(lmer.m2)
```

Advanced Models: Let's try it

- lmer does not report p-values or degrees of freedom
- but there are other ways that we can obtain that information - we will examine 3 different approaches
- 1. We can evaluate significance by comparing nested models (using Maximum Likelihood)
- We can get ML estimates by specifing "REML=FALSE" (true is the default) which is slightly different syntax from lme

```
lmer.m1ml<-lmer(js~age+gender+demands+control+(1|wp), REML=FALSE, data=orgs)
drop1(lmer.m1ml, test="Chisq")
```

Advanced Models: Let's try it

- 2. The second approach is to obtain an estimate of the degrees of freedom (and thus calculate p-values)
- This can be done with the help of the "lmerTest" library

```
library(lmerTest)
lmer.m2<-lmer(js~age+gender+demands+control+ support.wp+(1|wp), data=orgs)
summary(lmer.m2)
```

Advanced Models: Let’s try it

- 3. The final approach is through confidence obtaining confidence intervals for the estimates
- lmer provides the function "confint" to do that which allows to compute confidence intervals using 3 methods
- Profile (default): usus the likelihood profile to find the appropriate cut-off points (similar to the nested models but for confidence intervals)
- Wald: based on the standard normal distribution. Very fast but less accurate unless there is a large sample
- Bootstrap: Most accurate and most computationally intensive

```
confint(lmer.m2, method="profile")
confint(lmer.m2, method="Wald")
confint(lmer.m2, method="boot", 1000, .progress="txt", PBargs=list(style=3))
```

Advanced Models: Let’s try it

- Add the third level and test if the third level contributes to the model
- Since we are comparing models with different random effects we need to use the option "refit=FALSE"
- This is because the anova comnmand for lme4 will refit the models with ML rather than REML

```
lmer.m3<-lmer(js~age+gender+demands+control+ support.wp+ (1|org/wp), data=orgs)
summary(lmer.m3)
anova(lmer.m2, lmer.m3, refit=FALSE)
```

- Add group level predictor for the third level

```
lmer.m4<-lmer(js~age+gender+demands+control+ support.wp+resources.org+(1|org/wp),  data=orgs)
summary(lmer.m4)
```

Advanced Models: Let’s try it

- Add the cross-classified “jobs” and test if it contributes to the model fit

```
lmer.m5<-lmer(js~age+gender+demands+control+support.wp+resources.org+
        (1|org/wp) + (1|jobs), data=orgs)
summary(lmer.m5)
anova(lmer.m4, lmer.m5, refit=FALSE)
```

- Add group level predictors for jobs and re-test using Maximum Likelihood to evaluate contribution of predictors

```
lmer.m6<-lmer(js~age+gender+demands+control+ support.wp+resources.org+ tasksig.jobs+
          (1|org/wp) + (1|jobs), data=orgs)
lmer.m6ml<-lmer(js~age+gender+demands+control+ support.wp+resources.org+ tasksig.jobs+
          (1|org/wp) + (1|jobs), REML=FALSE, data=orgs)
drop1(lmer.m6ml, test="Chisq")
```

Advanced Models: GLMM

- Further to gaussian models lme4 can fit models for a variety of distributions and link-functions e.g.
  - logistic models (binomial distribution with logit link function)
  - poisson models
  - gamma
- We will not go into that but you can try it on your own using the “family” parameter of lmer
- The orgs dataset has a measure of days absence which is a count variable and you can model as a multilevel poisson model (family="poisson")

```
lmer(absence~age+gender+demands+control+(1|wp), family="poisson", data=orgs)
```

Advanced Models: Centering

Grand mean centering

- Subtracting the grand mean of the sample from each of the scores
- It is a linear transformation so results do not change from untransformed variables

Group mean centering

- Subtracting the Jth group mean from each observation within that group
- It is a non-linear transformation of the variable data so refitting a model will result in a change of all the model coefficients
- Each score of the centered variable is the deviation from the group mean so the meaning of the predictor changes

Advanced Models: Centering

Grand mean centering

- The intercept estimates reflect the expected value of the dependent variable for each group when all predictors are at their mean score.
- E.g. for the education data the intercepts will reflect the expected performance outcome for a student in class j who studied the mean time of hours, have an average class participation and his/her reported perceived cohesion is at the mean of all students (i.e the average student)

Group mean centering

- The intercept estimates reflect the expected value of the DV for group J when all predictors in are at the mean score of group J.
- E.g. For the education example the intercepts will reflect the expected performance outcome for a student in class j who studied the mean time of hours of class J, have an average class participation of class J and his/her reported perceived cohesion is at the mean of students in class J (i.e the average class J student)

Advanced Models: Centering

- Decisions about group or grand mean centering should be theory driven
- But there are also some methodological considerations
- For grand mean centering
  - it is easier to fit and interpret the varying intercept
  - It is easier to interpret models with level-1 interaction effects or quadratic effects
- For group mean centering
  - It removes collinearity when there are group level predictors of the same level-1 variables
  - Testing mediation effects
- For models with varying slopes (and/or cross-level interactions) it is easier to interpret (because with grand mean centering the estimate will also involve some between group variability)

Thank you. Any questions?