ADVANCED DATA ANALYSIS FOR PSYCHOLOGICAL SCIENCE

Part 1. Introduction to multilevel modeling

Luca Menghini Ph.D.

luca.menghini@unipd.it

***

Master degree in Developmental and Educational Psychology

University of Padova

2023-2024

Outline of Part 1

- **LM recap**: Short recap of linear regression modeling 🔬 Ⓡ
- **LMER**: Introduction to multilevel modeling (*linear mixed-effects regression*)
- **Data processing**: How to approach a multilevel data structure? How to manipulate and pre-process multilevel data? Ⓡ
- **Descriptives**: Which descriptive stats should be reported from a multilevel dataset? How to compute and interpret them?
- **Model fit**: How to fit a multilevel model in R? How to inspect, report, visualize, and interpret the results of a multilevel model? Ⓡ
- **Model evaluation**: Which are the assumptions of multilevel models? How to evaluate them? How to compare multiple models and select the best model? Ⓡ
- **Related**: Summaries & in-depth topics related to multilevel modeling (e.g., generalized and Bayesian LMER, power analysis) 🔬

🔬 = not for the exam

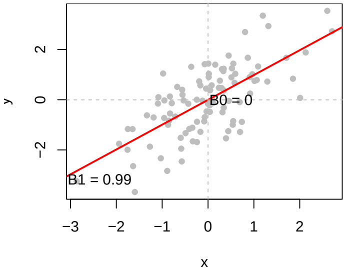
Ⓡ = exercises with R (bring your laptop!)

Linear regression models

Linear models (LM) allow to determinate the link between two variables as expressed by a linear function: $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Such a function can be graphically represented as a **straight line**, where:

- β_0 is the **intercept** (value assumed by y when x = 0)
- β_1 is the **slope** (predicted change in y when x increases by 1 unit)
- ϵ_i are the **errors** (distance between observation i and the regression line)



x_i and y_i are the values of observation i for the **casual variables** x and y

β_0, β_1, and ϵ_i are called "**parameters**", or "**coefficients**". They are *estimated* from the sampled data and *generalized* to the whole population.

Fitting linear models in R

```r
data("children", package = "npregfast") # loading children dataset from npregfast pkg
```

R uses the `lm()` function to fit linear models with the arguments `formula` (`y ~ x1 + x2 + ...`) and `data` (identifying the dataframe with the model variables).

Null model

Children' `height` is only predicted by the model **intercept** β_0 = expected (i.e., mean) value of `height` in the sample. σ^2 is the **variance of the residuals** ϵ_i (deviations from the intercept).

```r
m0 <- lm(formula = height ~ 1,
         data = children)
coefficients(m0) # model parameters
```

```
(Intercept)
   153.4013
```

```r
summary(m0)$sigma^2 # residual variance
```

```
[1] 243.9085
```

Simple regression model

`height` is now predicted by the **intercept** β_0 (mean value when `age` is 0), the **slope** β_1 (expected change for 1-unit increase in `age`), and the **residual variance** σ^2.

```r
m1 <- lm(formula = height ~ age,
         data = children)
coefficients(m1) # model parameters
```

```
(Intercept)         age
  94.904099    4.388803
```

```r
summary(m1)$sigma^2 # residual variance
```

```
[1] 56.19656
```

Multiple regression & interactions

LM also allow to include **multiple predictors** and the **interactions**[1] among them. This is done by estimating a separate slope (thus, a separate line) for each predictor by *holding constant* the value of the other predictors, which are fixed to zero.

Multiple regression model

β_0 = expected value in girls with `age` $= 0$

β_1 = `age` effect[2] within the same `sex`

β_2 = `sex` difference when `age` $= 0$

```
m2 <- lm(formula = height ~ age + sex,
         data = children)
coefficients(m2)
```

```
(Intercept)         age     sexmale
 95.0075706   4.3887983  -0.2001025
```

Interactive model

β_1 = `age` effect in girls

β_2 = `sex` difference in `height` when `age` $= 0$

β_3 = `sex` difference in `age` effect (**interaction**)

```
m3 <- lm(formula = height ~ age * sex,
         data = children)
round(coefficients(m3),2)
```

```
(Intercept)         age     sexmale age:sexmale
     104.25        3.70      -19.04        1.41
```

[1] The **interaction** between x_1 and x_2 is computed as the **product of** x_1 **and** x_2.

[2] In this context, “effect” is used as a synonym of “relationship” (not a *causal* effect).

Model comparison & model selection

Likelihood ratio test

Compares the *fit* of two *nested* models (i.e., predicting the same y variable, with the more complex model including all predictors included in the simpler model).

```
library(lmtest)
lrtest(m0,m1,m2,m3) # returns Chisq statistic
```

```
  #Df    LogLik Df    Chisq   Pr(>Chisq)
1   2 -10417.84 NA       NA           NA
2   3  -8582.42  1  3670.84 0.000000e+00
3   4  -8582.19  1     0.45 5.046155e-01
4   5  -8468.86  1   226.67 3.176229e-51
```

Information criteria

The Akaike (AIC) and the Bayesian Information Criterion (BIC) compare multiple models in terms of *fit & parsimony* (the lower number of parameters the better)

```
AIC(m0,m1,m2,m3) # AIC: the lower the better
```

```
[1] 20839.68 17170.83 17172.39 16947.72
```

```
# Akaike weights: from 0 (-) to 1 (+)
MuMIn::Weights(AIC(m0,m1,m2,m3))
```

```
 model weights
[1] 0 0 0 1
```

Here, *model fit to the data* is expressed by its **likelihood** = probability of observing the sampled data given the parameters estimated by the model, sometimes referred as the *evidence* of a model, or its *ability to predict/forecast* new data that are similar to the sampled data (see interactive visualization by Kristoffer Magnusson).

Parameter estimation in linear regression models

β_0 , β_1 , and ϵ must be **estimated** based on data sampled from a population: $\hat{\beta}_0 = b_0$; $\hat{\beta}_1 = b_1$; $\hat{\epsilon} = e$).

There are several methods to estimate unknown parameters, such as:

- **Ordinary least squares (OLS)**: finds the *parameter values* that *minimize the sum of the squared residuals* (default LM estimator)
- **Maximum likelihood estimator (MLE)**: finds the *parameter values* that *maximize the model likelihood*, making the observed data the most probable under that model
- **Bayesian estimator**: finds the *parameter posterior distributions* based on prior knowledge/beliefs (*prior*) and observed data (*likelihood*)

Regardless of the used method, parameters values (or distributions) are always accompanied with a measure of the **uncertainty/precision** associated with their estimate:
Standard errors (SE) = predicted *variability* in the parameter estimate if the data were collected from different random samples from the same population.
SE are used for computing *test statistics* (Est/SE) & *confidence intervals* ($Est \pm 1.96 \times SE$)

In LM, under the assumption of normally distributed residuals, OLS = MLE

What are residuals?

Residuals are the model-based estimates of the population errors.

Linear model:

$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Predicted values:

$\hat{y}_i = \beta_0 + \beta_1 x_i$

Observed values:

$y_i = \hat{y}_i + \hat{\epsilon}_i$

Residuals = observed - predicted

$\hat{\epsilon}_i = y_i - \hat{y}_i$

```
head(data.frame(observed = children$height,

                predicted = fitted(m3),

                residuals = residuals(m3)

                squared = residuals(m3)^2 ))
```

```
  observed predicted residuals squared
1   150.77    152.90     -2.13    4.55
2   170.59    156.61     13.98  195.33
3   167.31    160.31      7.00   49.01
4   165.72    165.52      0.20    0.04
5   171.67    160.31     11.36  129.06
6   143.74    151.07     -7.33   53.74
```

```
sum(residuals(m3)^2) # sum of squared (SS) residuals
```

```
## [1] 128188.3
```

```
var(residuals(m3)) # residual variance SIGMA2
```

```
## [1] 51.29585
```

In LM, **model parameters** include:

(1) intercept, (2) slope(s), and (3) **residual variance** σ^2

→ *How many parameters in the previous models? (= **No. predictors** + 2)*

Statistical inference on regression coefficients

In the NHST approach, we can **test the statistical significance** of regression coefficients (*two-tail t-test*). This is automatically done by R in the model summary.

```
summary(m3) # model results
```

```
            Estimate Std. Error t value      Pr(>|t|)
(Intercept)   104.25       0.88  118.22 0.000000e+00
age             3.70       0.06   57.45 0.000000e+00
sexmale       -19.04       1.26  -15.14 1.237494e-49
age:sexmale     1.41       0.09   15.39 3.897810e-51
```

- **Estimate** = estimated parameter
- **Std. Error** = parameter standard error
- **t value** = test statistic computed as $t = Estimate/Std.Error$
- **p-value** = p corresponding to the t-value with *No. Obs. – No. Coeff. – 1* degrees of freedom

Effect size:

Coefficient of determination

$R^2 = 1 - SS_{residuals}/SS_{total}$

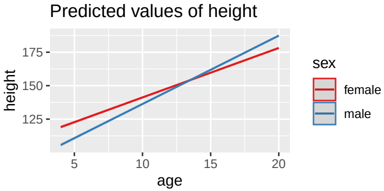
```
summary(m3)$r.squared
```

```
[1] 0.79
```

The model explains 79% of the variance in height.

Plotting effects:

```
sjPlot::plot_model(m3,type="pred",terms=c("age","sex"))
```



Hands on R

1. Download & read the dataset from the *“Pregnancy during pandemics”* study

`depr` = postnatal depression, `age` = mother’s age, `NICU` = intensive care, `threat` = fear of COVID

```r
library(osfr) # package to interact with the Open Science Framework platform
proj <- "https://osf.io/ha5dp/" # link to the OSF project
osf_download(osf_ls_files(osf_retrieve_node(proj))[2, ],conflicts="overwrite") # download
preg <- na.omit(read.csv("OSFData_Upload_2023_Mar30.csv",stringsAsFactors=TRUE)) # read data
colnames(preg)[c(2,5,12,14)] <- c("age","depr","NICU","threat") # set variable names
```

2. Explore the the variables `depr`, `threat`, `NICU`, and `age` (descr., corr., & plots)
3. Fit a null model `m0` of `depr`
4. Fit a simple regression model `m1` with `depr` being predicted by `threat`
5. Fit a multiple regression model `m2` also controlling for `NICU` and `age`
6. Fit an interactive model `m3` to check whether `age` moderates the relationship between `threat` and `depr`.
7. Compare the models with AIC and likelihood ratio test: which is the best model?
8. Print & interpret the coefficients estimated by the selected model
9. Print & interpret the statistical significance of the estimated coefficients
10. Plot the effects of the selected model
11. Compute the determination coefficient of the selected model

One step back: Linear model assumptions

Core assumptions:

1. Linearity: x_i and y_i are linearly associated → the expected (mean) value of ϵ_i is zero

2. Normality: residuals ϵ_i are normally distributed with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

3. Homoscedasticity: ϵ_i variance is constant over the levels of x_i (homogeneity of variance)

4. Independence of predictors & errors: predictors x_i are unrelated to residuals ϵ_i

5. Independence of observations: for any two observations i and j with $i \neq j$,

the residual terms ϵ_i and ϵ_j are independent (no common disturbance factors)

Additional assumptions:

6. Absence of influential observations (multivariate outliers)

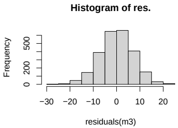
7. Absence of multicollinearity (for multiple regression):

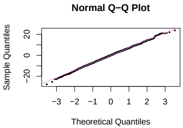
lack of linear relationship between x_1 and x_2

Model diagnostics: Assessing LM assumptions

Normality & linearity ☺

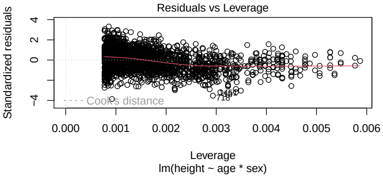
```
hist(residuals(m3))
qqnorm(residuals(m3)); qqline(residuals(m3))
```





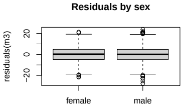
Absence of influential cases ☺

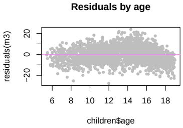
```
plot(m3,which=5)
```



Homoscedasticity & independence x, ϵ ☺

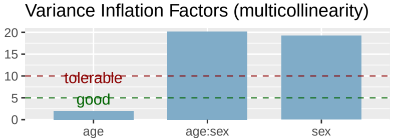
```
plot(residuals(m3) ~ children$sex)
plot(residuals(m3) ~ children$age)
```





Absence of multicollinearity ☹

```
sjPlot::plot_model(m3,"diag")[[1]]
```



Independence of observations ?

Are the unmeasured factors influencing y unrelated from one individual to another?

Cluster variables & nested data

In many cases, the *sampling method* creates **clusters** of *individual observations*

- students → schools
- children → families → neighborhoods → cities → regions → states → planets 🚀

Nested data structure (= *multilevel* or *hierarchical* data structure)
= when data points at the **individual level** appear *in only one group*
of the **cluster level** variable

→ individual observations are ***nested*** within clusters

How do you imagine such a nested dataset?

Individual observation = statistical unit = individual entity within a sample or population that is the subject of data collection & analysis (not necessarily a person)

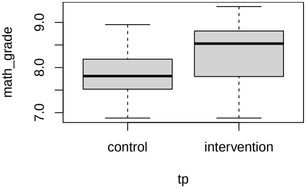
Case study: Innovative math teaching program

We're hired by a school principal to assess whether an *innovative teaching program* can improve *math achievement* in first-year high-school students.

```
# reading data
itp <- read.csv("data/studentData.csv")
# frequency table class by intervention
table(itp[,c("classID","tp")])
```

```
       tp
classID control intervention
      A      30            0
      B      22            0
      C       0           27
      D       0           11
```

```
boxplot(math_grade ~ tp, data=itp)
```



The teaching program **tp** was delivered over the first semester to 2 out of 4 classes and we got the students' end-of-semester **math_grade** (1-10).

Nested dataset: students are *nested within* classes, with each student only belonging to one class.

```
head(itp[,1:4],12)
```

```
   studID classID          tp math_grade
1      s1       A control       7.74
2      s2       A control       8.31
3      s3       A control       7.09
4      s4       A control       7.80
5      s5       A control       7.21
6      s6       A control       8.95
7      s7       A control       7.48
8      s8       A control       7.86
9      s9       A control       7.85
10    s10       A control       7.13
11    s11       A control       7.87
12    s12       A control       6.88
```

Non-independence of observations with nested data

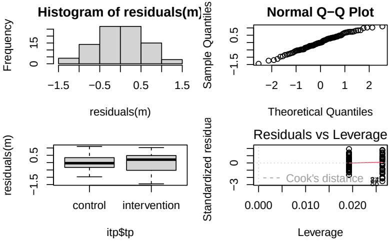
Let's try with a linear regression model:

```r
m <- lm(math_grade ~ tp, data=itp)
summary(m)$coefficients[,1:3]
```

```
##                  Estimate Std. Error t value
## (Intercept)          7.85       0.08   97.60
## tpintervention       0.48       0.12    3.87
```

Model diagnostics (see slide #11):

```r
hist(residuals(m)); qqnorm(residuals(m))
boxplot(residuals(m)~itp$tp); plot(m,5)
```



- Coefficient meaning?
- Linear model assumptions?

- **Independent observations**?

Are ϵ_i and ϵ_j independent for any $i \neq j$? Are the unmeasured factors influencing y unrelated from one individual to another?

NO: students are nested within classes and such cluster variable is likely to explain differences in the y variable (as well as in the relationship between x and y)

Thus, **we cannot rely on linear models** to analyze these data.

Local dependencies

Local dependencies = correlations that exist among observations within a specific cluster (but the software doesn't know that!)

e.g., grades from the same class will be more correlated than they are between different classes

***Why* is this a problem?**

1) Can result in **biased estimates of the standard errors** → underestimated *p*-values (+false positive)

2) Potentially important **variables at the cluster level** are neglected

e.g., teachers' characteristics, teaching CV, class social climate

***When* is this a problem?**

Virtually, any time that a cluster variable is potentially related to y

Pragmatically, we cannot account for all potential clusters

e.g., children → families → neighborhoods → cities → regions → states → planets 🚀

Based on theory & logic, we should focus on what we consider the most influential clustering factors for both y and x

Mixed-effects models

Multilevel models are part of the largest **linear mixed-effects regression (LMER)** family that include **additional variance terms** for handling local dependencies.

Why ‘mixed-effects’?

Because such additional terms come from the distinction between:

- **Fixed effects**: effects that remain ***constant across clusters***, whose levels are *exhaustively considered* (e.g., gender, levels of a Likert scale) and generally controlled by the researcher (e.g., experimental conditions)
- **Random effects**: effects that ***vary from cluster to cluster***, whose levels are *randomly sampled* from a population (e.g., schools)

When individual observations can change cluster over time, it is still a mixed-effects model but not a multilevel model.

Here, “levels” refers to the possible categories/classes of a categorical variable, but from now on we will use this term with a different meaning. . .

From LM to LMER

LM formula: $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Intercept and slope are **constant across all individual observations** i within the population; x, y, and the error term ϵ only variate across individual observations i

LMER formula: $y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij}$

Intercept and slope have both a **fixed** ($_{0/1}$) and a **random** component ($_j$); y, x, and ϵ variate across **individual observations** i as well as across **clusters** j

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij} = (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j})x + \epsilon_{ij}$$

LMER are an extension of LM where the intercept and the slope are decomposed into the **fixed components** β_{00} and β_{10} referred to the whole sample, and the **random components** λ_{0j} and λ_{1j} randomly varying across clusters.

In LMER, x **variables (predictors) always variate across clusters** j**, but not necessarily across individual observations** i (e.g., school principals' age only variate across schools, whereas students' age variate across students within schools)

Random intercept

Let's start with an **intercept-only model** (i.e., ***unconditional*** or ***null model***), where math grades (y_{ij}) are only predicted by the intercept β_{00} and the residuals ϵ_{ij}

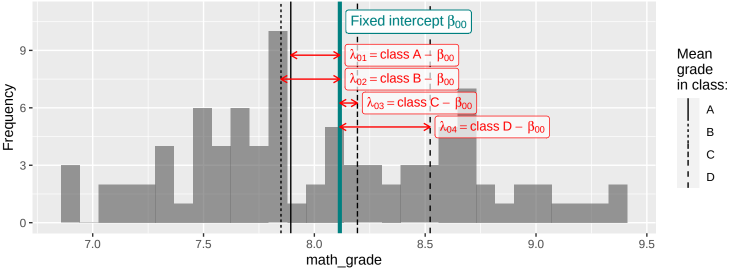
- *Linear model*: $y_i = \beta_0 + \epsilon_i$

  The intercept value β_0 is common to all individuals within the population

- *Linear mixed-effects model*: $y_{ij} = \beta_{0j} + \epsilon_{ij} = (\beta_{00} + \lambda_{0j}) + \epsilon_{ij}$

  - β_{00} is the **fixed intercept** (also called 'average' or 'general intercept') that applies to the whole population

  - λ_{0j} is the **random intercept** = *cluster-specific deviation from the fixed intercept* (i.e., mean class grade - fixed intercept)



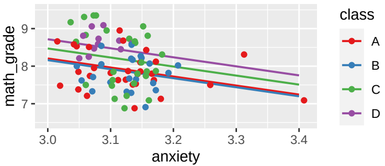
Random slope

Let's now add a predictor: students' `anxiety` levels x_{ij}.

Random intercept model

$$y_{ij} = \beta_{0j} + \beta_1 x_{ij} + \epsilon_{ij}$$
$$= (\beta_{00} + \lambda_{0j}) + \beta_1 x_{ij} + \epsilon_{ij}$$

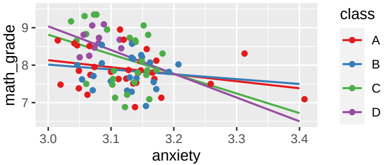
Math grades y_{ij} are predicted by the overall mean grade β_{00}, their ***average relationship*** with anxiety β_{10}, the random variation among clusters λ_{0j} (*random intercept*), and the random variation among individuals within clusters ϵ_{ij} (*residuals*).



Random intercept & **random slope** model

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij}$$
$$= (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j})\, x_{ij} + \epsilon_{ij}$$

Since the effect of anxiety might not be the same across all classes, we partition β_1 into the overall ***average relationship*** between anxiety and grades β_{10} (*fixed slope*) and the cluster-specific variation in the relationship λ_{1j} (***random slope***) - basically, an interaction between anxiety and class.



From LMER to multilevel modeling

LMER is often called *'multilevel modeling'* due to the underlying **variance decomposition** of the y_{ij} variable into the *within-cluster* and the *between-cluster* levels.

That is, the LMER formula $y_{ij} = (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j}) + \epsilon_{ij}$ can be expressed in two separate levels:

$$\begin{aligned} \textit{Level 1 (within)}: y_{ij} &= \beta_{0j} + \beta_{1j}x_{ij} + \epsilon_{ij} \\ \textit{Level 2 (between)}: \beta_{0j} &= \beta_{00} + \lambda_{0j} \\ \beta_{1j} &= \beta_{10} + \lambda_{1j} \end{aligned}$$

In some papers and textbooks, the coefficients β_{00} and β_{01} are indicated with γ_{00} and γ_{01}, while λ_{0j} and λ_{1j} are sometimes indicated with U_{0j} and U_{1j}, respectively.

That’s all for now!

Questions?

Homework (optional):

- read the slides presented today
  and write in the Moodle forum if you have any doubts
- refresh your familiarity with R: `R-intro.pdf`
- **exeRcises 1-3** from `exeRcises.pdf`

For each exercise, the solution (or one of the possible solutions) can be found in dedicated chunk of commented code within the `exeRcises.Rmd` file

In the last episode...

The problem

Sometimes the sampling method creates *clusters* of individual observations: **nested data structure** where individuals observations are *nested within* clusters.

→ **Local dependencies**
= correlations among observations within a cluster, violating the LM assumption of independence.

→ We cannot use ordinary LM

The solution

Linear mixed-effects regression (LMER) includes **additional variance terms**[1] to handle local dependencies.

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij}$$
$$= (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j})\, x_{ij} + \epsilon_{ij}$$

These can be expressed in two separate levels:

$$\begin{aligned} \textit{Level 1 (within)} : y_{ij} &= \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij} \\ \textit{Level 2 (between)} : \beta_{0j} &= \beta_{00} + \lambda_{0j} \\ \beta_{1j} &= \beta_{10} + \lambda_{1j} \end{aligned}$$

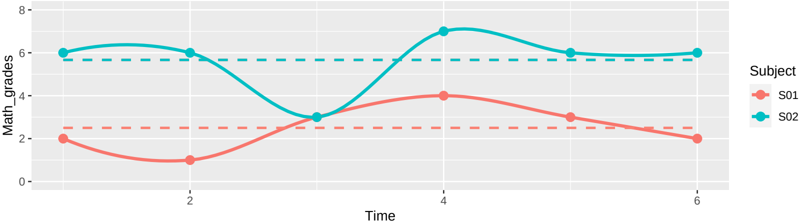
[1] The **additional variance terms** are the variance τ^2_{00} of the random intercept λ_{0j} and the variance τ^2_{10} of the random slope λ_{1j}. We will see this later...

Multilevel modeling in longitudinal designs

Longitudinal assessments (or repeated-measure designs) involve the collection of **multiple data from the same subjects at multiple time points**.

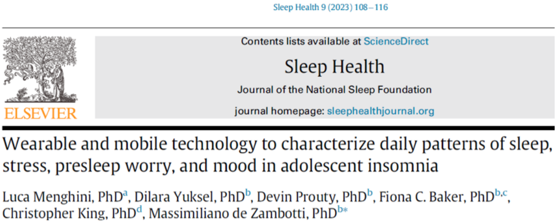
→ Observations from the same subject are not independent (*local dependencies*).

- Individual observations = time points (*level 1*: ***within-subject***)
- Clusters = subjects (*level 2*: ***between-subjects***)



If individuals are further nested within higher-level clusters, we can specify a *3-level model* (time points → students → classes)

Case study: Adolescent insomnia 🛏



Sleep Health 9 (2023) 108–116

Contents lists available at ScienceDirect

Sleep Health

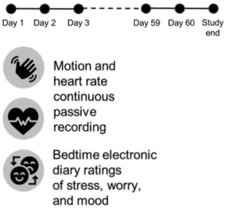
Journal of the National Sleep Foundation

journal homepage: sleephealthjournal.org

ELSEVIER

Wearable and mobile technology to characterize daily patterns of sleep, stress, presleep worry, and mood in adolescent insomnia

Luca Menghini, PhD[a], Dilara Yuksel, PhD[b], Devin Prouty, PhD[b], Fiona C. Baker, PhD[b,c], Christopher King, PhD[d], Massimiliano de Zambotti, PhD[b*]


Day 1
Day 2
Day 3
Day 59
Day 60
Study end
Motion and heart rate continuous passive recording
Bedtime electronic diary ratings of stress, worry, and mood

A sample of 93 US adolescents undertook a semi-structured clinical interview for **DSM-5 insomnia** symptomatology (*insomnia* vs. *healthy sleepers*).

Then, they were provided with a Fitbit wristband (recording **sleep** data) for 2 months. Over the same period, every evening they responded short questionnaires on their **stress** levels at bedtime.

We want to understand whether **daily stress predicts lower sleep time** (HP1); whether the stress impact on sleep is **moderated by insomnia symptomatology** (HP2).

Hands on R

1. Download & read the datasets from https://github.com/SRI-human-sleep/INSA-home

`ID` = subject ID, `dayNr` = day, `stress` = daily stress rating (1-5), `TST` = total sleep time (min), `insomnia` = subject's group (insomnia vs. healthy)

```r
repo <- "https://github.com/SRI-human-sleep/INSA-home" # loading datasets from GitHub
load(url(paste0(repo,"/raw/main/Appendix%20D%20-%20Data/emaFINAL.RData")))
load(url(paste0(repo,"/raw/main/Appendix%20D%20-%20Data/demosFINAL.RData")))
# selecting columns
ema <- ema[,c("ID","dayNr","stress","TST")] # ema = time-varying variables
demos <- demos[,c("ID","insomnia")] # demos = time-invariant variables
```

2. Print the first rows of the datasets: How many rows per subject?
3. Which variable includes individual observations, which is the cluster variable, which is the predictor?
4. Which variable(s) at the *within-cluster* level (Level 1)? Which variable(s) at the *between-cluster* level (Level 2)
5. Explore (descript., correlations, plots)
6. Compute the ***cluster mean*** for each level-1 variable using `aggregate()`
7. Join the cluster means to the `demos` dataset using `cbind()`
8. Join the cluster means to the `ema` dataset using `plyr::join()`
9. Subtract individual obs. from cluster means

Wide & Long data structure

Wide-form dataset

one row per cluster

```
clustMeans <- # computing cluster means
  aggregate(x = ema[,c("TST","stress")],
   by = list(ema$ID), FUN = mean, na.rm = T)
# join cluster means to the wide-form dataset
demos <- cbind(demos, clustMeans[,2:3])
colnames(demos)[3:4] <- c("TST.m","stress.m")
head(demos)
```

```
    ID insomnia    TST.m stress.m
1 s001        0 466.1786 1.707317
2 s002        0 431.0745 2.175000
4 s005        0 415.2059 1.872727
5 s006        1 413.1111 3.393443
6 s007        0 445.7642 1.983333
7 s008        0 422.8468 3.045455
```

Level-2 (*between*) variables:

ID, insomnia, TST.m, stress.m

Long-form dataset

one row per individual observation

```
library(plyr)
ema <- # join lv-2 variables to long-form
  join(x = ema, # long-form dataset
       y = demos, # wide-form dataset
       by = "ID", # joining variable
       type = "left") # keep all x rows
head(ema)
```

```
    ID dayNr stress   TST insomnia TST.m stress.m
1 s001     1      3 507.0        0 466.2      1.7
2 s001     2      1 502.5        0 466.2      1.7
3 s001     3      3 469.5        0 466.2      1.7
4 s001     4      2    NA        0 466.2      1.7
5 s001     5     NA    NA        0 466.2      1.7
6 s001     6      3    NA        0 466.2      1.7
```

Level-1 (*within*) variables:

dayNr, stress, TST

In R, NA values indicate **missing data**: time points where a level-1 variable was missing

Between & within cluster

Long-form dataset

one row per individual observation

```
head(ema[,-6], 20)
```

```
     ID dayNr stress   TST insomnia stress.m
1  s001     1      3 507.0        0      1.7
2  s001     2      1 502.5        0      1.7
3  s001     3      3 469.5        0      1.7
4  s001     4      2    NA        0      1.7
5  s001     5     NA    NA        0      1.7
6  s001     6      3    NA        0      1.7
7  s001     7      1    NA        0      1.7
8  s001     8      2    NA        0      1.7
9  s001     9      1    NA        0      1.7
10 s001    10      2    NA        0      1.7
11 s001    11      2    NA        0      1.7
12 s001    12      1    NA        0      1.7
13 s001    13      2    NA        0      1.7
14 s001    14      1    NA        0      1.7
15 s001    15      1    NA        0      1.7
16 s001    16     NA    NA        0      1.7
17 s001    17     NA    NA        0      1.7
18 s001    18     NA    NA        0      1.7
19 s001    19     NA 510.5        0      1.7
20 s001    20     NA 515.5        0      1.7
```

Long-form data structures are needed to fit multilevel models.

Here, **level-1 variables** x_{ij} (`stress`) and y_{ij} (`TST`) change both between and within cluster.

In contrast, **level-2 variables** x_j (`insomnia`, `stress.m`) only change between clusters, whereas they keep identical values across all the rows associated with the same cluster.

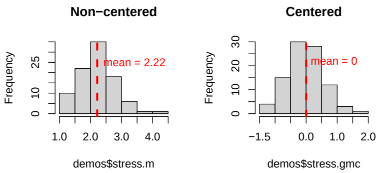
Data centering

Data centering = subtracting the mean of a variable from each variable value.

- The mean of a centered variables is always 0.
- Its variance and covariances are equivalent to those of the original variable.
- Centered scores represent *deviations from the mean*.

In both LM and LMER, **centering the predictors** is useful to *reduce collinearity* (linear relationship between predictors) and for *better interpreting a model intercept* (= value of y ***when*** x ***is at its mean***); but it *does not affect the slopes*.

```
demos$stress.gmc <- # grand-mean centering
  demos$stress.m - mean(demos$stress.m)
```



```
# non-centered x: b0 = predicted y when x = 0
coefficients(lm(TST.m ~ stress.m,data=demos))
```

```
(Intercept)    stress.m
 421.474599   -4.074498
```

```
# centered x: b0 = predicted y when x = mean x
coefficients(lm(TST.m ~ stress.gmc,data=demos))
```

```
(Intercept)  stress.gmc
 412.447988   -4.074498
```

Grand mean vs. Cluster mean centering

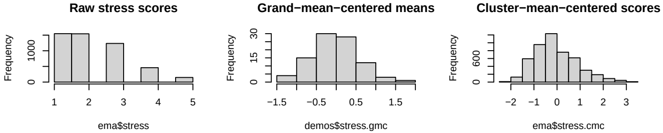
With LMER, we can distinguish two main ways to center the data:

1) Grand mean centering = subtracting the mean of the whole sample (*grand-mean* or *grand-average*) from each cluster's mean.

```r
# gmc stress = mean cluster's stress - grand mean
demos$stress.gmc <- demos$stress.m  -  mean(demos$stress.m)
```

2) Cluster mean centering (or '*group mean centering*') = subtracting the mean of the cluster (*group mean*) from each individual observation nested within that cluster.

```r
# cmc stress = individual obs. - mean of the corresponding cluster
ema$stress.cmc <- ema$stress  -   ema$stress.m
```



Hands on R: Compute the grand-mean-centered & the cluster-mean-centered values of `stress` and `TST`. Then, compute their Pearson's correlation with the `cor()` function

That's all for now!

Questions?

Homework (optional):

- read the slides presented today
  and write in the Moodle forum if you have any doubts
- **exeRcises 4-5** from `exeRcises.pdf`

For each exercise, the solution (or one of the possible solutions) can be found in dedicated chunk of commented code within the `exeRcises.Rmd` file

In the last episodes...

Problem & solution

The sampling method can create *clusters* of individual observations = *nested data* leading to *local dependencies*
→ **Multilevel modeling** (or LMER) includes *additional variance terms* to handle local dependencies.

$$\begin{aligned} \textit{Level 1 (within)}: y_{ij} &= \beta_{0j} + \beta_{1j}x_{ij} + \epsilon_{ij} \\ \textit{Level 2 (between)}: \beta_{0j} &= \beta_{00} + \lambda_{0j} \\ \beta_{1j} &= \beta_{10} + \lambda_{1j} \end{aligned}$$

Wide and long datasets

LMER require **long-form datasets**, with one row per each individual observation (level 1) and multiple rows for each cluster (level 2)

Between and within

In such datasets, **within-cluster (level-1)** variables variate both between and within clusters, while **between-cluster (level-2)** variables only variate across clusters, keeping identical values across the rows belonging to the same cluster.

Data centering & Variance decomposition

Data centering (= subtracting the mean from each variable value) can be used to decompose the variance into:

- the between-cluster component = **grand-mean-centered means**
- the within-cluster component = **cluster-mean-centered values**

The adolescent insomnia case study 🛏

A sample of 93 US adolescents undertook a semi-structured clinical interview for **DSM-5 insomnia** symptomatology (*insomnia* vs. *healthy sleepers*).
Then, they were provided with a Fitbit wristband (recording **sleep** data) for 2 months. Over the same period, every evening they rated their **stress** (1-5) at bedtime.

We want to test whether **day-to-day fluctuations** in **stress** predict **lower total sleep time** `TST` (HP1),
and whether the stress impact on `TST` is **moderated by `insomnia` symptomatology** (HP2).

```r
load("insa.RData") # read processed data
insa[,c("ID","TST","TST.m","TST.gmc","TST.cmc")]
```

```
     ID   TST  TST.m TST.gmc TST.cmc
1  s001 507.0 466.18   53.73   40.82
2  s001 502.5 466.18   53.73   36.32
3  s001 469.5 466.18   53.73    3.32
21 s001 496.0 466.18   53.73   29.82
22 s001 447.5 466.18   53.73  -18.68
23 s001 450.5 466.18   53.73  -15.68
24 s001 423.0 466.18   53.73  -43.18
29 s001 483.5 466.18   53.73   17.32
30 s001 450.0 466.18   53.73  -16.18
31 s001 529.0 466.18   53.73   62.82
```

`TST` = raw total sleep time (minutes)

`TST.gmc` = grand-mean-centered cluster means of TST (**level-2 component**)

`TST.cmc` = cluster-mean-centered TST (**level-1 component**)

Descriptive statistics of multilevel data

The **first section of the results section** in any quantitative report (including published papers) includes the **descriptive statistics** of the considered variables in the examined sample. Descriptive statistics are also the main output of any quantitative report you might draft or read in your **professional practice**.

With mutlilevel datasets, the descriptive statistics to be reported are the following:

1. **Mean and SD** of any considered quantitative variable
2. **Frequency (%)** of any considered categorical variable
3. **Level-specific correlations** among quantitative variables
4. **Intraclass correlation coefficient (ICC)** of any quantitative variable measured at the *within-cluster* level

Compute descriptive statistics 1-3, considering the variables `TST`, `stress`, and `insomnia` (*Note*: correlations can be computed with the `cor()` function; level-2 correlations should be computed on the cluster means in the `demos` dataset)

Response rate (or missing data) is a further important descriptive to report. Here, for simplicity, we omitted missing data points from the `insa` dataset.

Level-specific correlations

Between-cluster (level 2)

Cluster means

Level-2 correlation

= linear relationship **across clusters**

Do stressed subjects sleep worse *than unstressed subjects?*

```
wide <- insa[!duplicated(insa$ID),]
cor(wide[,c("stress.m", "TST.m")])
```

```
         stress.m  TST.m
stress.m    1.000 -0.067
TST.m      -0.067  1.000
```

Within-cluster (level 1)

Individual *deviations* from cluster mean

= cluster-mean-centered values

Level-1 correlation

= linear relationship **within cluster**

Do subjects ***sleep worse than usual*** *in those days where they are* ***more stressed than usual?***

```
cor(insa[,c("stress.cmc", "TST.cmc")])
```

```
           stress.cmc TST.cmc
stress.cmc       1.00   -0.06
TST.cmc         -0.06    1.00
```

Additional variance (& covariance) terms

LMER includes **additional variance and covariance terms** to handle local dependencies. → *Variance and covariance what?!*

Rembember the LMER formula:

$y_{ij} = (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j})x_{ij} + \epsilon_{ij}$

λ_{0j} are the random deviations of *cluster intercepts* from the *fixed intercept* β_{00}

λ_{1j} are the random deviations of *cluster slopes* from the *fixed slope* β_{10}

ϵ_{ij} is the **residual term** indicating the random deviations of *observed values* from *predicted values* (see slide #8)

In both LM and LMER, we don't report each single residual value ϵ_{ij}, but we use

σ^2 **= variance of the residuals** ϵ

Similarly, in LMER we summarize the random effects by reporting their variances:

τ_{00}^2 **= variance of random intercept** $\lambda_0 j$

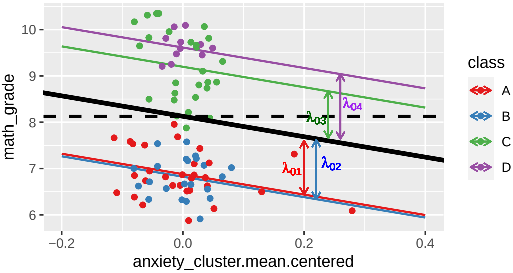
τ_{11}^2 **= variance of random slope** λ_{1j}

Moreover, when both λ_{0j} and λ_{1j} are included, we need to also consider the covariance term:

ρ_{01} **= covariance between** λ_{0j} **and** λ_{1j}

→ τ_{00}^2, τ_{11}^2, ρ_{01} *are the additional variance & covariance terms included in LMER*

Random intercept and random slope (1/2)



- **Random intercept (RI)**

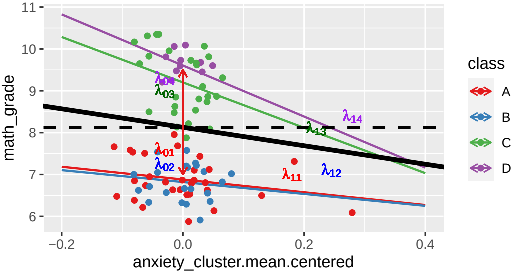
  $y_{ij} = (\beta_{00} + \lambda_{0j}) + \beta_1 x_{ij} + \epsilon_{ij}$

  RI = distances between each cluster's intercept and the **fixed intercept**

  Parallel lines: there is no random slope

  τ^2_{00}= variance of the RI (how much the RI differ among each other)

  $= \text{var}(\lambda_{01}, \lambda_{02}, \lambda_{03}, \lambda_{04},) = 2.22$



- **RI and random slope (RS)**

  $y_{ij} = (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j})x_{ij} + \epsilon_{ij}$

  RS = distances between each cluster's slope and the **fixed slope**

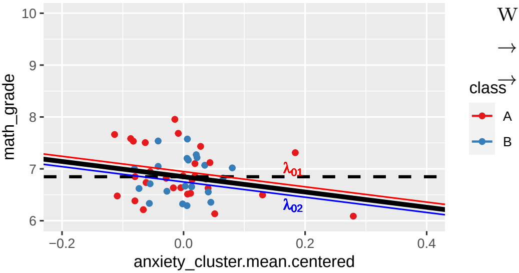
  τ^2_{00}= variance of the RI = 2.22

  τ^2_{10}= variance of the RS

  $= \text{var}(\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14},) =$ -3.73

  ρ_{01}= covariance between λ_{0j} & λ_{1j}

Random intercept & random slope (2/2)



What happens if we remove class C and D?

→ Both random effects become smaller

→ **lower variance τ_{00} and τ_{10}**

- **Random intercept (RI)**

  $y_{ij} = (\beta_{00} + \lambda_{0j}) + \beta_1 x_{ij} + \epsilon_{ij}$

  Class A and class B's intercepts are very close, their distances from the **fixed intercept** are very small

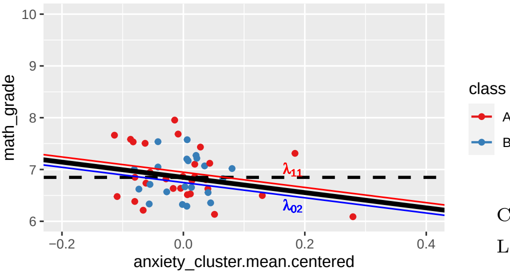
  $\lambda_{01} \sim \lambda_{02} \rightarrow \tau^2_{00} \sim 0$

- **RI and random slope (RS)**

  $y_{ij} = (\beta_{00} + \lambda_{0j}) + (\beta_{10} + \lambda_{1j}) x_{ij} + \epsilon_{ij}$

  Class A and class B's slopes are very close → their distances from the **fixed slope** are very small

  $\lambda_{01} \sim \lambda_{02} \rightarrow \tau^2_{00} \sim 0$



Conclusions: It makes no sense to use LMER (better using LM!)

Null model & variance decomposition (1/2)

A **null model** only includes the intercept and residual terms (see slide #20).

In **LM null models** $(y_i = \beta_0 + \epsilon_i)$

the intercept β_0 is simply the mean of y_i,

and the variance of ϵ_i (σ^2) is simply the variance of y_i.

```r
lm.fit <- lm(TST ~ 1, data = insa)
c(b0 = coefficients(lm.fit), mean_Y = mean(insa$TST, na.rm = TRUE))
```

```
     b0  mean_Y
413.686 413.686
```

```r
c(sigma2 = var(residuals(lm.fit)), var_Y = var(insa$TST, na.rm = TRUE))
```

```
  sigma2    var_Y
6291.752 6291.752
```

In **LMER null models** $(y_{ij} = \beta_{00} + \lambda_{0j} + \epsilon ij)$

the y **variance is decomposed** into:

- the variance σ^2 of the residuals ϵ_{ij} across **both levels**
- the between-cluster (level-2) variance τ_{00}^2 = variance of the random intercept λ_{0j}

Null model & variance decomposition (2/2)

Spoiler alert: How to fit LMER in R

```r
# fitting a null LMER model
library(lme4)
m0 <- lmer(TST ~ (1|ID), data = insa)
summary(m0)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: TST ~ (1 | ID)
   Data: insa

REML criterion at convergence: 49553.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4233 -0.6134 -0.0285  0.5760  5.6047

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept) 1183     34.39
 Residual             5158     71.82
Number of obs: 4333, groups:  ID, 93

Fixed effects:
            Estimate Std. Error t value
(Intercept)  410.838      3.769     109
```

If we inspect the summary of a null LMER model, starting from the bottom, we can see that:

- **Fixed effects** only include the *fixed intercept* β_{00} (= 410.838 minutes).

- **Random effects** include variance & SD of the *random intercept* λ_{0j} ($\tau_{00}^2 = 1183$) and that of the *residuals* ϵ_{ij} ($\sigma^2 = 5158$).

The sum $\sigma^2 + \tau_{00}^2$ of the residual (level-1) and the random intercept variance (level-2) is the **model estimate of the population-level total variance in** y_{ij}

Variance decomposition & Data centering

The **variance decomposition** implemented by LMER is basically equivalent to the **data centering procedures** shown in the last lecture (see slide #32).

```r
# random intercept LAMBDA_0j
round(head(  ranef(m0)$ID[[1]]  ),1)
```

```
[1] 50.0  6.2  4.7  4.1 31.1  7.9
```

```r
# random intercept variance TAU^2
(tau2 <- round(summary(m0)$varcor$ID[[1]]))
```

```
[1] 1183
```

```r
# residual variance SIGMA^2
(sigma2 <- summary(m0)$sigma^2)
```

```
[1] 5157.676
```

```r
# estimated total variance in TST
tau2 + sigma2
```

```
[1] 6340.676
```

```r
# grand-mean-centered TST cluster means
round(head(  wide$TST.gmc  ),1)
```

```
[1] 53.7 18.6  2.8  0.7 33.3 10.4
```

```r
# variance of TST cluster means
var(wide$TST.m)
```

```
[1] 1241.19
```

```r
# variance of cluster-mean-centered TST
var(insa$TST.cmc, na.rm=TRUE)
```

```
[1] 5072.426
```

```r
# observed total variance in TST
var(insa$TST, na.rm=TRUE)
```

```
[1] 6291.752
```

The small differences between model-based (on the left) and observed values (on the right) are due to slight adjustments (e.g., accounting for the number of clusters) used by LMER models (for details, see Finch & Bolin, 2014, chapter 2)

Intraclass correlation coefficient (ICC)

The last ‘descriptive’ statistics to be reported is the ICC

= Proportion of between-cluster variance over the total variance

The ICC is *estimated from the null model* as $ICC = \tau_{00}^2/(\tau_{00}^2 + \sigma^2)$ and can range between 0 and 1.

- **ICC = 1**: the variable *only* varies *across* clusters (‘cluster-only variable’)
- **0.50 < ICC < 1**: the variable *mainly* varies *across* clusters
- **ICC = 0.50**: the variable *equally* varies across & within clusters
- **0 < ICC < 0.50**: the variable *mainly* varies *within* clusters*
- **ICC = 0**: the variable *only* varies *within* cluster (‘individual-only variable’)

The ICC is important in multilevel modeling, because it indicates the *degree to which the nested data structure may impact a level-1 variable* → it **indexes of the local dependencies** implied by the nested data structure.

The ICC is an estimate of the population param. ρ_I but I think you're done with Greek letters :)

Descriptive statistics of multilevel data

Now we have all the core descriptive statistics! ☺

Variable	Mean (SD)/Freq. (Prop.)	ICC	1.	2.
1. TST (minutes)	413.69 (79.32)	0.19	1.00	-0.06
2. Stress (1 - 5)	2.21 (1.06)	0.26	-0.07	1.00
3. Insomnia group	47 (50.54%)	NA	NA	NA

Note: lv-1 and lv-2 correlations are shown below and above the main diagonal, respectively. In this case, the two variable are not so correlated at any level ☹

Hands on

1. Download and read the file `studentData.csv`
2. DESC: Compute the mean and SD of `anxiety` and `math_grade`; compute the number of students per `classID`
3. Compute the **cluster mean** for `anxiety` using `aggregate()` → wide-form
4. Join the cluster means to the long-form: `plyr::join(long,wide,by="cluster")`
5. Compute the **cluster-mean-centered** values of `anxiety`
6. Repeat points 4-5 for `math_grade`
7. DESC: Compute the **between-cluster (lv2) correlation** from the wide-form dataset (1 row per cluster)
8. DESC: Compute the **within-cluster (lv1) correlation** from the long-form dataset (1 row per individual obs.)
9. Fit a null multilevel model with the `lme4` package:

   ```
   m0 <- lmer(y ~ (1|cluster), data)
   ```

   and get σ^2: `summary(m0)$sigma^2`

   and τ_{00}^2: `summary(m0)$varcor$ID[[1]]`
10. DESC: Compute and interpret the ICC $= \tau_{00}^2 / (\tau_{00}^2 + \sigma^2)$

That's all for now!

Questions?

Homework (optional):

- read the slides presented today
  and write in the Moodle forum if you have any doubts
- **exeRcises 6-7** from `exeRcises.pdf`

For each exercise, the solution (or one of the possible solutions) can be found in dedicated chunk of commented code within the `exeRcises.Rmd` file

In the last episodes. . .

Problem & solution

The sampling method can create *clusters* of individual observations = *nested data* leading to *local dependencies*

→ **Multilevel modeling** (or LMER) includes *additional variance (and covarariance) terms* for local dependencies.

$$Level\ 1\ (within): y_{ij} = \beta_{0j} + \beta_{1j}x_{ij} + \epsilon_{ij}$$

$$Level\ 2\ (between): \beta_{0j} = \beta_{00} + \lambda_{0j}$$

$$\beta_{1j} = \beta_{10} + \lambda_{1j}$$

Wide and long datasets

LMER require **long-form datasets**, with one row per each individual observation (level 1) and multiple rows for each cluster (level 2)

Variance decomposition

LMER automatically *decompose the Y variance* into its **within-cluster (lv1)** and **between-cluster (lv2)** components.

Similarly, we can use *data centering* to better express *predictors* (X variables) at level 1 (cluster mean centering) or at level 2 (cluster means).

Descriptive statistics

- Mean (SD) / Freq. of any variable
- Level-specific correlations
- $ICC = \tau^2_{00}/(\tau^2_{00} + \sigma^2)$

indexing the ***proportion of level-2 variance***, where τ^2_{00} is the variance of the random intercept β_{00} (lv2) and σ^2 is the variance of the residuals ϵ_{ij} (lv1) from a *null model*

Fitting multilevel models (in R): Null model

We will use the `lme4` **package** (Bates et al 2014), which uses the `lmer()` function to fit linear models the exact same way of `lm()` (i.e., `formula` & `data` arguments).

```r
library(lme4) # loading package
```

Ordinary linear model (LM)

`TST` is predicted by the **intercept** β_0 (expected value of `TST` in the sample = grand average) & the **residual variance** σ^2, without accounting for local dependencies and the multilevel data structure.

```r
lm0 <- lm(formula = TST ~ 1,
         data = insa)
coefficients(lm0) # intercept
```

```
(Intercept)
    413.686
```

```r
summary(lm0)$sigma^2 # residual variance
```

```
[1] 6291.752
```

An alternative R package to fit LMER is the `nlme` package (see Finch & Bolin, 2014).

Fitting multilevel models (in R): Null model

We will use the `lme4` **package** (Bates et al 2014), which uses the `lmer()` function to fit linear models the exact same way of `lm()` (i.e., `formula` & `data` arguments).

```
library(lme4) # loading package
```

Ordinary linear model (LM)

`TST` is predicted by the **intercept** β_0 (expected value of `TST` in the sample = grand average) & the **residual variance** σ^2, without accounting for local dependencies and the multilevel data structure.

```
lm0 <- lm(formula = TST ~ 1,
          data = insa)
coefficients(lm0) # intercept
```

```
(Intercept)
    413.686
```

```
summary(lm0)$sigma^2 # residual variance
```

```
[1] 6291.752
```

Multilevel model (LMER)

`TST` is predicted by the **fixed intercept** β_{00} (lv2), the variance of the **random intercept** τ_{00}^2 (lv2), & the **residual variance** σ^2 (lv1).

```
lmer0 <- lmer(formula = TST ~ (1|ID),
              data = insa)
fixef(lmer0) # fixed effects
```

```
(Intercept)
   410.8383
```

```
summary(lmer0)$varcor$ID[[1]] # RI variance
```

```
[1] 1182.746
```

```
summary(lmer0)$sigma^2 # residual variance
```

```
[1] 5157.676
```

🔬 An alternative R package to fit LMER is the `nlme` package (see Finch & Bolin, 2014).

Random intercept (RI) model

A **RI model** can include 1+ predictors, but their effect does not variate across clusters.

Ordinary linear model (LM)

TST is predicted by the **intercept** β_0 (expected value when stress.cmc $= 0$),

the **slope** β_1 (indexing the predicted change in TST for a 1-unit increase in stress.cmc),

and the **residual variance** σ^2.

```
lm1 <- lm(formula = TST ~ stress.cmc,
          data = insa)
coefficients(lm1) # intercept & slope
```

```
(Intercept)  stress.cmc
 413.701214   -4.762748
```

```
summary(lm0)$sigma^2 # residual variance
```

```
[1] 6291.752
```

Multilevel model (LMER)

TST is predicted by the **fixed intercept** β_{00} (lv2), the variance of the **RI** τ^2_{00} (lv2), the **slope** β_1 (same meaning than in LM), & the **residual variance** σ^2 (lv1).

```
lmer1 <-
  lmer(formula = TST ~ stress.cmc + (1|ID),
       data = insa)
fixef(lmer1) # fixed effects
```

```
(Intercept)  stress.cmc
 410.848597   -4.920536
```

```
summary(lmer1)$varcor$ID[[1]] # RI variance
```

```
[1] 1186.171
```

```
summary(lmer1)$sigma^2 # residual variance
```

```
[1] 5137.951
```

Note that we are using the **cluster-mean-centered** predictor stress.cmc to focus on level 1!

Random slope (RS) model

In a **RS model** the effect of 1+ level-1 predictors randomly varies across clusters.

Random intercept (RI) model

The within-individual effect of `stress` on `TST` is **fixed across clusters**. The model only includes a **fixed slope** β_1 indexing the overall relationship between the two variables.

```
lmer1 <-
  lmer(TST ~ stress.cmc + (1|ID),
       data = insa)
fixef(lmer1) # fixed effects
```

```
(Intercept)  stress.cmc
 410.848597   -4.920536
```

```
summary(lmer1)$varcor$ID[[1]] # RI var
```

```
[1] 1186.171
```

```
summary(lmer1)$sigma^2 # residual var
```

```
[1] 5137.951
```

Random slope (RS) model

In a **RS model** the effect of 1+ level-1 predictors randomly varies across clusters.

Random intercept (RI) model

The within-individual effect of `stress` on `TST` is **fixed across clusters**. The model only includes a **fixed slope** β_1 indexing the overall relationship between the two variables.

```
lmer1 <-
  lmer(TST ~ stress.cmc + (1|ID),
       data = insa)
fixef(lmer1) # fixed effects
```

```
(Intercept)  stress.cmc
 410.848597   -4.920536
```

```
summary(lmer1)$varcor$ID[[1]] # RI var
```

```
[1] 1186.171
```

```
summary(lmer1)$sigma^2 # residual var
```

```
[1] 5137.951
```

Random slope (RS) model

The effect of **stress varies across clusters**. The model also includes the **RS variance** τ_{10}^2 and the **covariance** ρ_{01} between RI and RS.

```
lmer2 <-
  lmer(TST ~ stress.cmc + (stress.cmc|ID),
       data = insa)
fixef(lmer2) # fixed effects
```

```
(Intercept)  stress.cmc
 410.909025   -5.685554
```

```
# RI variance, RS variance, RI-RS covariance
matrix(summary(lmer2)$varcor$ID)[c(1,4,2),]
```

```
[1] 1183.70745   87.26116   21.22170
```

```
summary(lmer2)$sigma^2 # residual variance
```

```
[1] 5071.189
```

lmer() synthax: Random intercept & random slope

From the previous examples, we saw that `lmer()` includes an additional term using the syntax `(1 | cluster_variable)`, standing for the *random intercept*:

```
lmer(formula = TST ~ stress.cmc + (1 | ID), data = insa)
```

If we replace the value 1 in the first term between brackets with the name of a level-1 predictor included in the model, we get `(predictor | cluster_variable)`, standing for *the random intercept* ***and*** *the random slope*:

```
lmer(formula = TST ~ stress.cmc + (stress.cmc | ID), data = insa)
```

It is also possible to add further level-1 and level-2 predictors (*multiple regression*)

```
lmer(TST ~ stress.cmc + x2 + x3 + x4 + ... + (stress.cmc | ID), data = insa)
```

. . . and their *interactions*:

```
lmer(TST ~ stress.cmc + x2 + x2:stress.cmc + (stress.cmc | ID), data = insa)
```

`lmer()` also allows to include **multiple random intercepts** e.g., `(1 | j1) + (1 | j2/j3)` and **multiple random slopes** e.g., `(s1 | j1) + (s2 | j1) + (s1 + s2 | j2)`.

Hands on R (adolescent insomnia, again! 😆)

1. Download & read the pre-processed dataset `insa.RData` (omitting missing data)

`TST` = total sleep time (min), `stress.cmc` = cluster-mean-centered stress (1-5),

`insomnia` = insomnia group, `ID` = participant identifier

```r
getwd() # get where your working directory is, and save the data file in it
load("insa.RData") # read data
```

2. Mean, SD, correlations & plots
3. Fit a null LMER model `m0` of `TST` and compute the ICC
4. Fit a model `m1` with `TST` being predicted by `stress.cmc`
5. Fit a model `m2` with a random slope for `stress.cmc`
6. Inspect the `summary()` of each model:
   Is there a substantial within-individual relationship between `TST` and `stress` (***hypothesis 1***)
7. Fit a model `m3` that also includes `insomnia` group differences:
   Any group differences?
   Does it change the effect of `stress`?
8. Fit a model `m4` that also includes **the interaction** between `insomnia` and `stress.cmc`
9. Inspect the `summary()` of of model `m4`:
   Does `insomnia` moderate the within-individual relationship between `stress` and `TST`? (***hypothesis 2***)

lmer() model summary

Here we print and comment the summary of the interactive model `m4`.

```r
m4 <- lmer(TST ~ stress.cmc * insomnia + (stress.cmc|ID), data = insa)
```

```r
summary(m4)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: TST ~ stress.cmc * insomnia + (stress.cmc | ID)
   Data: insa

REML criterion at convergence: 49511.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4787 -0.6086 -0.0211  0.5756  5.5474

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 ID       (Intercept) 1196.32  34.588
          stress.cmc    86.44   9.297   0.06
 Residual             5071.75  71.216
Number of obs: 4333, groups:  ID, 93

Fixed effects:
                     Estimate Std. Error t value
(Intercept)           409.505      5.395  75.900
stress.cmc             -7.187      2.290  -3.138
insomnia1               2.759      7.572   0.364
stress.cmc:insomnia1    2.923      3.188   0.917
```

- **First lines**: model formula, data, and parameter estimation method (here, REML), info on estimation convergence
- **Scaled residuals**: descriptives of the model residuals
- **Random effects**: estimated variance (τ_{00}^2, τ_{10}^2), SD (τ_{00}, τ_{10}), and correlation (ρ_{10}) of random intercept and random slope, residual variance (σ^2) and SD (σ)
- Number of individual observations (lv1) and clusters (lv2) used by the model
- **Fixed effects**: fixed intercept and fixed slope for `stress`, `insomnia`, and their interaction (i.e., product)

LMER coefficient interpretation

Here, we interpret the fixed coefficients estimated by model `m4`.

```r
round( summary(m4)$coefficients, 1) # fixed effects part of the summary
```

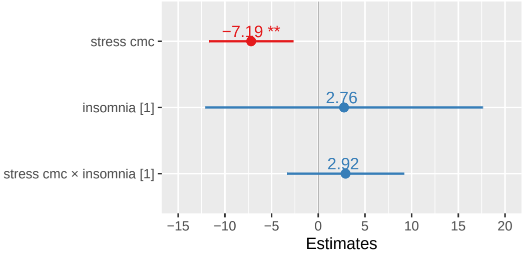
```
                     Estimate Std. Error t value
(Intercept)             409.5        5.4    75.9
stress.cmc               -7.2        2.3    -3.1
insomnia1                 2.8        7.6     0.4
stress.cmc:insomnia1      2.9        3.2     0.9
```

- **Fixed intercept**: the predicted value of `TST` when `stress.cmc = 0` (*average stress level*) and `insomnia = 0` (*controls = reference group*) is 409.5 minutes.
- **Fixed `stress` slope**: when `insomnia = 0` (*controls*), `TST` is predicted to decrease by -7.2 minutes for each 1-point increase in `stress.cmc` (*more stressed than usual*).
- **Fixed `insomnia` slope**: when `stress.cmc = 0` (*average stress*), the `insomnia` is expected to show an average `TST` of 2.8 minutes higher than the control group.
- **Interaction**: when `insomnia = 1`, the `stress`-related decrease in `TST` is predicted to be reduced by 2.9 minutes (i.e., -7.2 + 2.9 = -4.3 minutes per 1-unit increase in `stress`).
- ***t* values** ($= Estimate/Std.Error$) suggest that `stress.cmc` (*higher stress than usual*) predicts lower `TST` ($|t| > 1.96$), but their relationship does not change across the insomnia and the control group ($|t| < 1.96$) → HP1 supported, HP2 not supported

Visualizing fixed estimates & standard errors

🌲 Forest plot: The `plot_model()` function of the `sjPlot` package allows visualizing **fixed estimates** (dots) with their **95% confidence intervals (CI)** = $\mathit{Estimate} \pm 1.96\ \mathit{Std.Err.}$ indexing the precision of the estimate value (line limits).

```
sjPlot::plot_model(m4, show.values=TRUE)
```



Interpretation:

- Consistently with the previous slide, the only **95% CI excluding zero** are those of `stress.cmc` (*in line with HP1 but not HP2*).

- The `insomnia` estimate (lv2) varies more than that of `stress` (lv1) - also due to the *lower sample size at the between-cluster level*

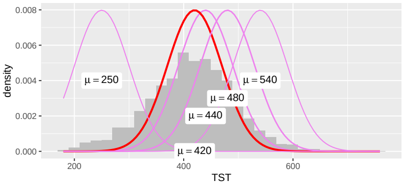
Both 95% CI and the t-value are derived from the **standard error (SE)** = predicted variability in the estimate if the data were collected from different random samples.

Parameter estimation in LMER

LMER coefficients and SE can be estimated with various methods (or algorithms), including the Bayesian estimator (see slide #7), but the most used are MLE and REML.

Maximum Likelihood Estimation (MLE)

Finds the *combination of parameter values* that *maximize the likelihood function* (= probability of observing our data given the model) using an iterative approach (the model is repeatedly fitted with different parameter values until the maximum is identified).



Restricted Maximum Likelihood (REML)

Similar to MLE, but estimates the *variance components* in a different way:

- **MLE** firstly estimates the mean μ and then the variance (as the distance from μ), but this was found to **underestimate the variance**
- **REML** applies a correction based on the number of fixed coefficients to get **less biased variance estimates**

Since variance components are critical in LMER (random effects), REML is generally preferred (default in R), but with large sample they are basically the same.

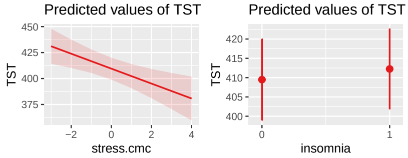
Extra resources: <MLE funny video>; <MLE interctive tutorial>; <REML video>

Visualizing fixed and random effects

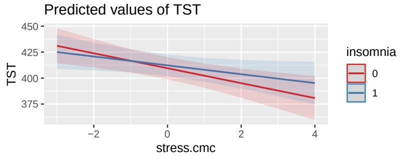
The `plot_model()` function also allows to visualize fixed and random effects.

Fixed effects Regression line & 95% CI

```r
plot_model(m4, type = "pred") # main effects
```



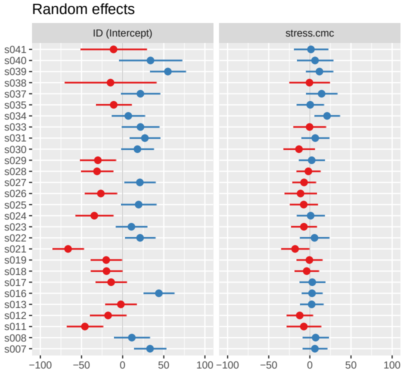
```r
plot_model(m4, type = "int") # interaction
```



Random effects

🌲 Estimate & 95% CI

```r
plot_model(m4, type = "re")
```



LMER results in a scientific paper/report

While the output of `summary()` is quite exhaustive, it slightly differs from what typically reported in scientific papers/reports. The `tab_model()` from `sjPlot` provides such a format.

You should now be able to understand the meaning of any reported value.

`sjPlot` calls random effect variances τ rather than τ^2.

```
tab_model(m4, show.se=TRUE, collapse.se=TRUE, string.est="b (SE)")
```

Predictors	b (SE)	CI	p
(Intercept	409.50 (5.40)	398.93 – 420.08	**<0.001**
Stress	-7.19 (2.29)	-11.68 – -2.70	**0.002**
Group [Insomnia]	2.76 (7.57)	-12.09 – 17.60	0.716
Stress × Group [Insomnia]	2.92 (3.19)	-3.33 – 9.17	0.359
Random Effects			
σ^2	5071.75		
τ_{00} ID	1196.32		
τ_{11} ID.stress.cmc	86.44		
ρ_{01} ID	0.06		
N $_{\text{ID}}$	93		
Observations	4333		

That’s all for now!

Questions?

Homework (optional):

- read the slides presented today
  and write in the Moodle forum if you have any doubts
- **exeRcises 8-9** from `exeRcises.pdf`

For each exercise, the solution (or one of the possible solutions) can be found in dedicated chunk of commented code within the `exeRcises.Rmd` file

Credits

The present slides are partially based on:

- Altoè, G. (2023) Corso Modelli lineari generalizzati ad effetti misti - 2023. https://osf.io/b7tkp/
- Beaujean, A. A. (2014) Latent Variable Modeling Using R. A Step-by-Step Guide. New york: Routledge
- Finch, W. H., Bolin, J. E., Kelley, K. (2014). Multilevel Modeling Using R (2nd edition). Boca Raton: CRC Press
- Pastore, M. (2015). Analisi dei dati in psicologie (e applicazioni in R). Il Mulino.

Useful resources on multilevel modeling

- Bates, D. (2022). lme4: Mixed-effects modeling with R. https://stat.ethz.ch/~maechler/MEMo-pages/lMMwR.pdf
- Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language, 59*(4), 390-412.
- Bliese, P. (2022). Multilevel modeling in R (2.7). https://cran.r-project.org/doc/contrib/Bliese_Multilevel.pdf
- McElreath, R. (2020). Statistical rethinking: A Bayesian course with examples in R and Stan. Chapman and Hall/CRC.
- Pinheiro, J., & Bates, D. (2006). Mixed-effects models in S and S-PLUS. Springer science & business media.

Papers on specific topics

Information criteria

- Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control, 19*(6), 716-723. https://doi.org/10.1109/TAC.1974.1100705
- Vrieze, S. I. (2012). Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods, 17*(2), 228. https://psycnet.apa.org/doi/10.1037/a0027127

Online resources on specific topics

- Jason Fernando (2023) R-Squared: Definition, Calculation Formula, Uses, and Limitations. Available at this link

Achronyms & Greek letters

- AIC: Akaike Information Criterion
- BIC: Bayesian Information Criterion
- ICC: intraclass correlation coefficient
- LM: linear models
- CI: confidence intervals
- MLE: maximum likelihood estimator
- OLS: ordinary least squares
- NHST: null hypothesis significance testing
- SD: standard deviation
- SE: standard error
- SS: sum of squares

- β = *beta*, indexing population-level intercept (β_0) and slope (β_1, β_2, etc.) parameters
- ϵ = *epsilon*, indexing population-level errors to be estimated based on model residuals
- λ = *lambda*, indexing random effects (cluster-specific deviation from fixed coefficients)
- σ = *sigma*, indexing the variance σ^2 of population-level errors (or model residual)
- $\mathcal{N}$ = capital *nu*, indexing that a variable is normally distributed
- ρ = *rho*, indexing the correlation between random effects
- τ = *tau*, indexing the variance of the random effects

Achronyms & Greek letters

- AIC: Akaike Information Criterion
- BIC: Bayesian Information Criterion
- ICC: intraclass correlation coefficient
- LM: linear models
- CI: confidence intervals
- MLE: maximum likelihood estimator
- OLS: ordinary least squares
- NHST: null hypothesis significance testing
- SD: standard deviation
- SE: standard error
- SS: sum of squares

- β = *beta*, indexing population-level intercept (β_0) and slope (β_1, β_2, etc.) parameters
- ϵ = *epsilon*, indexing population-level errors to be estimated based on model residuals
- λ = *lambda*, indexing random effects (cluster-specific deviation from fixed coefficients)
- σ = *sigma*, indexing the variance σ^2 of population-level errors (or model residual)
- $\mathcal{N}$ = capital *nu*, indexing that a variable is normally distributed
- ρ = *rho*, indexing the correlation between random effects
- τ = *tau*, indexing the variance of the random effects
- ciao