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

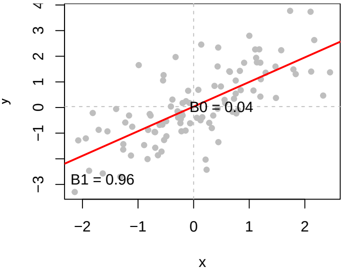
Ⓡ = exercises with R (bring your laptop!)

LM recap: 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

```r
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**)

```r
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).

```r
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)

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

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

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

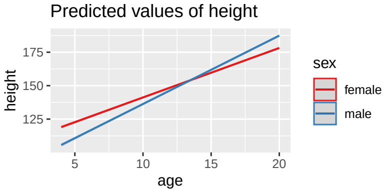
```
summary(m3)$r.squared
```

```
[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)

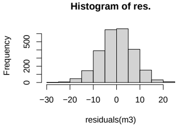
7. Absence of multicollinearity (for multiple regression):

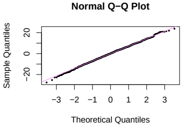
lack of linear relationship between x_1 and x_2

Model diagnostics: Assessing LM assumptions

Normality & linearity ☺

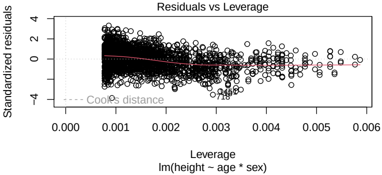
```
hist(residuals(m3))
qqnorm(residuals(m3)); qqline(residuals(m3))
```





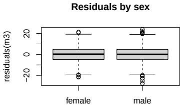
Absence of influential cases ☺

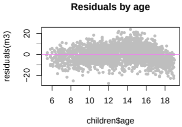
```
plot(m3,which=5)
```



Homoscedasticity & independence x, ϵ ☺

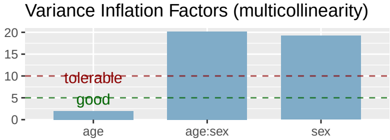
```
plot(residuals(m3) ~ children$sex)
plot(residuals(m3) ~ children$age)
```





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)

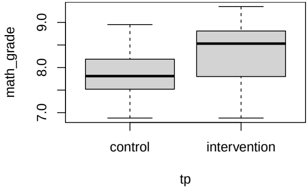
Case study: Innovative math teaching program

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

```r
# 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
```

```r
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.

```r
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

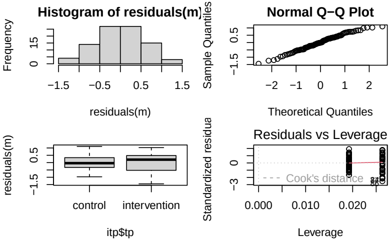
Let's try with a linear regression model:

```
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):

```
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}

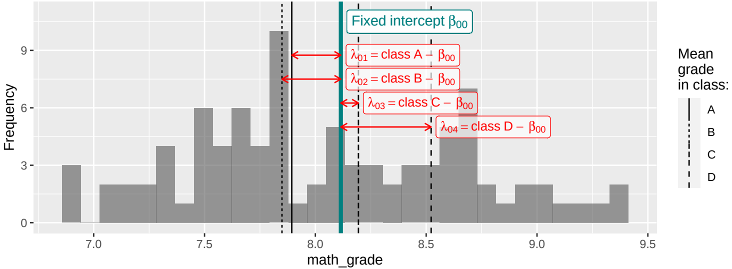
- *Linear model*: $y_i = \beta_0 + \epsilon_i$

  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

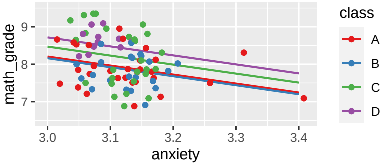
Let's now add a predictor: students' `anxiety` levels x_{ij}.

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 slope

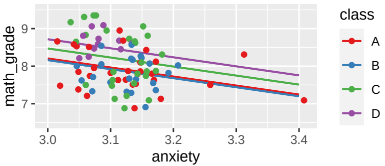
Let's now add a predictor: students' `anxiety` levels x_{ij}.

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*).

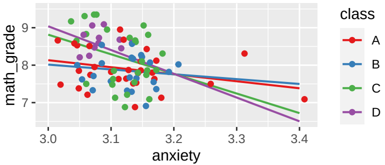


Random intercept & **random slope** model

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

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

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:

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

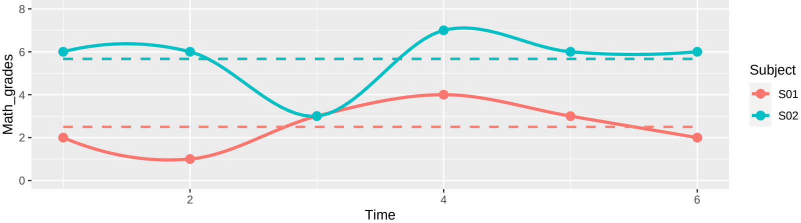
[1] The **additional variance terms** are the variance τ_{00}^2 of the random intercept λ_{0j} and the variance τ_{10}^2 of the random slope λ_{1j}. We will see this later. . .

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**.

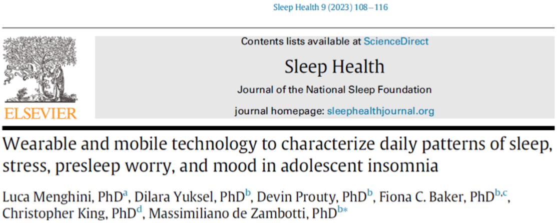
→ Observations from the same subject are not independent (*local dependencies*).

- 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

ELSEVIER

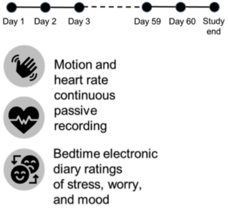
Sleep Health

Journal of the National Sleep Foundation

journal homepage: sleephealthjournal.org

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

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)

```
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.

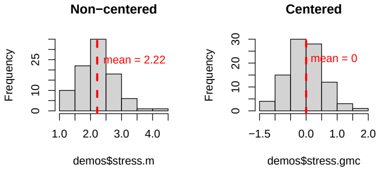
Data centering

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

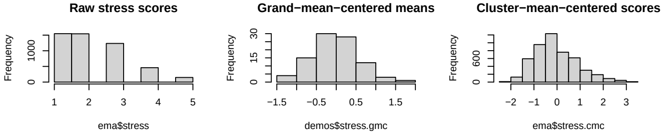
With LMER, we can distinguish two main ways to center the data:

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

```
# 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.

```
# 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

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
- LM = linear models
- CI = confidence intervals
- MLE = maximum likelihood estimator
- OLS = ordinary least squares
- NHST = null hypothesis significance testing
- SE = standard error
- SS = sum of squares

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

Achronyms & Greek letters

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

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