ADVANCED DATA ANALYSIS FOR PSYCHOLOGICAL SCIENCE

Part 2. Introduction to multivariate modeling

Luca Menghini Ph.D.

luca.menghini@unipd.it

***

Master degree in Developmental and Educational Psychology

University of Padova

2023-2024

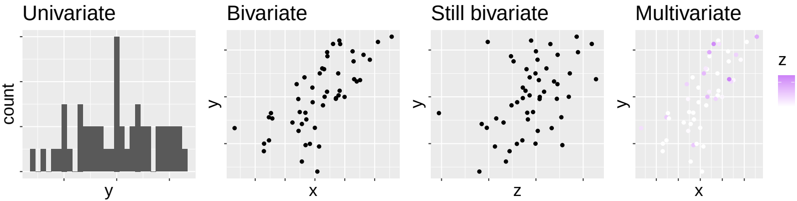
Outline of Part 2

- **`sem()` intro**: Gentle introduction to the world of structural equation modeling (SEM)
- **Path analysis**: Introduction to path analysis (aka SEM with observed variables) and focus on *mediation models*
- **Data structure**: How to approach a multivariate data structure, how to manipulate and pre-process multivariate data R
- **Model fit & evaluation**: How to fit a path analysis in R, to evaluate model fit, compare multiple models, and interpret model results R
- **`cfa()`**: How to conduct a confirmatory factor analysis (CFA) and to interpret its results R
- **Related topics**: In-depth topics related to multivariate modeling (e.g., cross-lagged panel models, multilevel and Bayesian SEM) 🔬

🔬 = not for the exam

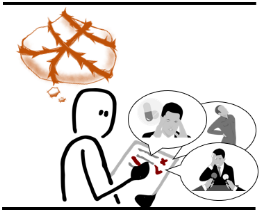
R = exercises with R (bring your laptop!)

Multivariate analyses for a multivariate reality



- In psychology, we mainly inspect empirical data focusing on **univariate** (y) or **bivariate** relationships (either y by x or y by z)
- But reality (particularly psychosocial reality) is complex, it is **multivariate** i.e., more than two variables covarying at the same time
- It is *reductionist* to separately analyze our variables without considering their overall interactions → **biased effect estimates**
- **Structural equation modeling (SEM)** allow to analyze the relationships of interest by accounting for the multivariate reality of psychosocial phenomena (e.g., y by x covarying with z; x affects y through z)

Observed indicators & latent variables

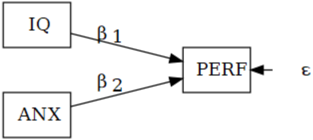


- In psychology, we are mainly interested in **latent variables** = phenomena that we cannot directly observe, but we can estimate from 1+ **observed indicators** (e.g., 10-item scale measuring anxiety)
- Are we allowed to do that? Yes (let's say yes), provided that we trust the indicator **construct validity** = their relationship with the latent variable they claim to measure
- **SEM** allow to evaluate that by ***quantifying* the latent variables** and their relationships with observed indicators

Structural what!?

Structural equation modeling (SEM)
= multivariate *linear* models formalized by **systems of equations**

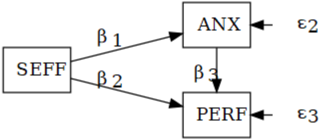
Linear models (LM): determining the link between a dependent and 1+ independent variables through a **single equation** like:
$PERF = \beta_1 IQ + \beta_2 ANX + \epsilon$



LM can only predict **one dependent variable at a time**, being either *univariate* (without predictors, i.e., intercept-only) or *bivariate* (with predictors).

SEM allow to simultaneously model multiple ~~dependent~~ *endogenous* variables with a **system of equations** like:

$$\begin{cases} ANX = \beta_1 SEFF + \epsilon_2 \\ PERF = \beta_2 SEFF + \beta_3 ANX + \epsilon_3 \end{cases}$$

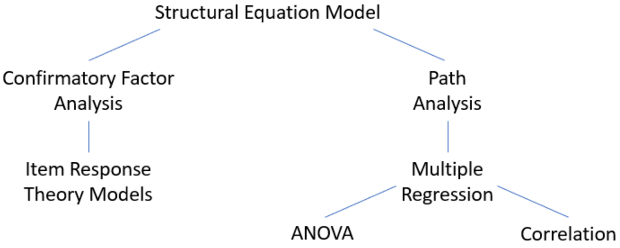


The SEM family

SEM = broad family of statistical models within which LM, ANOVA, and even correlation can be included.

Particularly, 2 main sub-families can be distinguished based on whether **latent variables** are included in the model or not:

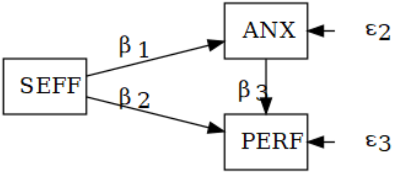
- **Path analysis**: multivariate linear models with observed variables only
- **Confirmatory factor analysis (CFA)**: multivariate linear models with both observed and latent variables


Structural Equation Model
Confirmatory Factor Analysis
Path Analysis
Item Response Theory Models
Multiple Regression
ANOVA
Correlation

Source: Beaujean (2014)

Path models & path analysis

Path models/diagrams = multivariate models with observed variables only
= pictorial representations (*diagrams*) of a theory of variable relationships

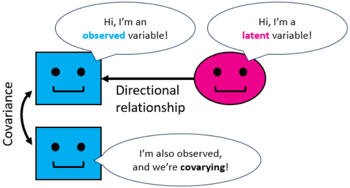


Paths = arrows (*edges*) linking the variables (*nodes*) in a model

Path analysis = analysis of multivariate relationships between observed variables (‘*quantification of the paths accounting for all other paths and errors*’)

Latent factors & CFA

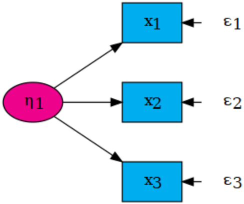
- **Observed/Manifest variable (OV)**
  variable that is directly observable (e.g., height, heart rate, item responses)
- **Latent variable/factor (LV)**
  variable that is *not* directly observable (e.g., anxiety, intelligence), but can be indexed by one or more observed variables
- In SEM, **OV**s are represented by squares/rectangles and indexed with lower case letters (e.g., x), whereas **LV**s are represented by circles/ellipses and indexed by the Greek letter η


Hi, I'm an observed variable!
Hi, I'm a latent variable!
Directional relationship
Covariance
I'm also observed, and we're covarying!

Confirmatory factor analysis (CFA)

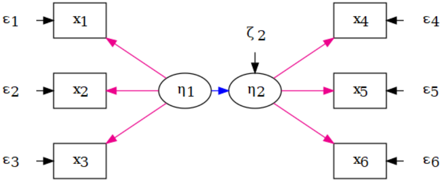
= analysis of the relationships (*factor loadings*) between a set of OVs and one or more LVs

CFA uses **latent variable models** to *form* or *quantify* LVs and their relationships with OVs (evaluation of **construct validity**)



SEM: Measurement & Structural model

To properly talk about 'full SEM' (or just SEM), we need both OVs and LVs



A SEM consists of two parts:

1. **Structural model**: Regression-like relationships among the variables, working similar to *path analysis*
2. **Measurement model** (or latent variable model): Relationships between OVs and LVs, working a little differently

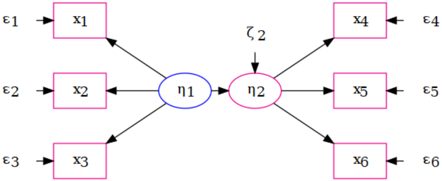
Notes:

In this sense, we may say that a CFA model is a 'full SEM' whereas a path model is not

A CFA is a SEM with just the measurement part (without the structural model)

A new classification: From in/dependent to exo/endogenous variables

In both SEM (e.g., CFA) and path models, the classic independent vs. dependent classification is replaced with a more meaningful one:



- **Exogenous variables**: variables (both OVs and LVs) without a direct ‘cause’ from inside the model (predictors), without error estimate
- **Endogenous variables**: variables (both OVs and LVs) directly ‘caused’ from inside the model (predictors & outcomes), with error estimate ϵ (OV) or ζ (LV)

A new starting point: From dataset columns to covariance matrices

The starting point of LM(ER) is a vector (or a set of vectors) of variable values, usually corresponding to one or more columns from a dataset.

```r
head(df,4)
```

```
  MAT QI WM STM
1  57 21 15  18
2  77 22 19  17
3  51 13 13  16
4  58 24  6  21
```

The starting point of SEM and path models is the **covariance matrix of the observed variables**.

$cov(x, y) = \sum(x_i - \overline{x})(y_i - \overline{y})/N$

```r
cov(df[,c("MAT","QI","WM","STM")])
```

```
       MAT    QI    WM  STM
MAT 100.70 24.89 17.21 7.99
QI   24.89 19.43  6.69 4.04
WM   17.21  6.69 17.33 2.23
STM   7.99  4.04  2.23 5.34
```

SEM estimate a number of parameters θ so that the **implied covariance matrix** $\hat{\sum}(\theta)$ (i.e., the covariance matrix predicted by the model based on the parameter estimates) is as close as possible to the **sample covariance matrix** S

Note: even the model parameters are estimated within **matrices of parameters**

Covariance & correlation

- **Variance** = Expected value of the **squared deviation from the mean** of a random variable, or degree to which it deviates from its expected value

  $var(x) = \sigma_x^2 = \frac{\sum (x_i - \overline{x})^2}{N}$

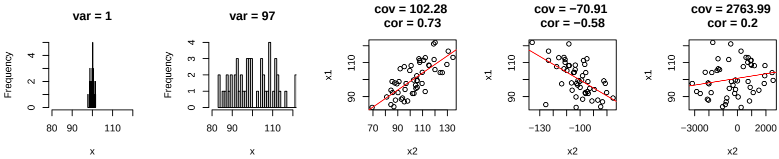
- **Covariance** = Measure of the **joint variability** of two random variables, or Degree to which they tend to deviate from their expected values in similar ways, either directly (positive cov) or inversely (negative cov), whose value depends on the variable scales of measurement (from $-\infty$ to $+\infty$)

  $cov(x_1, x_2) = \frac{\sum (x_{1i} - \overline{x_1})(x_{2i} - \overline{x_2})}{N}$

- **Correlation** = standardized covariance of two random variables

  Correlation ranges from -1 (perfectly negative) to +1 (perfectly positive)

  $cor(x_1, x_2) = \frac{cov(x_1, x_2)}{\sigma_{x_1}^2 \sigma_{x_2}^2}$



Important notes: $cov(x_1, x_1) = var(x_1)$; $cor(x_1, x_1) = 1$

Covariance matrix (S)

Given a set of p variables, we can define the covariance matrix:

$$S = \begin{bmatrix} s_{11} & ..., & s_{1j} & ... & s_{1p} \\ ... & ... & ... & ... & ... \\ s_{i1} & ... & s_{ij} & ... & s_{ip} \\ ... & ... & ... & ... & ... \\ s_{p1} & ... & s_{pj} & ... & s_{pp} \end{bmatrix}$$

```r
cov(df[,c("MAT","QI","WM","STM")])
```

```
       MAT    QI    WM  STM
MAT 100.70 24.89 17.21 7.99
QI   24.89 19.43  6.69 4.04
WM   17.21  6.69 17.33 2.23
STM   7.99  4.04  2.23 5.34
```

Properties of the covariance matrix:

1. **Symmetrical**: $s_{ij} = s_{ji}$
2. The **main diagonal** shows the **variances** (= covariance between each variable and itself)

SEM estimate a number of parameters θ so that the **implied covariance matrix** $\hat{\sum}(\theta)$ (i.e., the covariance matrix predicted by the model based on the parameter estimates) is as close as possible to the **sample covariance matrix** S

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 12-13** 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

Psychosocial reality is complex: it's **multivariate** (3+ variables interacting at the same time) and involves **latent variables** (not directly measurable)

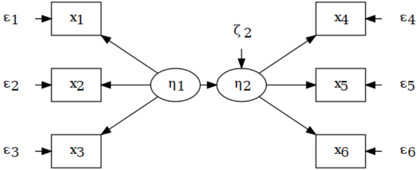
The solution

SEM allows to analyze the multivariate relationships among observed and latent variables through **systems of equations**:

$$\begin{cases} ANX = \beta_{21} SEFF + \epsilon_2 \\ PERF = \beta_{31} SEFF + \beta_{32} ANX + \epsilon_3 \end{cases}$$

SEM basics

- **Observed** (x) **vs latent variables** (η) depending on whether can be directly measured or not
- **Exogenous vs endogenous variables** depending on whether directly caused inside the model or not
- **Structural vs measurement model** depending on whether focusing on structural relationships or construct validity of the observed indicators
- **Path model**: SEM with observed variables only
- **CFA** = SEM with measurement model only
- Starting point of any SEM = **covariance matrix**



Path models: SEM with observed variables

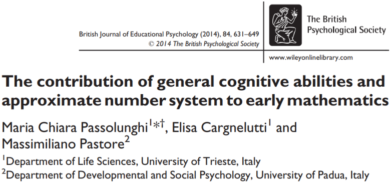
A path model is a pictorial representation (*diagram*) of a theory of variable relationships. Path analysis is widely used to model complex multivariate relationships (e.g., *mediation models*).

- Path analysis tests models of ***causal* relationships*** among observed variables
- All variables in path analysis are **observed**
- Path analysis uses **systems of regression equations**

*Note: Within path analysis (and SEM) we assume that the relationships are *causal*, but this is not necessarily true (e.g., observational studies) → causation requires experimental manipulation, control group, etc.

Case study: Early mathematical abilities

A sample of 120 first-grade children (58 females; mean age: 6 years, 3 months) was assessed over the following variables:



British Journal of Educational Psychology (2014), 84, 631–649
© 2014 The British Psychological Society

The British Psychological Society

www.wileyonlinelibrary.com

The contribution of general cognitive abilities and approximate number system to early mathematics

Maria Chiara Passolunghi[1*†], Elisa Cargnelutti[1] and Massimiliano Pastore[2]

[1]Department of Life Sciences, University of Trieste, Italy
[2]Department of Developmental and Social Psychology, University of Padua, Italy

- `MAT`: early mathematical abilities (e.g., comparison, classification) measured with the Early Numeracy Test
- `QI`: intelligence level measured with the Wechsler Intelligence Scale for Children (WISC-III)
- `WM`: working memory capacity measured with the Backward word recall task
- `STM`: short-term memory capacity measured with the Forward word recall task
- `ANS`: approximate number system = innate system for approximate quantity manipulation (e.g., approximate computations, comparing 2+ sets of elements without counting), measured with several tasks

RQ: **How much can MAT abilities be attributed to memory & ANS?**

Source: Pastore (2021). Analisi dei dati in ambito di comunità

Data exploration

First, let's explore the data:

```
library(devtools); install_github("https://github.com/masspastore/ADati") # install ADati pkg
```

```
data( earlymath, package = "ADati" ) # loading earlymath dataset from ADati pkg

head(earlymath,3) # showing first 3 rows
```

```
    gender MAT QI WM STM ANS
147      m  57 21 15  18  80
144      m  77 22 19  17  76
155      f  51 13 13  16  79
```

```
summary(earlymath[,c(2,4:ncol(earlymath))]) # summarizing variables (not showing QI due to space limits)
```

```
      MAT              WM              STM              ANS       
 Min.   :36.00   Min.   : 1.00   Min.   :13.00   Min.   :45.00  
 1st Qu.:61.75   1st Qu.:12.00   1st Qu.:17.00   1st Qu.:74.00  
 Median :68.00   Median :14.50   Median :18.00   Median :80.00  
 Mean   :68.56   Mean   :14.55   Mean   :18.43   Mean   :79.34  
 3rd Qu.:75.00   3rd Qu.:17.00   3rd Qu.:20.00   3rd Qu.:85.00  
 Max.   :91.00   Max.   :28.00   Max.   :26.00   Max.   :94.00  
```

```
round( cor(earlymath[,2:ncol(earlymath)]), 2) # correlations
```

```
     MAT   QI   WM  STM  ANS
MAT 1.00 0.56 0.41 0.34 0.26
QI  0.56 1.00 0.36 0.40 0.23
WM  0.41 0.36 1.00 0.23 0.12
STM 0.34 0.40 0.23 1.00 0.19
ANS 0.26 0.23 0.12 0.19 1.00
```

Linear model as a path diagram

Let's fit a multiple linear model: $MAT = \beta_0 + \beta_1 WM + \beta_2 STM + \beta_3 ASN + \epsilon$

```r
lm.fit <- lm(MAT ~ WM + STM + ANS, data = earlymath) # fitting LM
```

```r
summary(lm.fit)$coefficients # LM regression table
```

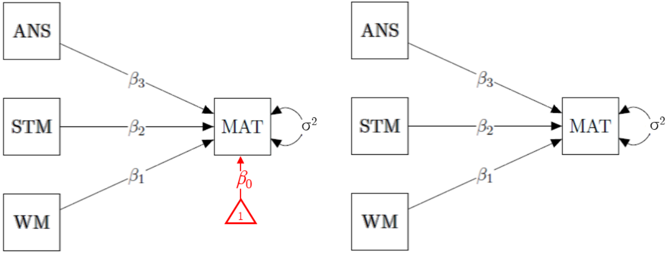
```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    20.03       9.61    2.09     0.04
WM              0.81       0.20    4.10     0.00
STM             1.01       0.36    2.81     0.01
ANS             0.23       0.11    2.16     0.03
```

Residual variance σ^2:

```r
summary(lm.fit)$sigma^2
```

```
[1] 75.94542
```

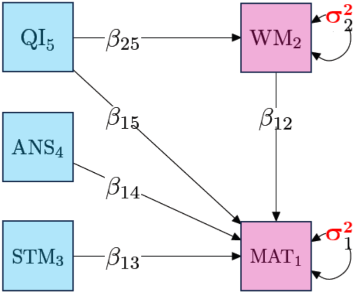
This model can be graphically represented as a path diagram and further simplified by removing the intercept β_0 (note: **triangles represent constants**)



How many parameters? **Five**: Intercept, 3 slopes, residual variance

Multivariate path models

In the previous example, we only considered **bivariate relationships** (i.e., 2 variables at a time, controlling for other variables). But what if we include `IQ` as a common predictor of both `WM` and `MAT`? We would have 3 variables interacting at the same time.



Both `MAT` and `WM` are **endogenous variables** because they receive 1+ arrow(s) and have error variance σ^2.
In contrast, `STM`, `ANS`, and `QI` are **exogenous variables** because they do not receive any arrow and have no errors.

A single LM equation is insufficient to describe this model. We need 2 separated equations: one for each variable that depends upon another variable

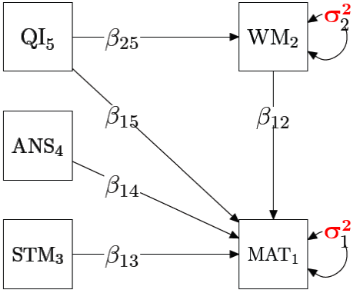
Path analysis (and SEM) uses **one equation per endogenous variable**:

$$\begin{cases} MAT_1 = \beta_{12}WM_2 + \beta_{13}STM_3 + \beta_{14}ANS_4 + \beta_{15}QI_5 + \epsilon_1 \\ WM_2 = \beta_{25}QI_5 + \epsilon_2 \end{cases}$$

Graphical notation (1/3): Error terms

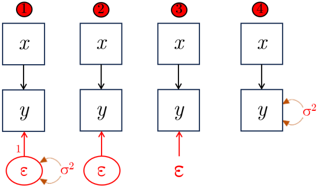
$$\begin{cases} MAT_1 = \beta_{12} WM_2 + \beta_{13} STM_3 + \beta_{14} ANS_4 + \beta_{15} QI_5 + \epsilon_1 \\ WM_2 = \beta_{25} QI_5 + \epsilon_2 \end{cases}$$

Errors = *residuals* or *disturbances* = discrepancy between observed and predicted values (as in LM!), they represent something *unexplained* = **exogenous** and *not directly observable* = **latent**



σ^2 = variance of a variable error (residual var.)

Alternative ways to represent errors: some highlight their latent nature (#1 and #2), some highlight their variance (#1 and #4), and some highlight both (#1). You need to know them

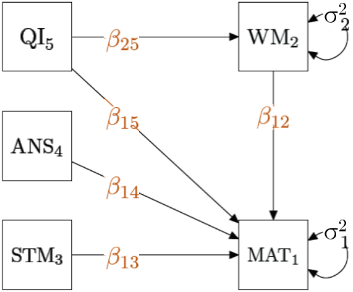


In this course, we will mainly use notation #4.

Graphical notation (2/3): Arrows & coefficients

$$\begin{cases} MAT_1 = \beta_{12} WM_2 + \beta_{13} STM_3 + \beta_{14} ANS_4 + \beta_{15} QI_5 + \epsilon_1 \\ WM_2 = \beta_{25} QI_5 + \epsilon_2 \end{cases}$$

Arrows = *relationships* between 2 variables (*paths* or *slopes*) or between a variable and itself (*residual variance*), such that we do not include an arrow when a relationship is not expected (e.g., between `QI` and `ASN`) → path models are *complete*



How to index variables and paths:

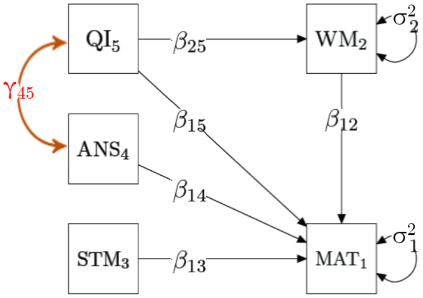
- **Variables** are indexed from the one receiving most arrows (MAT_1) to the last exogenous variable (QI_5)
- **Path coefficients** β are indexed by firstly reporting the index of the endogenous variable and then that of the exogenous variable

From plot to equations: endogenous v. ~ sum of all linked exogenous v. + error

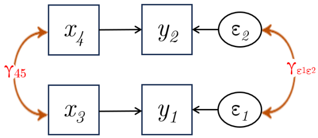
Graphical notation (3/3): Covariances

$$\begin{cases} MAT_1 = \beta_{12} WM_2 + \beta_{13} STM_3 + \beta_{14} ANS_4 + \beta_{15} QI_5 + \epsilon_1 \\ WM_2 = \beta_{25} QI_5 + \epsilon_2 \\ Cov(ANS_4, QI_5) = \gamma_{ANS_4, QI_5} \end{cases}$$

Covariances = *non-directional (symmetric)* relationships between 2 ***exogenous*** v.



- Covariances are usually *not* reported in the system of equations, but they *can* be graphically represented with (rounded) **double-headed arrows**
- **Endogenous variables cannot covary but their errors ϵ can**



Clarification on covariance terms in SEM

Covariances γ are intrinsic relationships between observed variables (we saw that SEM are fitted on the covariance matrix of observed variables).

In slide #22 , we saw that path models are assumed to be *complete* models (i.e., we don't include an arrow when a relationship is not expected).
However, this rule only applies to single-headed arrows (path coefficients β), whereas it **does not applies to the covariances** γ.
Covariances γ are always there, whether you estimate them or not. In contrast, if we don't specify a path coefficient β between two variables, the two variables can only covariate but they are not in a symmetric relationship.

→ the explicit inclusion of covariances γ does not affect the estimation of the path coefficients β, it only means that the models estimate the covariance parameter and its standard errror, but we will see this later...

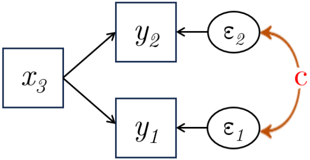
For the exam, you only need to know that double-headed arrows between two variables represent covariances, and that endogenous variables cannot covary but their errors can

Regression, partial correlation, and path coefficients

Path coefficients (single-headed arrows) are **partial regression coefficients** (*slopes*): as in LM, they index the *effect* of x on y by controlling for (i.e., after removing the effect of) other predictors, which are fixed to zero

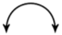
Covariances between two exogenous variables (double-headed arrows), or between the errors of two endogenous variables, are **partial correlation coefficients**: they express the relationship between two variables by controlling for (i.e., after removing the effect of) all other correlated variables, which are fixed to zero

For instance, the figure below (source: Beaujeau, 2014) shows a path model of a partial correlation. Variables y_1 and y_2 are not allowed to covary since they are endogenous, but their errors are allowed to do so. Thus, the c coefficient is the relationship between y_1 and y_2 after removing the effect of x_1 from both variables.

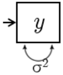


Graphical notation: Recap

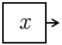
Directional (asymmetric) relationship



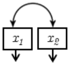
Non-directional (symmetric) relationship
(covariance/correlation)



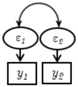
Endogenous observed variable
with associated **variance σ^2 of errors ε**



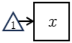
Exogenous observed variable
without associated error



Covarying exogenous variables



Endogenous variable
with **covarying errors**



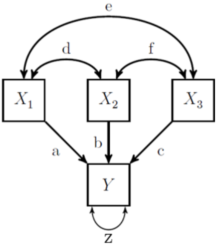
Constant (intercept)

Tracing rules & path coefficients

Sewall Wright (1889–1988): US geneticist that firstly developed rules for how to estimate values for a path model's coefficients by tracing the paths within it (i.e., path analysis).

Tracing rules = rules to estimate the covariance between 2 variables by *summing* the appropriate connecting paths:

1. Trace all paths between 2 variables multiplying all coefficients
2. Start by going backwards along single-headed arrows, no loops
3. Once you start going forward, you cannot no longer go back
4. Each path can only include one double-headed arrow



Starting from *observed covariances* (or correlations), we can compute the value of path coefficients. For instance, to compute path a starting from the observed correlations between X_1 and Y (e.g., $r = .70$), between X_1 and X_2 (e.g., $d = .24$), and between X_1 and X_3 (e.g., $e = .20$):

$$r_{X_1,Y} = \text{a + db + ec} \rightarrow .70 = \text{a} + .24\text{c} + .20\text{b} \rightarrow \text{a} = .70 \text{ - } .24\text{c} \text{ - } .20\text{b}$$

Standardized vs. *Un*standardized coefficients

Path coefficients are **partial regression coefficients** (relationship between an exogenous x and an endogenous variable y, controlling for all other exogenous variable affecting y). Similar to LM, they can be either *un*standardized or standardized:

- **Unstandardized coefficients** are obtained when the model is fitted on the variables expressed in their *natural metrics* (raw score units of measurement) → useful when raw score units are meaningful (e.g., age, meters, bpm) and when comparing the same variable relationship across samples
- **Standardized coefficients (ranging from -1 to 1)** are obtained when the model is fitted on standardized variables (i.e., variables transformed into ***z-scores***: $z_{x_i} = (x_i - \overline{x})/s_x$) → useful to compare coefficients within the same model and/or the same sample

To standardize an unstandardized coefficient: $b^* = b \times (s_Y/s_X)$

To unstandardize a standardized coefficient: $b = b^* \times (s_X/s_Y)$

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 14-15** 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. . .

The problem & the solution

Reality is **multivariate** and involves **latent variables**; SEM allows to analyze them through **systems of equations**:

$$\begin{cases} y_2 = \beta_{32} x_3 + \epsilon_2 \\ y_1 = \beta_{21} y_2 + \beta_{31} x_3 + \epsilon_1 \end{cases}$$

SEM basics

- Observed (x) vs latent (η)
- Exogenous vs endogenous
- Structural vs measurement model
- Path model: obs. variables only
- CFA = measurement model only
- Starting point of any SEM

= **covariance matrix**

Path analysis

Pictorial representation of a theory of (observed) variable relationship

Graphical notation

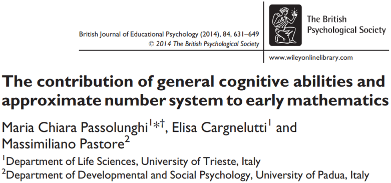
- Variables: end.→with error; ex.→without error
- Errors: always exogenous and latent
- Path coefficients: single-headed arrows, *complete*
- (Co)variances: double-headed arrows - they are always there, whether you estimate them or not

Path coefficients

= partial regression coefficients (slopes) either **unstandardized** (computed from raw variables, depending on the variable scale of measurement) or **standardized** (computed from standardized variables, ranging from -1 to 1)

Case study: Early mathematical abilities 🖩

A sample of 120 first-grade children (58 females; mean age: 6 years, 3 months) was assessed over the following variables:



British Journal of Educational Psychology (2014), 84, 631–649
© 2014 The British Psychological Society

The British Psychological Society

www.wileyonlinelibrary.com

The contribution of general cognitive abilities and approximate number system to early mathematics

Maria Chiara Passolunghi[1*†], Elisa Cargnelutti[1] and Massimiliano Pastore[2]

[1]Department of Life Sciences, University of Trieste, Italy
[2]Department of Developmental and Social Psychology, University of Padua, Italy

- `MAT`: early mathematical abilities (e.g., comparison, classification) measured with the Early Numeracy Test
- `QI`: intelligence level measured with the Wechsler Intelligence Scale for Children (WISC-III)
- `WM`: working memory capacity measured with the Backward word recall task
- `STM`: short-term memory capacity measured with the Forward word recall task
- `ANS`: approximate number system = innate system for approximate quantity manipulation (e.g., approximate computations, comparing 2+ sets of elements without counting), measured with several tasks

RQ: **How much can MAT abilities be attributed to memory & ANS?**

Source: Pastore (2021). Analisi dei dati in ambito di comunità

Data structure in multivariate analyses

In SEM (including path analysis and CFA), data analyses are usually based on wide-form datasets with one row per participant:

```
head(earlymath) # showing first 6 rows
```

```
    gender MAT QI WM STM ANS
147      m  57 21 15  18  80
144      m  77 22 19  17  76
155      f  51 13 13  16  79
55       f  58 24  6  21  86
6        f  64 28 15  19  75
13       m  68 27 14  19  86
```

Provided that we have **no missing data** (but there are ways to deal with that), such wide-form dataset is used by the model to automatically compute the **covariance matrix of observed variables**, which is the starting points to fit the models.

```
cov(earlymath[,2:ncol(earlymath)])
```

```
           MAT        QI        WM      STM       ANS
MAT 100.702451 24.889286 17.211345 7.991317 20.261415
QI   24.889286 19.427941  6.692017 4.039496  7.844328
WM   17.211345  6.692017 17.325210 2.230252  3.902941
STM   7.991317  4.039496  2.230252 5.340056  3.413725
ANS  20.261415  7.844328  3.902941 3.413725 59.924300
```

Note: since the covariance matrix is the starting point, many software (including R) can fit SEM directly on the covariance matrix

Fitting a (bivariate) path model with R

We will use the **lavaan** (*la*tent *va*riable *an*alysis) package (Rosseel, 2012), which uses the **sem()** function to fit SEM with observed (path analysis) and/or latent variables.

```
library(lavaan)
```

Let's start with a bivariate model (with only one endogenous variable) to highlight the differences between path analysis and LM in the model specification:

Linear model (LM)

```
# fitting model
fit.lm <- lm(MAT ~ WM + STM + ANS,
             data = earlymath)
# parameter estimates
summary(fit.lm)$coefficients
```

```
      Estimate Std. Error t value Pr(>|t|)
(Int)    20.03       9.61    2.09     0.04
WM        0.81       0.20    4.10     0.00
STM       1.01       0.36    2.81     0.01
ANS       0.23       0.11    2.16     0.03
```

```
# residual variance sigma2
summary(fit.lm)$sigma^2
```

```
[1] 75.95
```

Path model (PM)

```
# specifying model
mymodel <- 'MAT ~ WM + STM + ANS'
# fitting model to the data
fit.sem <- sem(model = mymodel, data = earlymath)
parameterestimates(fit.sem) # par. estimates
```

```
   lhs op rhs   est   se    z pvalue ci.lower ci.upper
1  MAT  ~  WM  0.81 0.19 4.17   0.00     0.43     1.19
2  MAT  ~ STM  1.01 0.35 2.85   0.00     0.32     1.71
3  MAT  ~ ANS  0.23 0.10 2.19   0.03     0.02     0.43
4  MAT ~~ MAT 73.41 9.48 7.75   0.00    54.84    91.99
5   WM ~~  WM 17.18 0.00   NA     NA    17.18    17.18
6   WM ~~ STM  2.21 0.00   NA     NA     2.21     2.21
7   WM ~~ ANS  3.87 0.00   NA     NA     3.87     3.87
8  STM ~~ STM  5.30 0.00   NA     NA     5.30     5.30
9  STM ~~ ANS  3.39 0.00   NA     NA     3.39     3.39
10 ANS ~~ ANS 59.42 0.00   NA     NA    59.42    59.42
```

Path model summary

```
summary(fit.sem)
```

```
lavaan 0.6.16 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         4

  Number of observations                           120

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  MAT ~
    WM                0.812    0.195    4.172    0.000
    STM               1.012    0.354    2.855    0.004
    ANS               0.228    0.104    2.195    0.028

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .MAT              73.414    9.478    7.746    0.000
```

- **First lines**: info on convergence, parameter estimation method (ML), optimization (. . .), and number of estimated parameters (3 path coeff. + 1 residual variance)
- **Model test User Model**: info on model fit (we will see this later)
- **Parameter Estimates**: other info on parameter estimation method
- **Regressions**: path coefficients estimated by the **structural model**, with their **standard error**, **z-value**, and **p-value**
- **Variances**: estimated **residual variance** of any **endogenous** variable

Path coefficient interpretation

```
summary(fit.sem)
```

```
lavaan 0.6.16 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         4

  Number of observations                           120

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  MAT ~
    WM                0.812    0.195    4.172    0.000
    STM               1.012    0.354    2.855    0.004
    ANS               0.228    0.104    2.195    0.028

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .MAT              73.414    9.478    7.746    0.000
```

- `Estimate` = estimated parameter/coefficient (predicted difference or change for a 1-unit increase in the exogenous variable)
- `Std.Error` = standard error (uncertainty) of the estimate
- `z-value` = test statistic computed as $z = Estimate/Std.Error$
- `P(>|z|)` = p corresponding to the t-value with *No. Obs. − No. Coeff. − 1* degrees of freedom
- Here, for instance:

  - a 1-unit increase in `WM` predicts an increase in `MAT` by 0.812 units

  - the residual variance of `MAT` is 60.787

  - p-values suggest that all path coefficients are significant

Path model vs. linear model estimates

We can see that the coefficients estimated by the path model are very similar to those estimated with LM:

```
coef(fit.lm); summary(fit.lm)$sigma^2 # LM estimate
```

```
(Intercept)          WM         STM         ANS      sigma2
     20.035       0.812       1.012       0.228      75.945
```

```
coef(fit.sem) # path model estimates
```

```
  MAT~WM  MAT~STM  MAT~ANS MAT~~MAT
   0.812    1.012    0.228   73.414
```

They are the same, but where is the intercept?

In SEM, intercepts are usually not considered as ‘direct’ model parameters. To estimate them, we need to set `meanstructure = TRUE`

```
fit.sem <- sem(model = mymodel, data = earlymath, meanstructure= TRUE)
coef(fit.sem) # Here's the intercept!
```

```
  MAT~WM  MAT~STM  MAT~ANS MAT~~MAT    MAT~1
   0.812    1.012    0.228   73.414   20.035
```

Note: Path analysis coefficients can be interpreted identically to LM coefficients

Hands on R (part 1)

1. Open the `earlymath` dataset from the `ADati package`

```
# how to install the ADati package:
library(devtools) # install and open the devtools package
install_github("https://github.com/masspastore/ADati") # install the ADati pkg
data(earlymath, package = "ADati") # load earlymath dataset from ADati pkg
```

2. Fit a linear model `lm1` predicting `MAT` by `WM`, `STM`, `ANS`, and `QI`
3. Fit a path model `pm1` with the same 'outcome' and 'predictor' variables
4. Print, interpret, and compare the parameters estimated by both models
5. Inspect the predicted covariance matrix by running `inspect(pm1,"estimates")$psi[2:5,2:5]` and compare it with the observed covariance matrix of exogenous variables `cov(earlymath[,c("WM","STM","ANS","QI")])`
6. Standardize all variables ($z_{x_i} = (x_i - \overline{x})/s_x$), re-fit the same model (call it `pm1.z`), and print the estimated parameters
7. Standardize the parameters estimated by the original model `pm1` by running `standardizedsolution(pm1)` and compare the output with that of `parameterestimates(pm1.z)`

*Un*standardized vs. standardized solution

In SEM (including path analysis and CFA), we refer to the ***un*standardized solution** when the parameters are unstandardized, i.e., they are estimated from unstandardized variables and their size depends on the measurement scale of each variable

```r
# unstandardized solution
parameterestimates(pm1)[1:5,]
```

```
  lhs op rhs    est    se     z pvalue ci.lower ci.upper
1 MAT  ~  WM  0.537 0.185 2.895  0.004    0.173    0.900
2 MAT  ~ STM  0.465 0.341 1.364  0.172   -0.203    1.132
3 MAT  ~ ANS  0.154 0.096 1.612  0.107   -0.033    0.341
4 MAT  ~  QI  0.937 0.188 4.993  0.000    0.569    1.305
5 MAT ~~ MAT 60.787 7.848 7.746  0.000   45.406   76.168
```

In contrast, we refer to the **standardized solution** when the parameters are standardized, i.e., they range from -1 to +1 because they have been either estimated from standardized variables or transformed into standardized coefficient after estimation

```r
# standardized solution
standardizedsolution(pm1)[1:5,]
```

```
  lhs op rhs est.std    se     z pvalue ci.lower ci.upper
1 MAT  ~  WM   0.223 0.075 2.978  0.003    0.076    0.369
2 MAT  ~ STM   0.107 0.078 1.373  0.170   -0.046    0.260
3 MAT  ~ ANS   0.119 0.073 1.626  0.104   -0.024    0.262
4 MAT  ~  QI   0.412 0.075 5.463  0.000    0.264    0.559
5 MAT ~~ MAT   0.609 0.062 9.763  0.000    0.487    0.731
```

Parameter matrices

As anticipated in slide #11, SEM works with matrices: it starts from a matrix (i.e., the observed covariance matrix) and it returns **matrices of estimated parameters**.

Whereas we saw that parameters can be printed into regression-like tables, something more complex is happening under the hood: the model returns matrices of estimates:

λ = matrix of **factor loadings**

```
inspect( pm1, "estimates")[1]
```

```
$lambda
    MAT WM STM ANS QI
MAT   1  0   0   0  0
WM    0  1   0   0  0
STM   0  0   1   0  0
ANS   0  0   0   1  0
QI    0  0   0   0  1
```

θ = matrix of **latent factor (co)variances**

```
inspect( pm1, "estimates")[2]
```

```
$theta
    MAT WM STM ANS QI
MAT   0
WM    0  0
STM   0  0   0
ANS   0  0   0   0
QI    0  0   0   0  0
```

Note: λ and θ require latent variables

ψ = matrix of observed variable **(co)variances** (i.e., the (co)variances estimated by the model)

```
inspect( pm1, "estimates")[3]
```

```
$psi
       MAT     WM    STM    ANS     QI
MAT 60.787
WM   0.000 17.181
STM  0.000  2.212  5.296
ANS  0.000  3.870  3.385 59.425
QI   0.000  6.636  4.006  7.779 19.266
```

β = matrix of **regression coefficients** (paths)

```
inspect( pm1, "estimates")[4]
```

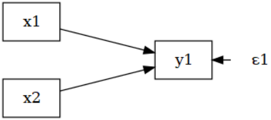
```
$beta
    MAT    WM   STM   ANS    QI
MAT   0 0.537 0.465 0.154 0.937
WM    0 0.000 0.000 0.000 0.000
STM   0 0.000 0.000 0.000 0.000
ANS   0 0.000 0.000 0.000 0.000
QI    0 0.000 0.000 0.000 0.000
```

Fitting a (multivariate) path model with R

As we saw in slide #20 , path analysis uses **one equation per endogenous variable**, and so does the model syntax used by `lavaan`:

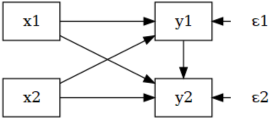
- 1 endogenous = 1 equation

```
model <- 'y1 ~ x1 + x2'
```



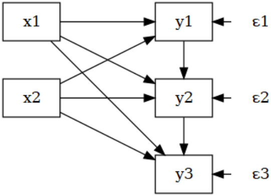
- 2 endogenous = 2 equations

```
model <- 'y1 ~ x1 + x2
          y2 ~ x1 + x2 + y1'
```



- 3 endogenous = 3 equations

```
model <- 'y1 ~ x1 + x2
          y2 ~ x1 + x2 + y1
          y3 ~ x1 + x2 + y1 + y2'
```



Note: similar to `lm()` and `lmer()`, we do not specify the error term in the formula, but just the exogenous variables (x) related to each endogenous variable (y)

Hands on R (part 2)

1. Using the `sem()` function from the `lavaan` package, fit a model corresponding to the path diagram represented in slide #20 ; you can use the `semPaths(model_name)` function from the `semPlot` package to check whether you did it right
2. How many unknown parameters? Try answering before running the code
3. Print, interpret, and evaluate the statistical significance of the parameters estimated by the **unstandardized solution** and those estimated by the **standardized solution**
4. In the model formula, label the path[1] β_{25} as “a”, the β_{12} as “b”, and the path β_{15} as “c”, then add a new line of equation: `ab := a*b` [2], fit the model again, and print the parameters

[1] Note: to **label** a path with a letter (or a word), just write the letter before the corresponding predictor and put a * between them, for example: `MAT ~ a*QI`

[2] The symbol `:=` stands for “Define non-model parameter” (i.e., creating a parameter by combining other parameters)

Labeled and composed parameters in lavaan

In `lavaan`, it is possible to **label parameters** (i.e., to give a name to a parameter, similar to how we do when we create an R object with the `<-` symbol) by ‘multiplying’ the parameter label with the name of the variable corresponding to that parameter.

For instance, here we call the path between QI and WM “a”, whereas we call “b” and “c” the paths linking `MAT` to `WM` and `QI`, respectively. The path linking `STM` to `MAT` is called “tony” :)

```r
model <- 'MAT ~ b*WM + c*QI + ANS + tony*STM
          WM ~ a*QI'
```

Why should we label parameters? Because this allows **creating new parameters as a combination** of other parameters. And this is needed in many analyses, including **mediation**.

```r
model <- 'MAT ~ b*WM + c*QI + ANS + tony*STM
          WM ~ a*QI
          # the new parameter "ab" is the product between "a" and "b"
          ab := a*b
          # the new parameter "total" is the sum of "c" and "ab"
          total := c + ab'
```

Mediation analysis

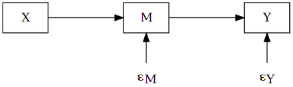
A **mediation model** is a multivariate model that attempts to identify and explain the relationship between a **predictor** (X) and an **outcome** variable (Y) when we hypothesize that a third variable (**mediator** M) can influence the direct relationship between X and Y.

Note: mediation is different from moderation

Mediation → indirect effect

A **mediator** is expected to be influenced by the predictor and to influence the outcome → **indirect effect** of the predictor *through* the mediator.

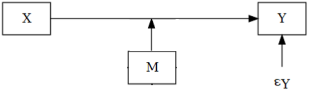
```
model <- 'Y ~ M
          M ~ X'
```



Moderation → interaction

A **moderator** is expected to modulate the relationship between predictor and outcome (e.g., stronger/weaker relationship for higher levels of the moderators), without being necessarily related to X and Y.

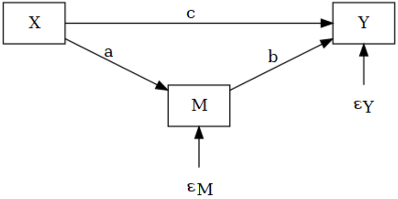
```
model <- 'Y ~ M * X'
```



Types of effects in a mediation model

A mediation model involves three types of effects:

- **Direct effects**: direct influence of the predictor X on the outcome Y (path c), as indexed by regression/path coefficients β
- **Indirect effects**: indirect influence of X on Y through a third variable M that mediates the two of them, computed as the **product of the direct effects** of X on M and Y ($a \times b$)
- **Total effects**: sum of direct and indirect effects of X on Y ($a \times b + c$)



Note: when the indirect effect is equal to zero (and thus, total effect = indirect effect), we talk about **full mediation**, otherwise we talk about **partial mediation**

Mediation analysis in lavaan

First, we specify the model as we are used:

```
model <- 'Y ~ X + M
          M ~ X'
```

Second, to distinguish *direct* and *mediation* effects, we can rewrite the same model by splitting the first equation in two different equation (i.e., equivalent to the first one):

```
model <- '# direct effect
          Y ~ X
          # mediation effects
          Y ~ M
          M ~ X'
```

Note: the symbol := stands for "*non-model parameter defined as*"

Third, we label the effects:

```
model <- '# direct effect
          Y ~ c*X
          # mediation effects
          Y ~ b*M
          M ~ a*X'
```

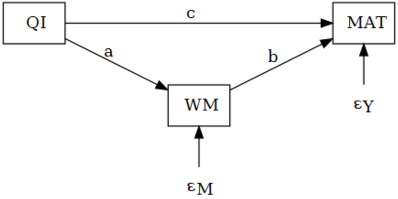
Finally, we define new parameters as the combination of the other parameters:

```
model <- '# direct effect
          Y ~ c*X
          # mediation effects
          Y ~ b*M
          M ~ a*X
          # indirect effect
          ab := a*b
          # total effect
          tot := c + (a*b)'
```

Mediation model fit

Here’s our mediation model of `QI` (predictor), `WM` (mediator), and `MAT` (outcome):

```
mymodel <- '# direct effect
            MAT ~ c*QI
            # mediation effects
            MAT ~ b*WM
            WM ~ a*QI
            # indirect effect
            ab := a*b
            # total effect
            tot := c + (a*b)'
fit <- sem(model = mymodel, data = earlymath)
```



Mediation model output

Here are the parameters estimated by our mediation model:

```r
parameterestimates(fit)[,1:8]
```

```
  lhs op     rhs label    est    se     z pvalue
1 MAT  ~      QI     c  1.083 0.178 6.093  0.000
2 MAT  ~      WM     b  0.575 0.188 3.056  0.002
3  WM  ~      QI     a  0.344 0.080 4.291  0.000
4 MAT ~~     MAT       63.317 8.174 7.746  0.000
5  WM ~~      WM       14.895 1.923 7.746  0.000
6  QI ~~      QI       19.266 0.000    NA     NA
7  ab :=     a*b    ab  0.198 0.080 2.489  0.013
8 tot := c+(a*b)   tot  1.281 0.172 7.457  0.000
```

- How to interpret them? Is this a partial or a full mediation?

Mediation model output

Here are the parameters estimated by our mediation model:

```r
parameterestimates(fit)[,1:8]
```

```
  lhs op     rhs label    est    se     z pvalue
1 MAT  ~      QI     c  1.083 0.178 6.093  0.000
2 MAT  ~      WM     b  0.575 0.188 3.056  0.002
3  WM  ~      QI     a  0.344 0.080 4.291  0.000
4 MAT ~~     MAT       63.317 8.174 7.746  0.000
5  WM ~~      WM       14.895 1.923 7.746  0.000
6  QI ~~      QI       19.266 0.000    NA     NA
7  ab :=     a*b    ab  0.198 0.080 2.489  0.013
8 tot := c+(a*b)   tot  1.281 0.172 7.457  0.000
```

- How to interpret them? Is this a partial or a full mediation?
- **Direct effect** $=$ c $= 1.083$ (SE $= 0.178$), $z = 6.093$, $p = 0$ → in this case it is positive and significant (i.e., it is a **partial mediation**)

Mediation model output

Here are the parameters estimated by our mediation model:

```r
parameterestimates(fit)[,1:8]
```

```
  lhs op      rhs label    est    se     z pvalue
1 MAT  ~       QI     c  1.083 0.178 6.093  0.000
2 MAT  ~       WM     b  0.575 0.188 3.056  0.002
3  WM  ~       QI     a  0.344 0.080 4.291  0.000
4 MAT ~~      MAT       63.317 8.174 7.746  0.000
5  WM ~~       WM       14.895 1.923 7.746  0.000
6  QI ~~       QI       19.266 0.000    NA     NA
7  ab :=      a*b    ab  0.198 0.080 2.489  0.013
8 tot := c+(a*b)   tot  1.281 0.172 7.457  0.000
```

- How to interpret them? Is this a partial or a full mediation?
- **Direct effect** $= c = 1.083$ (SE $= 0.178$), $z = 6.093$, $p = 0 \rightarrow$ in this case it is positive and significant (i.e., it is a **partial mediation**)
- **Indirect effect** $= a \times b = 0.198$ (SE $= 0.08$), $z = 2.489$, $p = 0.013 \rightarrow$ in this case it is positive and significant

Mediation model output

Here are the parameters estimated by our mediation model:

```r
parameterestimates(fit)[,1:8]
```

```
  lhs op     rhs label    est    se     z pvalue
1 MAT  ~      QI     c  1.083 0.178 6.093  0.000
2 MAT  ~      WM     b  0.575 0.188 3.056  0.002
3  WM  ~      QI     a  0.344 0.080 4.291  0.000
4 MAT ~~     MAT       63.317 8.174 7.746  0.000
5  WM ~~      WM       14.895 1.923 7.746  0.000
6  QI ~~      QI       19.266 0.000    NA     NA
7  ab :=     a*b    ab  0.198 0.080 2.489  0.013
8 tot := c+(a*b)   tot  1.281 0.172 7.457  0.000
```

- How to interpret them? Is this a partial or a full mediation?
- **Direct effect** $= c = 1.083$ $(\mathrm{SE} = 0.178)$, $z = 6.093$, $p = 0 \rightarrow$ in this case it is positive and significant (i.e., it is a **partial mediation**)
- **Indirect effect** $= a \times b = 0.198$ $(\mathrm{SE} = 0.08)$, $z = 2.489$, $p = 0.013 \rightarrow$ in this case it is positive and significant
- **Total effect** $= c + (a \times b) = 1.281$ $(\mathrm{SE} = 0.172)$, $z = 7.457$, $p = 0 \rightarrow$ in this case it is positive and significant

Hands on R (part 3)

1. Modify the model specified in the last point of Part 2 by adding the indirect and total effect
2. Print and interpret the estimated parameters
3. Visualize the model by using the `semPaths(model_name)` function from the `semPlot` package

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 16-17** 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 psicologia (e applicazioni in R). Il Mulino.
- Pastore, M. (2021). Analisi dei dati in ambito di comunità

Achronyms & Greek letters

- CFA: confirmatory factor analysis
- LM: linear models/modeling
- LV: latent variable
- OV: observed variable
- SEM: structural equation models/modeling
- SS: sum of squares

- β = *beta*, indexing path coefficients (or regression coefficients)
- ϵ = *epsilon*, indexing the error of an observed variable
- σ = *sigma*, indexing the variance σ^2 of the errors ϵ
- η = *eta*, indexing latent variables
- θ = *theta*, indexing overall model parameters

Achronyms & Greek letters

- CFA: confirmatory factor analysis
- LM: linear models/modeling
- LV: latent variable
- OV: observed variable
- SEM: structural equation models/modeling
- SS: sum of squares

- β = *beta*, indexing path coefficients (or regression coefficients)
- ϵ = *epsilon*, indexing the error of an observed variable
- σ = *sigma*, indexing the variance σ^2 of the errors ϵ
- η = *eta*, indexing latent variables
- θ = *theta*, indexing overall model parameters
- ciao