

Analisi fattoriale confermativa multilivello

Introduzione, sintassi ed esempi pratici

Luca Menghini Ph.D.

Dipartimento di Psicologia Generale,
Università degli Studi di Padova

luca.menghini@unipd.it

Psicostat handZone

Università degli Studi di Padova
26 Marzo 2024



Outline

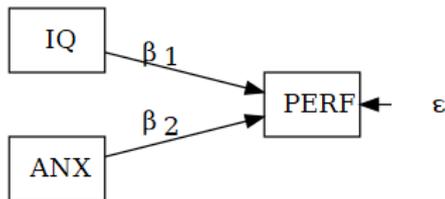
- SEM, CFA e rationale della CFA multilivello
- HandZone: Come condurre una CFA multilivello con `lavaan`
- Esempio pratico: Gruppi e individui
- Esempi pratici: Misure ripetute
- Invarianza cross-livello: Dalla pratica alla teoria

Multilevel what!?

Modelli di equazioni strutturali (SEM):

modelli *lineari* multivariati formalizzati da sistemi di equazioni

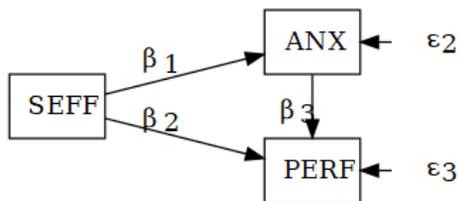
Modelli lineari 'standard': stimano la relazione tra una variabile dipendente e 1+ predittori attraverso una singola equazione tipo: $PERF = \beta_1 IQ + \beta_2 ANX + \epsilon$



Possono predire solo **una variabile dipendente alla volta** → *univariati* (solo intercetta) o *bivariati* (intercetta + pendenze).

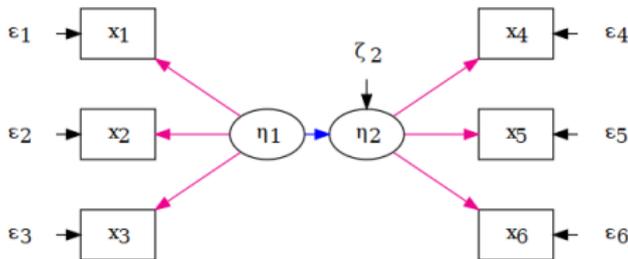
SEM: permettono di modellare simultaneamente più variabili ~~dependenti~~ *endogene* attraverso un sistema di equazioni tipo:

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



Le due parti fondamentali di un SEM

I SEM permettono di quantificare le relazioni tra **variabili latenti** (cerchi), a partire dalla matrice varianza-covarianza tra un set di **variabili osservate** (quadrati).



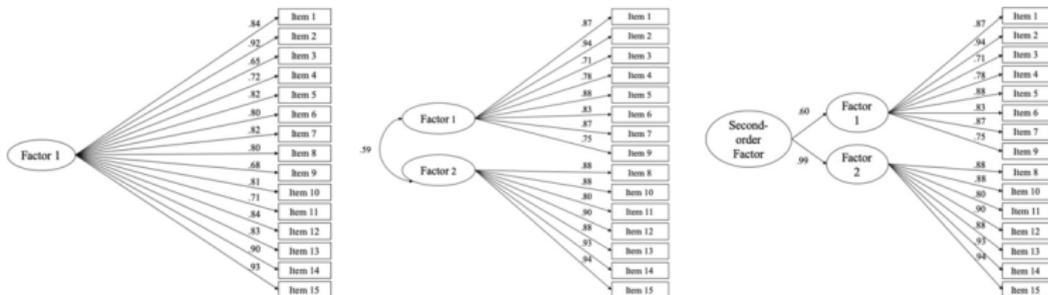
Un SEM include generalmente due parti:

1. **Modello strutturale**: Relazioni “regression-like” tra variabili latenti o tra variabili osservate (*path analysis*)
2. **Modello di misurazione** (*latent variable model*): Relazioni tra le variabili latenti e le variabili osservate corrispondenti (indicative delle latenti) → CFA

Analisi fattoriale confermativa (CFA)

Un modello di CFA include soltanto il modello di misurazione (senza modello strutturale) per ‘formare’ le variabili latenti e quantificarne le relazioni con le relative variabili osservate.

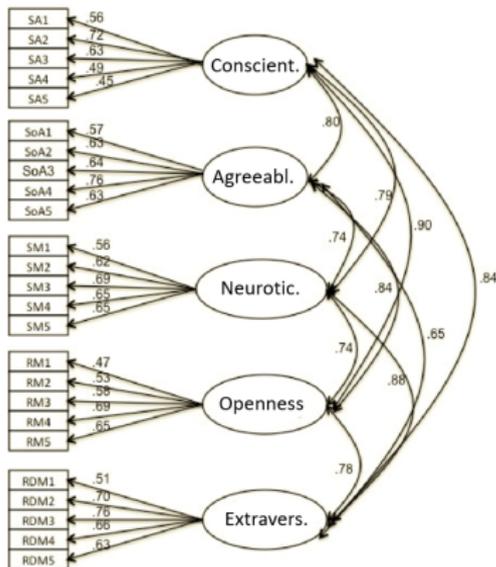
Queste relazioni sono chiamate **saturationi** (*factor loadings*) e sono considerate indicatori quantitativi della **validità di costrutto** di un set di indicatori osservati (es. item di un questionario).



Struttura fattoriale

Una struttura fattoriale (*factor structure*) è una delle possibili configurazioni di relazioni tra un set di variabili osservate che si assumono essere indicative di un particolare fattore latente, definendo:

1. il **numero** di variabili latenti (es. modello a uno, due, . . . , n fattori)
2. le **relazioni** tra ciascuna variabile osservata e la variabile latente corrispondente



A partire dalla matrice di varianza-covarianza delle variabili osservate, una CFA testa la **bontà di adattamento** di una o più strutture fattoriali ipotizzate e fornisce una stima delle relative **saturazioni**.

Il punto di partenza: Matrici di varianza-covarianza

Il punto di partenza di un modello lineare ‘standard’ è un vettore (o un set di vettori) di valori attribuiti a una o più variabile (i.e., colonne di un dataset):

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

	x1	x2	x3	x4
1	57	21	15	18
2	77	22	19	17
3	51	13	13	16
4	58	24	6	21

Il punto di partenza di un SEM (inclusa CFA) è la **matrice di covarianza tra le variabili osservate**:

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

```
cov(df)
```

	x1	x2	x3	x4
x1	100.70	24.89	17.21	7.99
x2	24.89	19.43	6.69	4.04
x3	17.21	6.69	17.33	2.23
x4	7.99	4.04	2.23	5.34

I SEM (incluse le CFA) stima un certo numero di parametri θ in modo che la **matrice di covarianza predetta** sulla base dei parametri stimati $\hat{\Sigma}(\theta)$ sia quanto più ‘vicina’ alla **matrice di covarianza osservata** S .

Nota: anche i parametri del modello sono stimati in **matrici di parametri** 

E se i dati sono multilivello?

Quando le nostre osservazioni sono **nidificate** entro dei fattori di raggruppamento (**cluster**), si creano delle **dipendenze locali** (correlazioni tra osservazioni appartenenti allo stesso cluster) che violano l'assunto di indipendenza tra le osservazioni e possono produrre delle stime distorte degli errori standard.

Ad esempio:

- soggetti nidificati entro **gruppi**
(es. scuole, classi, organizzazioni)
- misure ripetute nidificate **entro i soggetti** (es. disegni longitudinali intensivi)

```
head(df, 10)
```

	cluster	x1	x2	x3	x4
1	1	57	21	15	18
2	1	77	22	19	17
3	1	51	13	13	16
4	1	58	24	6	21
5	1	64	28	15	19
6	2	68	27	14	19
7	2	91	21	23	20
8	2	46	11	1	16
9	2	61	24	18	21
10	2	75	31	15	22

Scomposizione della (co)varianza

In questi casi, è possibile **scomporre la (co)varianza** su due (o più) livelli.

Livello 2: Between-clusters

Punteggi medi ottenuti da ciascun cluster per ogni variabile.

```
# calcolo medie per cluster
wide <-
  cbind(
    aggregate(
      x1~cluster, data=df, FUN=mean),
    x2 = aggregate(
      x2~cluster, data=df, FUN=mean)[,2])
head(wide)
```

	cluster	x1	x2
1	1	61.4	21.6
2	2	68.2	22.8
3	3	68.4	21.8
4	4	66.0	21.0
5	5	66.8	22.0
6	6	74.8	25.8

Livello 1: Within-clusters

Punteggi centrati sulla media di ogni cluster.

```
# aggiungo le medie al dataset long
colnames(wide)[2:3] <- c("x1.b", "x2.b")
df <- plyr::join(df, wide, by="cluster")

# calcolo deviazioni dalla media
df$x1.w <- df$x1 - df$x1.b
df$x2.w <- df$x2 - df$x2.b
head(df[,c("cluster", "x1", "x1.b", "x1.w")])
```

	cluster	x1	x1.b	x1.w
1	1	57	61.4	-4.4
2	1	77	61.4	15.6
3	1	51	61.4	-10.4
4	1	58	61.4	-3.4
5	1	64	61.4	2.6
6	2	68	68.2	-0.2

Matrici livello-specifiche

A partire da questi valori, è possibile calcolare due matrici di covarianza distinte.

Livello 2: Between-clusters

Covarianza tra i punteggi medi:

```
cov(wide[,c("x1.b", "x2.b")])
```

	x1.b	x2.b
x1.b	25.140797	6.148043
x2.b	6.148043	3.008043

I cluster con un QI medio più elevato hanno anche un MAT medio più elevato?

Livello 1: Within-clusters

Covarianza tra i punteggi centrati:

```
cov(df[,c("x1.w", "x2.w")])
```

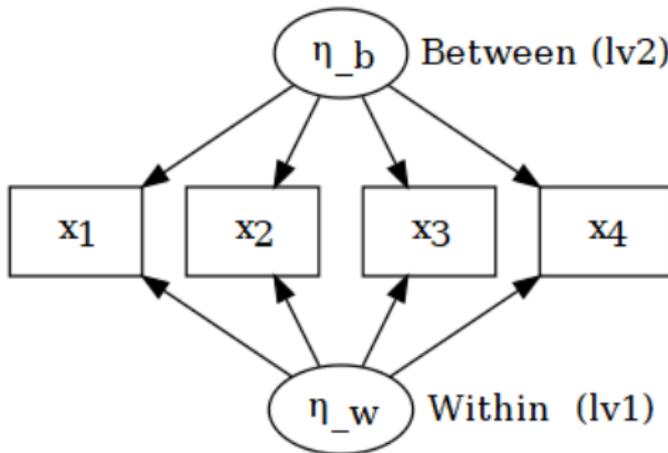
	x1.w	x2.w
x1.w	76.40672	18.94790
x2.w	18.94790	16.52101

I soggetti con un QI più elevato della media del loro gruppo hanno anche un MAT più alto della media?

Nelle occasioni in cui un soggetto mostra un QI più alto del solito, anche il suo MAT è più elevato del solito?

Multilevel CFA

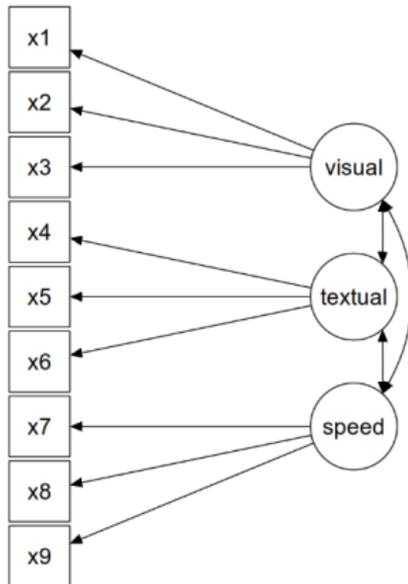
L'analisi fattoriale multilivello parte da queste due matrici di varianza-covarianza per testare la bontà di adattamento di una o più strutture fattoriali e quantificarne le saturazioni su entrambi i livelli.



HandZone: CFA con 

```
data( HolzingerSwineford1939, package = "lavaan" )  
p <- cov(HolzingerSwineford1939[,paste0("x",1:9)])  
p[upper.tri(p)] <- NA  
round(p,2)
```

```
##      x1    x2    x3    x4    x5    x6    x7    x8    x9  
## x1 1.36    NA    NA    NA    NA    NA    NA    NA    NA  
## x2 0.41  1.39    NA    NA    NA    NA    NA    NA    NA  
## x3 0.58  0.45  1.28    NA    NA    NA    NA    NA    NA  
## x4 0.51  0.21  0.21  1.36    NA    NA    NA    NA    NA  
## x5 0.44  0.21  0.11  1.10  1.67    NA    NA    NA    NA  
## x6 0.46  0.25  0.24  0.90  1.02  1.20    NA    NA    NA  
## x7 0.09 -0.10  0.09  0.22  0.14  0.14  1.19    NA    NA  
## x8 0.26  0.11  0.21  0.13  0.18  0.17  0.54  1.03    NA  
## x9 0.46  0.24  0.38  0.24  0.30  0.24  0.37  0.46  1.02
```



CFA con lavaan: Fittare un modello su dei dati

In `lavaan` le saturazioni sono indicate con il simbolo “`=~`” (“*is reflected by*”).

```
# specifico modello a tre fattori
```

```
m1 <- 'visual =~ x1 + x2 + x3  
      textual =~ x4 + x5 + x6  
      speed =~ x7 + x8 + x9'
```

```
# fitto modello
```

```
library(lavaan)  
fit1 <- cfa(model = m1, data = HolzingerSwineford1939)
```

```
# specifico e fitto modello alternativo a un fattore
```

```
m2 <- 'general =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9'  
fit2 <- cfa(model = m2, data = HolzingerSwineford1939)
```

CFA con lavaan: Indici di fit e confronto tra modelli

Gli indici di fit sono generalmente indicativi della **similarità/distanza** tra la matrice di covarianza osservata e quella predetta dal modello sulla base dei parametri stimati.

```
round(  
  lavInspect(fit1, what = "fit")[c("rmsea","cfi","tli","srmr")], 3)  
  
## rmsea  cfi  tli  srmr  
## 0.092 0.931 0.896 0.065
```

Gli indici di fit e altri criteri (es. AIC e BIC) possono essere usati per confrontare più modelli (strutture fattoriali alternative).

```
round(  
  lavInspect(fit2, what = "fit")[c("rmsea","cfi","tli","srmr")], 3)  
  
## rmsea  cfi  tli  srmr  
## 0.187 0.677 0.569 0.143
```

```
library(MuMIn)  
Weights(AIC(fit1, fit2))  
  
## model weights  
## [1] 1 0
```

```
Weights(BIC(fit1, fit2))  
  
## model weights
```

CFA con lavaan: Parametri stimati

Una volta selezionato un modello, possiamo valutare le saturazioni:

```
parameterestimates(fit1)[1:9,] # soluzione non standardizzata
```

##	lhs	op	rhs	est	se	z	pvalue	ci.lower	ci.upper
## 1	visual	--	x1	1.000	0.000	NA	NA	1.000	1.000
## 2	visual	--	x2	0.554	0.100	5.554	0	0.358	0.749
## 3	visual	--	x3	0.729	0.109	6.685	0	0.516	0.943
## 4	textual	--	x4	1.000	0.000	NA	NA	1.000	1.000
## 5	textual	--	x5	1.113	0.065	17.014	0	0.985	1.241
## 6	textual	--	x6	0.926	0.055	16.703	0	0.817	1.035
## 7	speed	--	x7	1.000	0.000	NA	NA	1.000	1.000
## 8	speed	--	x8	1.180	0.165	7.152	0	0.857	1.503
## 9	speed	--	x9	1.082	0.151	7.155	0	0.785	1.378

```
standardizedsolution(fit2)[1:9,] # soluzione standardizzata
```

##	lhs	op	rhs	est.std	se	z	pvalue	ci.lower	ci.upper
## 1	general	--	x1	0.438	0.050	8.705	0.000	0.339	0.537
## 2	general	--	x2	0.220	0.058	3.772	0.000	0.106	0.335
## 3	general	--	x3	0.223	0.058	3.816	0.000	0.108	0.337
## 4	general	--	x4	0.848	0.023	37.368	0.000	0.803	0.892
## 5	general	--	x5	0.841	0.023	36.402	0.000	0.796	0.886
## 6	general	--	x6	0.838	0.023	35.963	0.000	0.792	0.884
## 7	general	--	x7	0.180	0.059	3.041	0.002	0.064	0.297
## 8	general	--	x8	0.201	0.059	3.416	0.001	0.086	0.317
## 9	general	--	x9	0.307	0.056	5.494	0.000	0.197	0.416

CFA con lavaan: Cmq sarebbero dati ordinali :)

```
fit <- cfa(model = m1, data = HolzingerSwineford1939,  
           ordered = TRUE) # ;)
```

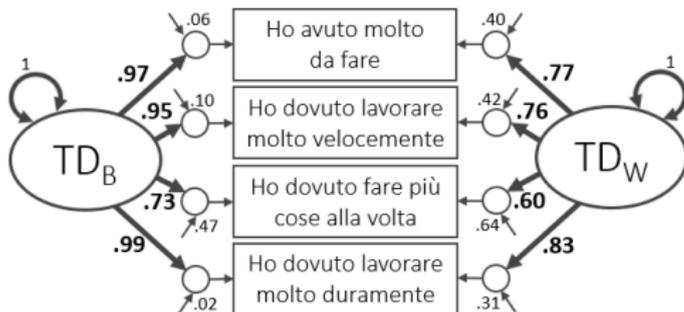
MCFA con 

```
# scarico e leggo dataset da repository OSF
library(osfr)
repo <-
  osf_retrieve_node(
    "https://doi.org/10.17605/OSF.IO/87A9P")
osf_download(osf_ls_files(repo)[1, ],
  conflicts="overwrite")
load("S5_processedData/ESM_processed.RData")
```

```
# item job-related Task Demand (TD)
head(ESMdata[,c("ID", "day",
                "d1", "d2", "d3", "d4")],
  18)
```

##	ID	day	d1	d2	d3	d4
## 1	S001	1	5	4	3	3
## 2	S001	1	5	2	4	3
## 3	S001	1	5	3	2	2
## 4	S001	1	5	5	3	5
## 5	S001	2	3	2	2	3
## 6	S001	2	1	1	2	2
## 7	S001	2	4	5	3	3
## 8	S001	3	5	3	6	4
## 9	S001	3	5	2	5	5
## 10	S001	3	6	6	3	6
## 11	S001	3	5	5	3	5
## 12	S002	1	NA	NA	NA	NA
## 13	S002	1	3	1	1	2
## 14	S002	1	5	4	3	4
## 15	S002	1	5	3	3	4
## 16	S002	1	6	4	3	6
## 17	S002	2	4	3	5	4
## 18	S002	2	5	3	6	7

Pensando all'attività lavorativa che stavi svolgendo negli ultimi 10 minuti:



Multilevel CFA: Fittare un modello su dei dati

Con un modello a due livelli, possiamo usare la sintassi “level: 1” e “level: 2” per specificare la struttura fattoriale sui due livelli, aggiungendo l’argomento `cluster = "cluster_name"` nella funzione `cfa()`.

```
# specifico modello a tre fattori
m1 <- 'level: 1
      TD_w =~ d1 + d2 + d3 + d4
      level: 2
      TD_b =~ d1 + d2 + d3 + d4'

# fitto modello
library(lavaan)
fit1 <- cfa(model = m1, data = ESMdata, cluster = "ID")
```

Multilevel CFA: Indici di fit livello-specifici

Gli indici come RMSEA, CFI e TLI sono **indici di fit “globali”** che sono per lo più imputabili a informazioni sul fit a livello within (perché ci sono più osservazioni a livello 1) mentre sono risultati poco sensibili a “misspecification” a livello between.

Al contrario, **l'SRMR è un indice livello-specifico** che può essere calcolato separatamente per la matrice between e quella within. Quindi **l'SRMR between** è l'unico indice informativo del fit a livello 2 (Ryu & West, 2009; Hsu et al. 2015).

```
round(
  lavInspect(fit1, what = "fit")[
    c("rmsea", "cfi", "srmr_within", "srmr_between")
  ], 3)

##      rmsea      cfi  srmr_within  srmr_between
##      0.058      0.992      0.012      0.036
```

Multilevel CFA: Parametri livello-specifici

```
# Saturazioni standardizzate a livello within:
```

```
standardizedsolution(fit1)[1:4,]
```

##	lhs	op	rhs	est.std	se	z	pvalue	ci.lower	ci.upper
## 1	TD_w	--	d1	0.773	0.014	56.239	0	0.746	0.800
## 2	TD_w	--	d2	0.762	0.014	53.608	0	0.734	0.790
## 3	TD_w	--	d3	0.621	0.018	33.609	0	0.584	0.657
## 4	TD_w	--	d4	0.832	0.012	67.338	0	0.807	0.856

```
# Saturazioni standardizzate a livello between:
```

```
standardizedsolution(fit1)[15:18,]
```

##	lhs	op	rhs	est.std	se	z	pvalue	ci.lower	ci.upper
## 15	TD_b	--	d1	0.961	0.013	74.193	0	0.935	0.986
## 16	TD_b	--	d2	0.935	0.016	56.899	0	0.903	0.967
## 17	TD_b	--	d3	0.607	0.059	10.267	0	0.491	0.723
## 18	TD_b	--	d4	0.996	0.010	104.200	0	0.977	1.015

Nota: Le saturazioni sono sempre più forti a livello between perché l'errore di misura tende ad accumularsi a livello 1 (Hox, 2010)

Multilevel CFA: Cmq sarebbero dati ordinali

```
fit <- cfa(model = m1, data = ESMdata, cluster = "ID",  
           ordered = TRUE) # :(
```

```
Error in lavaan::lavaan(model = m1, data = ESMdata, ordered = TRUE, cluster = "ID", :  
lavaan ERROR: categorical + clustered is not supported yet.
```

Soggetti nidificati entro gruppi

Quando reclutiamo i soggetti da diversi gruppi (es. classi, scuole, organizzazioni), dovremmo tenere conto delle **dipendenze locali**: correlazioni più forti tra soggetti appartenenti allo stesso gruppo che tra soggetti appartenenti a gruppi diversi.

Esempio con dati reali

Individual vs. Team participation

Menghini, Balducci & Toderi (*In preparation*)

- *Campione*: 608 educatrici/tori nidificate/i in 68 scuole d'infanzia
- *Misura*: Grado di partecipazione a programmi di prevenzione e gestione dei rischi psicosociali (es. “Sei stata/o coinvolta/o nelle attività di valutazione e riduzione dello stress?”) - 4 item da 1 (Per niente) a 5 (Moltissimo)

```
head(edudata[,c("schoolID",  
                paste0("p",2:5))],4)
```

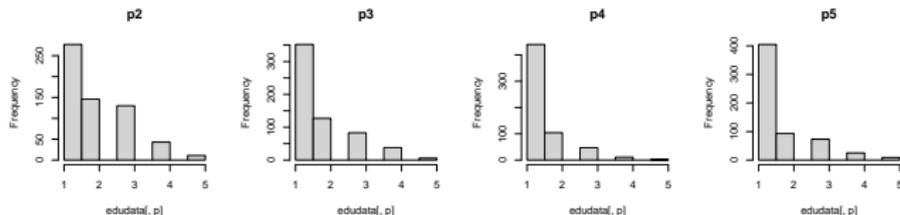
```
##  schoolID p2 p3 p4 p5  
## 1    BO_01 1 1 1 1  
## 2    BO_01 1 1 2 2  
## 3    BO_01 1 1 1 1  
## 4    BO_01 1 1 1 1
```

```
p <- cor(edudata[,paste0("p",2:5)],  
        use="complete.obs")  
p[upper.tri(p)] <- NA  
round(p, 2)
```

```
##      p2  p3  p4 p5  
## p2 1.00  NA  NA NA  
## p3 0.70 1.00  NA NA  
## p4 0.50 0.53 1.00 NA  
## p5 0.64 0.63 0.43 1
```

Single-level CFA

```
par(mfrow=c(1,5)); for(p in paste0("p",2:5)){hist(edudata[,p],main=p)}
```



```
m1 <- 'EP =~ p2 + p3 + p4 + p5'
# Maximum Likelihood robust estimator
fit1 <- cfa(m1, edudata, estimator = "MLR")
round(lavInspect(fit1,"fit")[c("chisq","df","rmsea.robust","cfi.robust","srmr")],3)
```

##	chisq	df	rmsea.robust	cfi.robust	srmr
##	3.946	2.000	0.028	0.999	0.012

```
# Weighted Least Square Mean and Variance adjusted
fit2 <- cfa(m1, edudata, ordered = TRUE, estimator = "WLSMV")
round(lavInspect(fit2,"fit")[c("chisq","df","rmsea.robust","cfi.robust","srmr")],3)
```

##	chisq	df	rmsea.robust	cfi.robust	srmr
##	2.083	2.000	0.080	0.995	0.014

Multilevel CFA: Modelli preliminari (lv1)

Hox (2010, cap 14) suggerisce di specificare un set di modelli preliminari per valutare la struttura fattoriale sui due livelli:

- A **livello 1 (within)**, fittiamo una CFA ‘classica’ sulla matrice di covarianza *pooled* a livello within.

```
# listwise deletion
lwd <- na.omit(edudata[,c("schoolID",paste0("p",2:5))])
# centro i punteggi sulle medie delle scuole
ms <- data.frame(p2=ave(lwd$p2,lwd$schoolID),p3=ave(lwd$p3,lwd$schoolID),
                 p4=ave(lwd$p4,lwd$schoolID),p5=ave(lwd$p5,lwd$schoolID))
cs <- lwd[,2:ncol(lwd)] - ms
# pooled within-cluster covariance matrix
pw.cov <- (cov(cs) * (nrow(lwd) - 1))/(nrow(lwd) - nlevels(lwd$schoolID))
# fitto modello ed estraggo indici di fit
fit <- cfa(m1, sample.cov = pw.cov, sample.nobs=nrow(lwd))
round(lavInspect(fit,"fit")[c("chisq","df","rmsea","cfi","tli","srmr")])

## chisq   df rmsea   cfi   tli   srmr
##      8    2     0     1     1     0
```

Multilevel CFA: Modelli preliminari (lv2)

Hox (2010, cap 14) suggerisce di specificare un set di modelli preliminari per valutare la struttura fattoriale sui due livelli:

- A **livello 2 (between)**, fittiamo una serie di MCFA ‘banchmark’ al livello del cluster:
 1. **Modello nullo:** Nessuna struttura a livello 2
→ se ha un buon fit, significa che *“non c’è proprio una struttura a livello 2”* (meglio usare CFA classica)
 2. **Modello indipendente:** Solo varianze ma non covarianze a livello 2
→ se ha un buon fit, significa che *“c’è una struttura a livello 2 ma non un modello strutturale sostanzialmente interessante”* (meglio usare la matrice within pooled)
 3. **Modello saturo:** Rendiamo il modello saturo a livello 2
→ se ha un buon fit, significa che *“il costrutto ‘esiste’ solo a livello 1, mentre le covarianze a livello 2 sono spurie”*

Multilevel CFA: Modelli preliminari (lv2)

Modello nullo

```
m1 <- 'level: 1
      p.w =~ p2 + p3 + p4 + p5
      level: 2
      p2 ~~ 0*p2
      p3 ~~ 0*p3
      p4 ~~ 0*p4
      p5 ~~ 0*p5'
```

Modello indipendente

```
m2 <- 'level: 1
      p.w =~ p2 + p3 + p4 + p5
      level: 2
      p2 ~~ p2
      p3 ~~ p3
      p4 ~~ p4
      p5 ~~ p5'
```

Modello saturo

```
m3 <- 'level: 1
      p.w =~ p2 + p3 + p4 + p5
      level: 2
      p2 ~~ p2 + p3 + p4 + p5
      p3 ~~ p3 + p4 + p5
      p4 ~~ p4 + p5
      p5 ~~ p5'
```

Fitto modelli ed estraggo gli indici di fit:

```
fit1 <- cfa(m1,edudata,cluster="schoolID",estimator="MLR")
fit2 <- cfa(m2,edudata,cluster="schoolID",estimator="MLR")
fit3 <- cfa(m3,edudata,cluster="schoolID",estimator="MLR")
f <- c("df","rmsea.robust","cfi.robust","srmr_within","srmr_between")
round( rbind(lavInspect(fit1,"fit")[f],lavInspect(fit2,"fit")[f],
             lavInspect(fit3,"fit")[f]), 3)
```

##	df	rmsea.robust	cfi.robust	srmr_within	srmr_between
## [1,]	12	0.155	0.800	0.078	NA
## [2,]	8	0.163	0.851	0.083	0.652
## [3,]	2	0.055	0.996	0.018	0.008

MCFA

Modello ipotizzato (notare che alcuni cluster non mostrano variabilità)

```
m4 <- 'level: 1
      p.w =~ p2 + p3 + p4 + p5
      level: 2
      p.b =~ p2 + p3 + p4 + p5'

fit4 <- cfa(m4, edudata, cluster = "schoolID", estimator = "MLR")

## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:
## Level-1 variable "p2" has no variance within some clusters. The
## cluster ids with zero within variance are: BO_01 BO_14 CE_4 FE_05
## FE_2 FE_4 IM_4 MO_4 PC_05 RA_6 RE_02 RN_3 VIG_2
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:
## Level-1 variable "p3" has no variance within some clusters. The
## cluster ids with zero within variance are: BO_13 BO_14 BO_16 BO_17
## CE_2 FE_2 FE_4 MO_SU PR_08 RA_2 RE_09 RN_2 RN_3
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:
## Level-1 variable "p4" has no variance within some clusters. The
## cluster ids with zero within variance are: BO_14 BO_17 CE_2 CE_3
## FE_01 FE_05 FE_2 FE_4 IM_4 MO_1 MO_N_1 MO_SU PC_1 PR_08 RA_6 RE_02
## RE_08 RE_09 RE_10 RN_1 VIG_2
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:
## Level-1 variable "p5" has no variance within some clusters. The
## cluster ids with zero within variance are: BO_14 BO_16 BO_17 CE_2
## FE_05 FE_2 FE_4 MO_SU MO_N_4 PAV_1 PC_1 PR_08 RA_6 RA_8 RE_02 RE_06
## RE_08 RE_09 RN_2 RN_3
```

MCFA model comparison

Modello ipotizzato (notare che alcuni cluster non mostrano variabilità)

```
round( rbind(lavInspect(fit1,"fit")[f],lavInspect(fit2,"fit")[f],  
            lavInspect(fit3,"fit")[f],lavInspect(fit4,"fit")[f]), 3)
```

```
##      df rmsea.robust cfi.robust smmr_within smmr_between  
## [1,] 12      0.155      0.800      0.078          NA  
## [2,]  8      0.163      0.851      0.083      0.652  
## [3,]  2      0.055      0.996      0.018      0.008  
## [4,]  4      0.070      0.986      0.018      0.055
```

```
Weights(AIC(fit1,fit2,fit3,fit4))
```

```
## model weights  
## [1] 0.000 0.000 0.935 0.065
```

```
Weights(BIC(fit1,fit2,fit3,fit4))
```

```
## model weights  
## [1] 0.00 0.00 0.15 0.85
```

Non proprio alla grande :(

Saturazioni livello-specifiche

```
(sl <- standardizedsolution(fit4)[c(1:4,15:18),1:6])
```

```
##   lhs op rhs est.std   se    z
## 1 p.w == p2  0.767 0.044 17.601
## 2 p.w == p3  0.769 0.037 20.655
## 3 p.w == p4  0.559 0.053 10.500
## 4 p.w == p5  0.670 0.045 15.025
## 15 p.b == p2  0.984 0.019 53.193
## 16 p.b == p3  0.986 0.022 45.043
## 17 p.b == p4  0.811 0.108  7.520
## 18 p.b == p5  0.933 0.047 19.708
```

Nota: le saturazioni possono essere usate per stimare dei coefficienti di attendibilità ω livello-specifici (Geldhof et al., 2014).

```
sl.w <- sl[1:4,"est.std"] # saturazioni within
re.w <- 1 - sl.w^2 # varianza res. within
round( sum(sl.w)^2 / (sum(sl.w)^2 + sum(re.w)), 2) # omega within

## [1] 0.79
```

```
sl.b <- sl[5:8,"est.std"] # saturazioni between
re.b <- 1 - sl.b^2 # varianza res. between
round( sum(sl.b)^2 / (sum(sl.b)^2 + sum(re.b)), 2) # omega between

## [1] 0.96
```

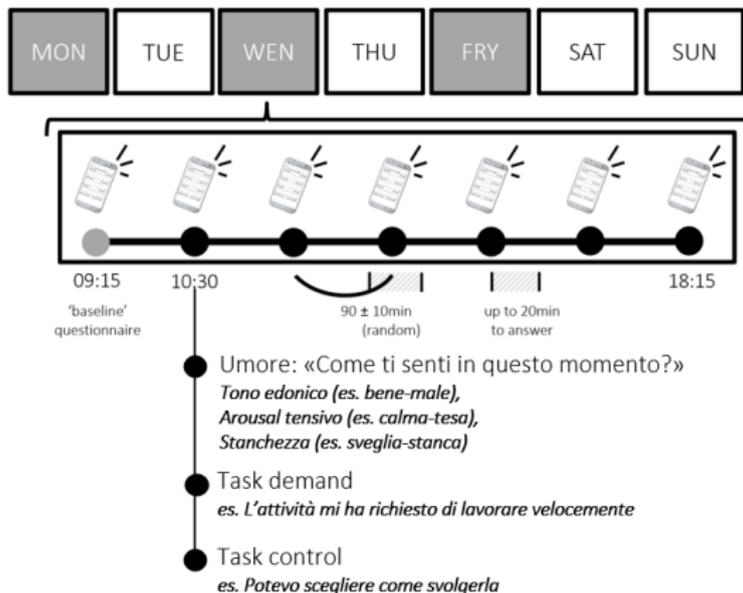
Osservazioni nidificate entro soggetti

Quando somministriamo gli stessi item agli stessi soggetti in più occasioni (es. disegno longitudinale intensivo), dovremmo tenere conto delle **dipendenze locali**: correlazioni più forti tra i punteggi forniti dallo stesso soggetto che tra i punteggi forniti da soggetti diversi.

Esempio con dati reali: *Stressor e strain*

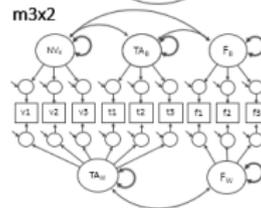
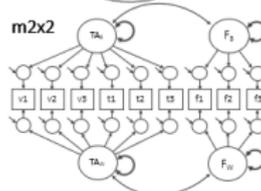
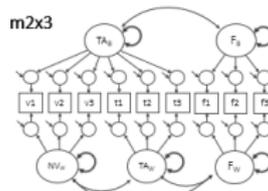
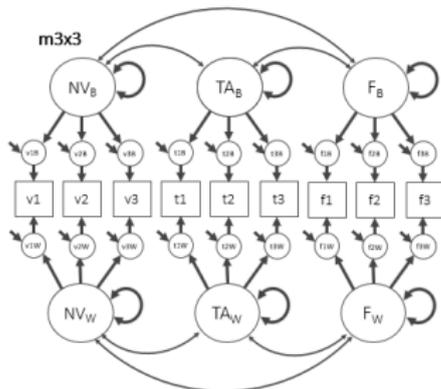
Menghini, Pastore & Balducci (2023) <https://doi.org/10.1027/1015-5759/a000725>

- *Campione*: 139 lavoratrici/tori d'ufficio
- *Procedura*: 7 questionari al giorno per 3 giorni lavorativi non consecutivi



MCFA con più di un fattore

Quando abbiamo una struttura fattoriale ipotizzata con più di un fattore possiamo confrontarla con strutture alternative manipolando il numero di fattori su entrambi i livelli.



MCFA con più di un fattore

```
m3x3 <-
```

```
'level: 1
```

```
NV_w =~ v1 + v2 + v3
```

```
TA_w =~ t1 + t2 + t3
```

```
FA_w =~ f1 + f2 + f3
```

```
level: 2
```

```
NV_b =~ v1 + v2 + v3
```

```
TA_b =~ t1 + t2 + t3
```

```
FA_b =~ f1 + f2 + f3'
```

```
m2x3 <-
```

```
'level: 1
```

```
NV_w =~ v1 + v2 + v3
```

```
TA_w =~ t1 + t2 + t3
```

```
FA_w =~ f1 + f2 + f3
```

```
level: 2
```

```
NV_b =~ v1 + v2 + v3 + t1 + t2 + t3
```

```
FA_b =~ f1 + f2 + f3'
```

```
m2x2 <-
```

```
'level: 1
```

```
NV_w =~ v1 + v2 + v3 + t1 + t2 + t3
```

```
FA_w =~ f1 + f2 + f3
```

```
level: 2
```

```
NV_b =~ v1 + v2 + v3 + t1 + t2 + t3
```

```
FA_b =~ f1 + f2 + f3'
```

```
m3x2 <-
```

```
'level: 1
```

```
NV_w =~ v1 + v2 + v3 + t1 + t2 + t3
```

```
FA_w =~ f1 + f2 + f3
```

```
level: 2
```

```
NV_b =~ v1 + v2 + v3
```

```
TA_b =~ t1 + t2 + t3
```

```
FA_b =~ f1 + f2 + f3'
```

Fitto i modelli

```
fit2x2 <- cfa(m2x2, ESMdata, cluster="ID")
```

```
fit3x2 <- cfa(m3x2, ESMdata, cluster="ID")
```

```
fit2x3 <- cfa(m2x3, ESMdata, cluster="ID")
```

```
fit3x3 <- cfa(m3x3, ESMdata, cluster="ID")
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "v1" has no variance within some clusters. The  
## cluster ids with zero within variance are: S045 S047 S049 S123
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "v2" has no variance within some clusters. The  
## cluster ids with zero within variance are: S089 S040
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "v3" has no variance within some clusters. The  
## cluster ids with zero within variance are: S049 S065 S089 S123
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "t2" has no variance within some clusters. The  
## cluster ids with zero within variance are: S045 S123
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "t3" has no variance within some clusters. The  
## cluster ids with zero within variance are: S031 S049 S058 S097
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "f1" has no variance within some clusters. The  
## cluster ids with zero within variance are: S031 S047 S089
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING:  
## Level-1 variable "f3" has no variance within some clusters. The  
## cluster ids with zero within variance are: S055 S123
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov  
## variances are negative
```

Varianze negative!? 😬

Il problema delle **varianze negative a livello 2** (Heywood cases o soluzioni inappropriate) è molto comune nelle MCFA perché l'errore tende ad accumularsi a livello 1, mentre a livello 2 ci sono spesso saturazioni molto forti e (di conseguenza) varianze residue molto basse, vicine allo zero (Hox, 2010).

Kolenikov & Bollen (2012) suggeriscono una serie di controlli per valutare se queste varianze negative possono essere dovute ad una 'misspecification' empirica (saturazioni vicine a zero, overfactoring, correlazioni tra fattori vicine a 0 o a 1) o strutturale (se i 95% CI non includono lo zero) o ad altre ragioni (es. problemi di convergenza, dati mancanti).

```
p <- parameterestimates(fit3x3)
p[p$op=="~~" & p$ci.lower<0,]
```

##	lhs	op	rhs	block	level	est	se	z	pvalue	ci.lower	ci.upper
## 48	v3	~~	v3	2	2	0.002	0.012	0.197	0.844	-0.022	0.027
## 50	t2	~~	t2	2	2	0.010	0.011	0.946	0.344	-0.011	0.032
## 51	t3	~~	t3	2	2	-0.002	0.010	-0.226	0.821	-0.022	0.018
## 54	f3	~~	f3	2	2	0.028	0.016	1.763	0.078	-0.003	0.058

Varianze negative e analisi dei casi influenti

Quando possiamo escludere queste possibili spiegazioni, è probabile che la varianza negativa sia il risultato di **fluttuazioni campionarie** intorno ad un parametro di popolazione vicino a zero.

In questo caso, un approccio utile è quello dell'**analisi dei casi influenti**: escludiamo i soggetti (o le osservazioni) uno alla volta a partire da quelli la cui esclusione aumenta il valore della varianza negativa, finché la varianza non torna negativa.

```
clean <- ESMdata[!ESMdata$ID %in% c("S017", "S035", "S139", "S008", "S106", "S142"), ]  
  
fit2x2 <- cfa(m2x2, clean, cluster="ID")  
fit3x2 <- cfa(m3x2, clean, cluster="ID")  
fit2x3 <- cfa(m2x3, clean, cluster="ID")  
fit3x3 <- cfa(m3x3, clean, cluster="ID")
```

Niente male! ;)

Confronto tra modelli

```
f <- c("df", "rmsea", "cfi", "srmr_within", "srmr_between")
round( rbind(lavInspect(fit2x2, "fit")[f], lavInspect(fit3x2, "fit")[f],
             lavInspect(fit2x3, "fit")[f], lavInspect(fit3x3, "fit")[f]), 3)
```

```
##      df rmsea  cfi srmr_within srmr_between
## [1,] 52 0.071 0.935      0.044      0.046
## [2,] 50 0.066 0.945      0.044      0.045
## [3,] 50 0.059 0.956      0.030      0.045
## [4,] 48 0.055 0.963      0.030      0.044
```

```
Weights(AIC(fit2x2, fit3x2, fit2x3, fit3x3))
```

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

```
Weights(BIC(fit2x2, fit3x2, fit2x3, fit3x3))
```

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

Invarianza tra gruppi

Problema: stiamo misurando la stessa cosa in tutti i soggetti/cluster?

Solitamente rispondiamo a questa domanda con una **CFA multigruppo**, in cui applichiamo una serie di restrizioni (*constraints*) progressivamente restrittive per rendere i parametri invariati tra gruppi.

#	Invariance	Constrained parameters	Comparison model
0	Separate models	-	-
1	Configural	None	-
2	Metric	λ_{ij}	Configural
3	Scalar	λ_{ij}, τ_i	Metric
4	Observed residual var.	$\lambda_{ij}, \tau_i, \theta_{ij}^{\delta}$	Scalar
5	Latent variances	$\lambda_{ij}, \tau_i, \theta_{ij}^{\delta}, \phi_{ii}$	Observed residual var.
6	Latent covariances	$\lambda_{ij}, \tau_i, \theta_{ij}^{\delta}, \phi_{ii}, \phi_{ij}$	Latent variances
7	Latent means	$\lambda_{ij}, \tau_i, \theta_{ij}^{\delta}, \phi_{ii}, \phi_{ij}, \kappa_i$	Latent covariances

- **Steps from 1 to 4:** *Measurement Invariance*
- **Steps from 5 to 7:** *Structural Invariance*

Invarianza tra cluster!? 😊

Però la CFA multigruppo diventa impraticabile quando abbiamo tanti cluster (nel nostro dataset sono 139). Come fare?

Ancora una volta entra in gioco la **logica multilivello**: trattiamo i gruppi come **effetti random** anziché come effetti fissi e ci concentriamo sull'**invarianza cross-livello**. Diversi studi di simulazione (es. Jak & Jorgensen, 2017) mostrano infatti la corrispondenza tra le assunzioni dell'invarianza tra gruppi e quelle dell'invarianza tra livello within e livello between.

	Restrictions in multigroup model	Implications in two-level model	Implications reliability
LEVEL OF FACTORIAL INVARIANCE			
Configural	$\text{pattern}(\Lambda_g) = \text{pattern}(\Lambda)$	-	
Weak	$\Lambda_g = \Lambda$	$\Lambda_{\text{WITHIN}} = \Lambda_{\text{BETWEEN}}$	
Strong	$\Lambda_g = \Lambda, \nu_g = \nu$	$\Lambda_{\text{WITHIN}} = \Lambda_{\text{BETWEEN}}, \Theta_{\text{BETWEEN}} = 0$	$\omega_{\text{BETWEEN}} = 1$

ν is a p -dimensional vector of intercepts. Subscript g is used for group/cluster.

Invarianza cross-livello: Invarianza configurale

Qui proviamo a testare i tre livelli di invarianza sui nostri dati:

Nel primo (**configurale**) specifichiamo la stessa struttura fattoriale sui due livelli

```
conf <- # invarianza configurale (stessa struttura a livello 1 e 2)
'level: 1
  NV_w =~ v1 + v2 + v3
  TA_w =~ t1 + t2 + t3
  FA_w =~ f1 + f2 + f3
level: 2
  NV_b =~ v1 + v2 + v3
  TA_b =~ t1 + t2 + t3
  FA_b =~ f1 + f2 + f3'
```

Invarianza cross-livello: Invarianza debole

Nel secondo (**debole**) specifichiamo la stessa struttura fattoriale sui due livelli e fissiamo le saturazioni (non standardizzate) perché siano equivalenti tra i due livelli.

```
weak <- # invarianza 'debole' (stessa struttura e saturazioni equivalenti)
'level: 1
  NV_w =~ a*v1 + b*v2 + c*v3
  TA_w =~ d*t1 + e*t2 + f*t3
  FA_w =~ g*f1 + h*f2 + i*f3
level: 2
  NV_b =~ a*v1 + b*v2 + c*v3
  TA_b =~ d*t1 + e*t2 + f*t3
  FA_b =~ g*f1 + h*f2 + i*f3'
```

Invarianza cross-livello: Invarianza debole

Nel terzo (**forte**) aggiungiamo l'ulteriore restrizione che le varianze residue sono uguali a zero a livello 2 (i.e., attendibilità perfetta! Ma con dati psicologici è praticamente impossibile!)

```
strong <- # invarianza 'forte' (come 'debole' + no var. residua a livello 2)
```

```
'level: 1  
NV_w =~ a*v1 + b*v2 + c*v3  
TA_w =~ d*t1 + e*t2 + f*t3  
FA_w =~ g*f1 + h*f2 + i*f3  
level: 2  
NV_b =~ a*v1 + b*v2 + c*v3  
TA_b =~ d*t1 + e*t2 + f*t3  
FA_b =~ g*f1 + h*f2 + i*f3  
v1 ~~ 0*v1  
v2 ~~ 0*v2  
v3 ~~ 0*v3  
t1 ~~ 0*t1  
t2 ~~ 0*t2  
t3 ~~ 0*t3  
f1 ~~ 0*f1
```

Confronto tra modelli

```
fit.conf <- cfa(conf, clean, cluster = "ID")
fit.weak <- cfa(weak, clean, cluster = "ID")
fit.strong <- cfa(strong, clean, cluster = "ID")

round( rbind(lavInspect(fit.conf, "fit")[f], lavInspect(fit.weak, "fit")[f],
             lavInspect(fit.strong, "fit")[f]), 3)

##      df rmsea  cfi srmr_within srmr_between
## [1,] 48 0.055 0.963      0.030      0.044
## [2,] 54 0.054 0.961      0.032      0.043
## [3,] 63 0.070 0.922      0.038      0.075
```

```
Weights(AIC(fit.conf, fit.weak, fit.strong))
```

```
## model weights
## [1] 0.987 0.013 0.000
```

```
Weights(BIC(fit.conf, fit.weak, fit.strong))
```

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

Bene ma non benissimo :|

Vediamo con i dati sulla partecipazione

```
conf <- # invarianza configurale (stessa struttura)
  'level: 1
  p.w =~ p2 + p3 + p4 + p5
  level: 2
  p.b =~ p2 + p3 + p4 + p5'

weak <- # invarianza 'debole' (stessa struttura e saturazioni equivalenti)
  'level: 1
  p.w =~ a*p2 + b*p3 + c*p4 + d*p5
  level: 2
  p.b =~ a*p2 + b*p3 + c*p4 + d*p5'

strong <- # invarianza 'forte' (come 'debole' + no var. residua a livello 2)
  'level: 1
  p.w =~ a*p2 + b*p3 + c*p4 + d*p5
  level: 2
  p.b =~ a*p2 + b*p3 + c*p4 + d*p5
  p2 ~~ 0*p2
  p3 ~~ 0*p3
  p4 ~~ 0*p4
  p5 ~~ 0*p5'
```

Vediamo con i dati sulla partecipazione

```
fit.conf <- cfa(conf, edudata, cluster = "schoolID", estimator = "MLR")
fit.weak <- cfa(weak, edudata, cluster = "schoolID", estimator = "MLR")
fit.strong <- cfa(strong, edudata, cluster = "schoolID", estimator = "MLR")

f <- c("df", "rmsea.robust", "cfi.robust", "srmr_within", "srmr_between")
round( rbind(lavInspect(fit.conf, "fit")[f],
             lavInspect(fit.weak, "fit")[f],
             lavInspect(fit.strong, "fit")[f]), 3)
```

```
##      df rmsea.robust cfi.robust srmr_within srmr_between
## [1,]  4      0.070      0.986      0.018      0.055
## [2,]  7      0.049      0.988      0.020      0.064
## [3,] 11      0.077      0.954      0.036      0.162
```

```
Weights(AIC(fit.conf, fit.weak, fit.strong))
```

```
## model weights
## [1] 0.143 0.857 0.000
```

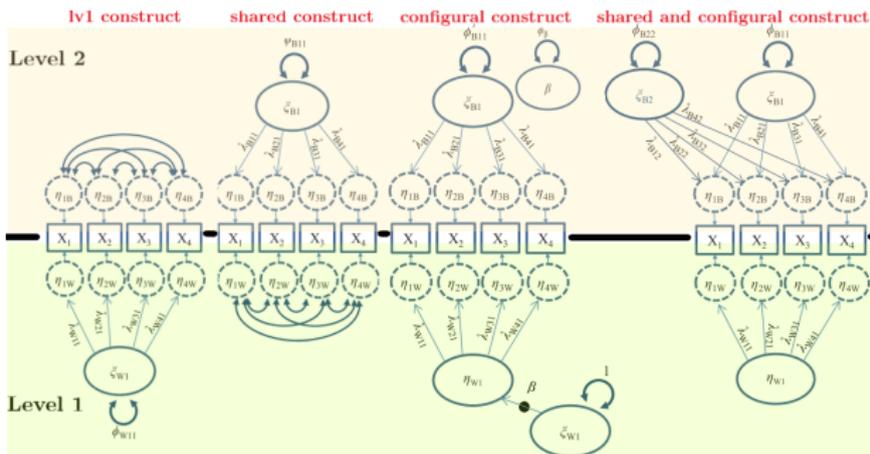
```
Weights(BIC(fit.conf, fit.weak, fit.strong))
```

```
## model weights
## [1] 0.000 0.974 0.026
```

Proprio niente male! ;)

Costrutti multilivello e invarianza cross-livello

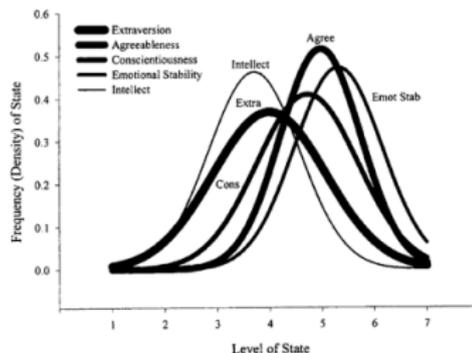
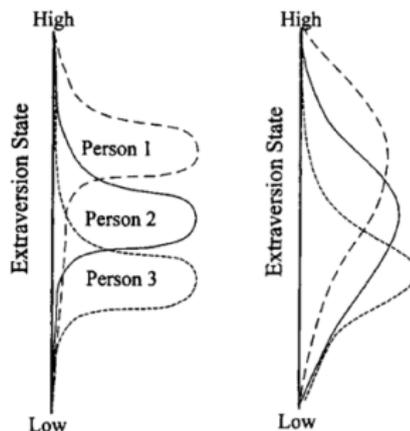
Stapleton et al (2016) identificano quattro tipi di costrutto multilivello in base alle assunzioni sulla relazione tra livello 1 e livello 2, ciascuno con delle restrizioni specifiche. In particolare, per supportare la concettualizzazione di un “*configural cluster construct*” (i.e, lv2 = forma aggregata del lv1) è necessario che ci sia un’**invarianza debole cross-livello**.



Tratti e stati di personalità

Ad esempio, alcune recenti teorie in psicologia della personalità assumono che i **tratti di personalità** non siano ‘valori fissi’ ma piuttosto delle distribuzioni di **stati di personalità**, che si assumono essere causate e riflesse dello stesso dominio di contenuto delle rispettive componenti di tratto.

Questo però richiederebbe che sia verificata l'**invarianza debole cross-livello**.

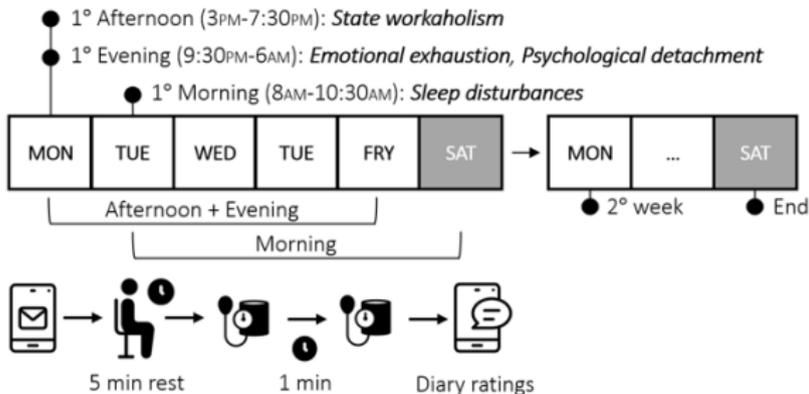


Esempio con dati reali: *State workaholism*

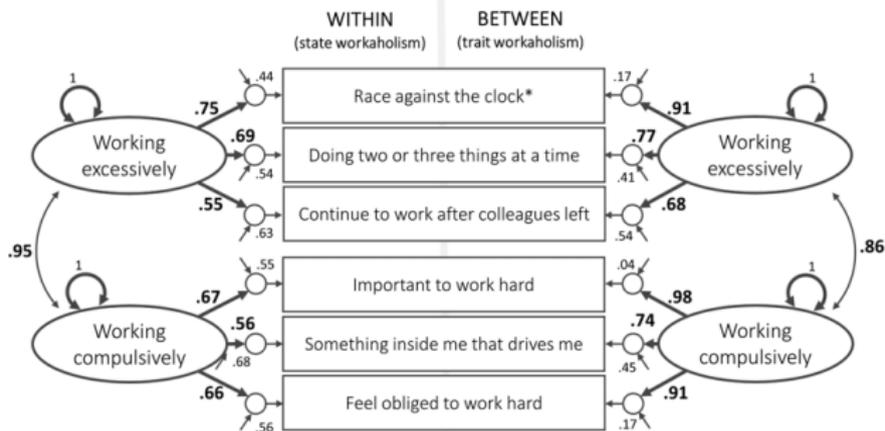
Menghini & Balducci (*Under review*)

https://osf.io/awbxj/?view_only=99eb5bd9b91b4ab88fbc887d9463bef

- *Campione*: 135 lavoratrici/tori d'ufficio
- *Procedura*: 3 questionari al giorno (+ pressione) per 10 giorni lavorativi



Tratti e stati di workaholism



Invarianza cross-livello

```
conf <- # invarianza configurale (stessa struttura)
'level: 1
sWE =~ WHLSM1 + WHLSM3 + WHLSM5
sWC =~ WHLSM2 + WHLSM4 + WHLSM6
level: 2
tWE =~ WHLSM1 + WHLSM3 + WHLSM5
tWC =~ WHLSM2 + WHLSM4 + WHLSM6'

weak <- # invarianza 'debole' (stessa struttura e saturazioni equivalenti)
'level: 1
sWE =~ a*WHLSM1 + b*WHLSM3 + c*WHLSM5
sWC =~ d*WHLSM2 + e*WHLSM4 + f*WHLSM6
level: 2
tWE =~ a*WHLSM1 + b*WHLSM3 + c*WHLSM5
tWC =~ d*WHLSM2 + e*WHLSM4 + f*WHLSM6'
```

Invarianza parziale? 😐

```
# modification indices
```

```
mi <- modificationindices(fit.weak)
```

```
mi[order(mi$mi,decreasing=TRUE),][1:3,]
```

```
##   lhs op rhs block group level   mi   epc sepc.lv sepc.all sepc.nox
## 43  p4 -- p5     2     1     2 6.183 -0.030 -0.030 -0.624 -0.624
## 42  p3 -- p5     2     1     2 4.442  0.035  0.035  1.128  1.128
## 38  p2 -- p3     2     1     2 3.859 -0.046 -0.046 -4.482 -4.482
```

```
weak.part <- # invarianza 'debole' parziale (lascio 'libera' una saturazione)
```

```
'level: 1
```

```
sWE =~ WHLSM1 + b*WHLSM3 + c*WHLSM5
```

```
sWC =~ d*WHLSM2 + e*WHLSM4 + f*WHLSM6
```

```
level: 2
```

```
tWE =~ WHLSM1 + b*WHLSM3 + c*WHLSM5
```

```
tWC =~ d*WHLSM2 + e*WHLSM4 + f*WHLSM6'
```

```
fit.weak.part <- cfa(weak.part, dat, cluster = "ID", estimator = "MLR")
```

Confronto tra modelli

	rmsea	cfi	srmrW	srmrB	BICw
1F conf	.063	.962	.029	.052	0
1F metr	.062	.953	.031	.067	.005
2F conf	.054	.975	.027	.042	.004
2F metr	.056	.967	.029	.076	.205
2F partMetr	.053	.972	.027	.056	.786



Costrutti multilivello e omologia cross-livello

Ma non è finita qui! Una volta attestata l'**invarianza cross-livello** (similarità della misurazione tra livelli) su diversi campioni, possiamo fare un'altro step per valutare quanto il costrutto a livello 1 "funziona in modo simile" a quello a livello 2 valutandone l'**omologia cross-livello** (similarità del network nomologico tra livelli) (Chen et al, 2005)

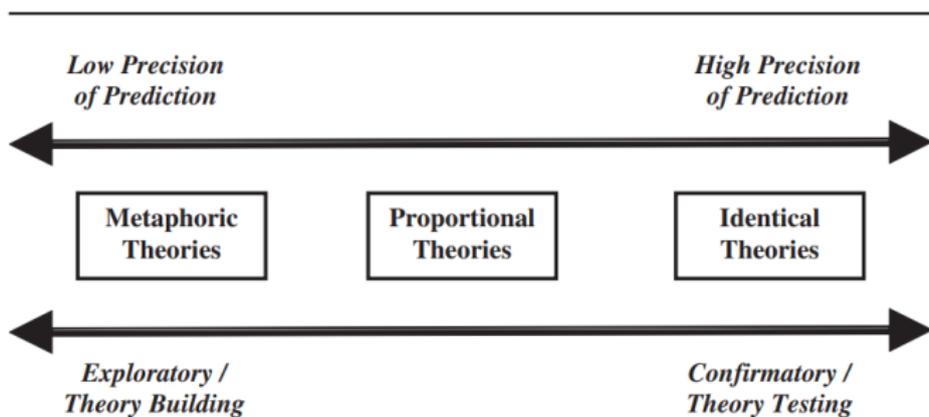


Figure 1: Differentiation of Homology Theories Along Two Theoretical Dimensions