Le tecniche di simulazione Monte Carlo.*

Massimiliano Pastore
Università di Padova

I metodi Monte Carlo riguardano l'uso delle tecniche di campionamento casuale e della simulazione al computer per ottenere una soluzione approssimata ad un problema matematico o fisico (Merriam-Webster, Inc., 1994, pp. 754-755). In altri termini, li possiamo considerare una famiglia di metodi di simulazione tramite i quali è possibile riprodurre e studiare dei sistemi empirici in forma controllata. Si tratta, in pratica, di processi tramite i quali vengono generati dati sulla base di un modello specificato a priori e possiamo dire che ormai sono diventati parte integrante delle tecniche standard utilizzate nei metodi statistici (Robert e Casella, 2004).
Alla base dei metodi Monte Carlo vi è la generazione di variabili casuali. A ciascuna variabile da generare lo sperimentatore assegna delle proprietà relative alle caratteristiche distribuzionali (es. media, varianza, grado di asimmetria), alla loro relazione reciproca (es. correlazione), alla quantità di errore presente nella loro misurazione. Tali proprietà definiranno pertanto la popolazione di riferimento dalla quale estrarre a caso un certo numero di campioni con dimensione definita (che può essere fissa oppure variabile in funzione delle esigenze di studio) sui quali verranno calcolate le statistiche oggetto di studio. Al termine della simulazione lo sperimentatore potrà disporre delle distribuzioni campionarie delle statistiche e da queste dedurre il comportamento in funzione delle proprietà definite nella popolazione.
Le simulazioni Monte Carlo sono molto usate nella letteratura legata alle equazioni strutturali (Paxton, Curran, Bollen, Kirby, Chen, 2001; Fan e Fan, 2005) e permettono di studiare varie tipologie di problemi ad esse connessi; ad esempio il problema dei metodi di stima dei parametri (vedi ad es. Flora e Curran, 2004; Ximénez, 2006) con gli annessi problemi legati alla convergenza dell'algoritmo di stima (Boomsma, 1985), la valutazione dell'adattamento e degli indici di fit (Gerbing e Anderson, 1993; Hu e Bentler, 1999; Sivo, Fan, Witta e Willse, 2006), la prestazione delle procedure di stima e degli indici di adattamento in condizioni non ottimali (ad esempio errata specificazione del modello o distribuzioni non normali nei dati; Fan, Thompson, Wang, 1999; Fan e Sivo, 2005; Fan e Sivo, 2007; Hu e Bentler, 1998; Olsson, Foss, Troye, Howell, 2000), problemi connessi con la presenza di dati mancanti (Arbuckle, 1996; Davey, Savla e Luo, 2005; Enders e Bandalos, 2001; Muthén, Kaplan e Hollis, 1987; Schafer e Graham, 2002), la stima dell'affidabilità (Green e Yang, 2009; Yang e Green, 2010).
In questo capitolo vogliamo introdurre i concetti alla base delle simulazioni Monte Carlo a partire dalla simulazione di un semplice modello di regressione lineare fino alla generazione di insiemi di variabili sulla base di un modello strutturale. Da ultimo presenteremo un esempio

*in C. Barbaranelli & S. Ingoglia (Eds.), *I modelli di equazioni strutturali: temi e prospettive*, pp. 295-324. LED, Milano.

molto semplice di come impostare un esperimento Monte Carlo. I vari algoritmi saranno presentati in dettaglio facendo riferimento all'ambiente R (R Development Core Team, 2010). R è un ambiente statistico *open source* che negli ultimi anni si sta affermando in maniera consistente (Fox, 2009; Muenchen, 2010) al punto da costituire un vero e proprio punto di riferimento per l'analisi statistica e non solo. Dato che non è obiettivo di questo volume la presentazione di R, gli interessati a maggiori approfondimenti possono fare riferimento alla vasta letteratura disponibile in proposito (si veda, ad esempio Dalgaard, 2002; Iacus e Masarotto, 2003; Rizzo, 2008).

1 Introduzione alle simulazioni

1.1 Generazione di numeri casuali

Per poter lavorare con i metodi Monte Carlo è fondamentale disporre di un buon generatore di variabili casuali. Dato che i programmi (intesi nel senso di software) disponibili a tale scopo si basano su algoritmi di tipo deterministico si parla di numeri pseudocasuali piuttosto che casuali in senso stretto. Nella pratica infatti, un generatore produce una sequenza di numeri apparentemente casuali, ma che, da un certo punto in poi, tenderà a ripetersi. Un buon metodo generatore sarà pertanto quello che produce la sequenza di valori più lunga possibile prima di ripetersi.
In ambiente R, ad esempio, il metodo utilizzato per default è il Mersenne-Twister (Matsumoto e Nishimura, 1998), che garantisce una sequenza di $2^{19937} - 1$ elementi prima di ripetersi. In R disponiamo di una serie di comandi (o funzioni) che permettono la generazione di variabili casuali secondo le principali distribuzioni di probabilità, ad esempio `runif()` genera valori con distribuzione uniforme, oppure `rnorm()` genera valori secondo una distribuzione normale.

```
> # distribuzione uniforme
> runif(10)
 [1] 0.4048294744 0.4354041880 0.5463090008 0.5214953858 0.2565510822
 [6] 0.5785783038 0.3139559799 0.1370091424 0.8136617769 0.0001847860
>
> # distribuzione normale standard
> z <- rnorm(1000)
> mean(z)
[1] 0.04570314
> sd(z)
[1] 0.968164
```

Box 1.1: Esempi di uso della funzione `runif()` per generare 10 numeri con distribuzione uniforme e della funzione `rnorm()` per generare 1000 numeri con distribuzione normale standard.

Nel Box 1.1 vediamo due esempi relativi all'utilizzo di queste funzioni. Nel primo caso, il comando `runif(10)` genera 10 numeri casuali, riportati nel box di seguito al comando stesso, campionandoli da una distribuzione uniforme di valori compresi tra 0 e 1. Nel secondo caso, il comando `rnorm(1000)` produce 1000 numeri campionandoli da una distribuzione normale standard. Con la scrittura `z <- rnorm(1000)` i 1000 numeri prodotti vengono assegnati ad

un vettore di nome `z`. Calcolando la media (`mean(z)`) e la deviazione standard (`sd(z)`) di questo insieme di numeri verifichiamo che tali statistiche siano più o meno simili a quelle vere di una normale standard (ossia media 0 e deviazione standard 1).

1.2 Simulazione di un modello di regressione

Vogliamo simulare il seguente modello di regressione lineare semplice

$$y = \beta_0 + \beta_1 x + \epsilon$$

In cui x e y sono le variabili osservate, β_0 e β_1 i parametri incogniti del modello ed ϵ l'errore. Nei contesti empirici disponiamo di x e y ed abbiamo come obiettivo la stima dei valori dei parametri β_0, β_1 e σ_ϵ (ossia la variabilità associata all'errore). In una simulazione di tipo Monte Carlo invece, fissiamo i valori dei parametri e, sulla base di questi, produciamo i valori di x e y (per distinguerli dai valori rilevati empiricamente li indicheremo con x^* e y^*). Il problema principale, per ottenere dati verosimili, è la generazione degli errori in modo che rispettino i requisiti ottimali del modello ossia che abbiano distribuzione normale con media zero, formalmente: $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$. Si tratta in pratica di definire opportunamente il valore da assegnare al parametro σ_ϵ.
Consideriamo la situazione più semplice: generiamo x^* e y^* come variabili standardizzate con correlazione, ad esempio, $\rho = 0.6$. Se le variabili sono standardizzate vale la relazione $\beta_1 = \rho$, di conseguenza anche il coefficiente di regressione sarà $\beta_1 = 0.6$ mentre l'intercetta sarà $\beta_0 = 0$.
Dalla teoria dei modelli di regressione lineare sappiamo che l'errore standard dei residui è

$$\sigma_\epsilon = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{N-2}}$$

in cui $(y_i - \hat{y}_i)$ indica l'i-mo residuo, ossia la differenze tra il valore osservato (y_i) e il valore atteso dal modello di regressione $(\hat{y}_i)$, e N la numerosità campionaria. Sappiamo anche che

$$\sum (y_i - \hat{y}_i)^2 = (1 - r^2) \sum (y_i - \overline{y})^2$$

in cui r è il coefficiente di correlazione tra x e y e $\sum (y_i - \overline{y})^2$ indica la devianza totale di y, ovvero la somma dei quadrati degli scarti tra i valori osservati (y_i) e la loro media $(\overline{y})$. Dal momento che vogliamo generare una variabile standardizzata, la cui varianza è 1, risulterà di conseguenza che la devianza totale di y sarà $(N-1)$. Pertanto avremo:

$$\sigma_\epsilon = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{N-2}} = \sqrt{\frac{(1-r^2)(N-1)}{N-2}} \tag{1}$$

A questo punto il nostro algoritmo sarà così articolato:

1. Determinazione dell'errore standard dei residui (σ_ϵ).
2. Generazione dei valori del predittore x^* con distribuzione normale standardizzata.
3. Generazione dei valori della variabile dipendente y^* utilizzando la relazione $y^* = \beta_0 + \beta_1 x^* + \epsilon$.

```
> N <- 1000; r <- .6
> se <- sqrt((1-r^2)*(N-1)/(N-2))
> xstar <- rnorm(N)
> ystar <- r*xstar + rnorm(N,sd=se)
```

Box 1.2: Generazione di due variabili casuali normali standardizzate con correlazione 0.6.

Nel Box 1.2 vediamo come implementare l'algoritmo in R: 1) Fissiamo la numerosità del campione da generare $N = 1000$ e la correlazione tra le variabili $r = 0.6$; 2) calcoliamo l'errore standard dei residui (σ_ϵ) sfruttando la relazione (1); 3) utilizziamo la funzione `rnorm()` per generare N valori con distribuzione normale standardizzata (`xstar`); 4) creiamo i valori di y^* (`ystar`) aggiungendo alla parte sistematica del modello ($\beta_0 + \beta_1 x^* = 0 + .6x^*$) la componente di errore con distribuzione normale, media 0 e deviazione standard pari a σ_ϵ.
Per verificare la corretta generazione dei valori utilizziamo la funzione `lm()` che stima i parametri di un modello di regressione lineare (vedi Box 1.3). Si può osservare che le stime dei parametri β_0 e β_1 corrispondono circa ai valori che avevamo fissato per il modello ossia 0 e 0.6.

```
> lm(ystar~xstar)

Call:
lm(formula = ystar ~ xstar)

Coefficients:
(Intercept)        xstar
    0.02112      0.63102
```

Box 1.3: Stima dei parametri del modello lineare simulato con l'algoritmo nel Box 1.2.

1.3 Generazione di dati con una determinata struttura di covarianza

Consideriamo ora una situazione multivariata in cui vogliamo generare un insieme di variabili normali con vettore di medie $\boldsymbol{\mu}$ e la cui struttura di covarianza sia definita dalla matrice $\boldsymbol{\Sigma}$. In questo caso avremo bisogno di un algoritmo più complesso in quanto le condizioni da tenere sotto controllo risultano essere maggiori rispetto al modello bivariato visto precedentemente. Un metodo utilizzabile, chiamato *Spectral decomposition*, si basa sulla scomposizione della matrice $\boldsymbol{\Sigma}$ nel seguente modo

$$\boldsymbol{\Sigma}^{1/2} = \mathbf{V}\boldsymbol{\Delta}^{1/2}\mathbf{V}'$$

in cui $\boldsymbol{\Delta}$ è una matrice diagonale contenente gli autovalori di $\boldsymbol{\Sigma}$ e $\mathbf{V}$ la matrice composta dai relativi autovettori[1].
La funzione `eigen()` consente di calcolare autovalori e autovettori della matrice $\boldsymbol{\Sigma}$. Nel Box 1.4 vediamo come procedere: 1) Definiamo la numerosità (`N`), la matrice di covarianza `sigma` (in questo caso standardizzata, ma potrebbe anche non esserlo) ed il vettore di medie `mi` (per

[1]Data $\mathbf{A}$, matrice quadrata di ordine n, se esistono un vettore $\mathbf{v} \neq 0$ ed uno scalare δ tali che $\mathbf{Av} = \delta\mathbf{v}$ allora δ e $\mathbf{v}$ sono rispettivamente autovalore e autovettore di $\mathbf{A}$.

```
> N <- 10000
> (sigma <- matrix(c(1,.7,.3,.7,1,.2,.3,.2,1),3,3))
     [,1] [,2] [,3]
[1,]  1.0  0.7  0.3
[2,]  0.7  1.0  0.2
[3,]  0.3  0.2  1.0
> mi <- rep(0,3)
> ev <- eigen(sigma,symmetric=TRUE)
> delta <- ev$values
> V <- ev$vectors
> R <- V %*% diag(sqrt(delta)) %*% t(V)
> Z <- matrix(rnorm(N*length(mi)),N,length(mi))
> X <- Z%*%R + matrix(mi,N,length(mi),byrow=TRUE)
> round(cor(X),1)
     [,1] [,2] [,3]
[1,]  1.0  0.7  0.3
[2,]  0.7  1.0  0.2
[3,]  0.3  0.2  1.0
```

Box 1.4: Generazione di tre variabili con una fissata struttura di covarianza $\mathbf{\Sigma}$.

semplicità tutte uguali a 0: $\boldsymbol{\mu} = (0, 0, 0)$); 2) Calcoliamo autovalori ed autovettori di `sigma`, con la funzione `eigen()`; 3) Creiamo una matrice `R` utilizzando gli elementi ottenuti dalla decomposizione (`delta` e `V`); 4) Moltiplichiamo quest'ultima matrice per una matrice `Z` di valori casuali, ottenuti con `rnorm()`.
Al termine della procedura, la matrice `X` si compone di N righe (le unità statistiche generate, 10000 nel nostro caso) e q colonne (le variabili, 3 nel nostro caso) le cui medie sono approssimativamente pari a zero, ed ha una matrice di correlazione identica a quella voluta (`sigma`). Con lo stesso algoritmo possiamo generare anche variabili con media diversa da zero: è sufficiente modificare il vettore di medie `mi` definendo tre valori che indichino le medie desiderate. Ad esempio, per avere medie 2, 5 e 8, basta scrivere `mi <- c(2,5,8)`.

2 Simulazioni di modelli di misura

Un modello di misura per le variabili esogene x, secondo la notazione di Jöreskog-Keesling-Wiley (JKW), è definito dall'equazione

$$\mathbf{X} \quad = \quad \mathbf{\Lambda}\boldsymbol{\xi} + \boldsymbol{\delta} \tag{2}$$

in cui $\mathbf{\Lambda}$ è la matrice ($q \times h$, in cui q indica il numero di variabili osservate e h il numero di variabili latenti) dei pesi fattoriali, $\boldsymbol{\xi}$ il vettore ($h \times 1$) dei fattori latenti, $\boldsymbol{\delta}$ il vettore ($q \times 1$) degli errori. Per definire completamente il modello si devono considerare anche la matrice di covarianza ($h \times h$) tra i fattori $\mathbf{\Phi}$ e la matrice di covarianza ($q \times q$) tra gli errori $\mathbf{\Theta}_\delta$.
Il modello di misura definisce la relazione tra le variabili osservate x e i fattori latenti ξ tramite la matrice dei parametri $\mathbf{\Lambda}$ i cui elementi λ_{ij} sono coefficienti di regressione parziali.
Dati i seguenti assunti (Jöreskog & Sörbom, 1996a):

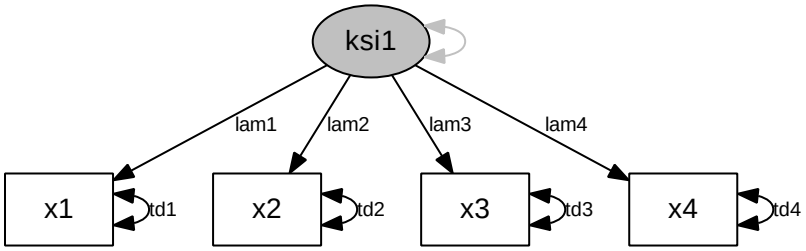

Figura 1: Modello fattoriale con 4 variabili osservate (x) ed una variabile latente (ξ). La freccia grigia indica il parametro del modello fissato ad uno.

1. ξ e δ sono variabili casuali con media zero
2. le variabili ξ non sono correlate con le variabili δ
3. le variabili osservate x sono misurate come scarti dalle proprie medie

si ottiene la seguente relazione con la matrice di covarianza tra le x

$$\mathbf{\Sigma} \quad = \quad \mathbf{\Lambda \Phi \Lambda}^{'} + \mathbf{\Theta}_{\delta} \tag{3}$$

In pratica quindi, per simulare un modello di questo tipo, dobbiamo definirne i parametri, calcolare la matrice $\mathbf{\Sigma}$ e, sulla base di quest'ultima, generare i dati sfruttando quanto illustrato nella sezione precedente.

2.1 Modello a fattore singolo

Supponiamo di voler simulare un modello del tipo rappresentato in figura 1. Si tratta di un modello in cui ipotizziamo l'esistenza di un singolo fattore (ξ) rappresentato da quattro variabili osservate (x_i; $i = 1, ..., 4$).
Questo modello si caratterizza con un sistema di 4 equazioni, direttamente derivato dall'equazione (2):

$$\begin{cases} x_1 = \lambda_1 \xi_1 + \delta_1 \\ x_2 = \lambda_2 \xi_1 + \delta_2 \\ x_3 = \lambda_3 \xi_1 + \delta_3 \\ x_4 = \lambda_4 \xi_1 + \delta_4 \end{cases} \tag{4}$$

Prima di iniziare dobbiamo definire i valori dei parametri del modello, ossia i λ_i e i θ_{δ_i}; per semplicità definiremo tali parametri come costanti nelle varie equazioni. Ciascuna equazione del sistema (4) avrà un suo valore di R^2 (*Squared Multiple Correlations*; Bollen, 1989) definito come

$$R^2_{x_i} = 1 - \frac{\theta_{\delta_i}}{\hat{\sigma}^2_i} \tag{5}$$

in cui θ_{δ_i} indica la varianza dell'errore δ_i e $\hat{\sigma}^2_i$ indica la varianza stimata della variabile osservata x_i. In pratica $R^2_{x_i}$ indica la proporzione di varianza di x_i spiegata dalla relazione lineare con ξ.

Vogliamo generare le quattro variabili x_1, x_2, x_3 e x_4 in modo che siano normali, standardizzate e che abbiano dei valori associati di $R^2_{x_i}$ di circa 0.85. Utilizzando la (5) otteniamo il valore da assegnare ai θ_δ ossia 0.15 (dato che $\hat{\sigma}^2_i = 1$, $\forall i \in \{1, 2, 3, 4\}$).
Per definire i valori dei λ consideriamo che esiste la seguente relazione

$$\hat{\sigma}^2_i = \lambda^2_i + \theta_{\delta_i} \tag{6}$$

in cui λ_i indica la relazione diretta tra ξ e x_i. Posto che $\hat{\sigma}^2_i = 1$ e $\theta_{\delta_i} = 0.15$ avremo $\lambda_i \simeq 0.922$
A questo punto possiamo definire le tre matrici che definiscono il modello ossia:

$$\mathbf{\Lambda} = \begin{bmatrix} 0.922 \\ 0.922 \\ 0.922 \\ 0.922 \end{bmatrix} \quad \mathbf{\Theta}_\delta = \begin{bmatrix} 0.15 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 \\ 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0.15 \end{bmatrix} \quad \mathbf{\Phi} = \begin{bmatrix} 1 \end{bmatrix}$$

Infine possiamo utilizzare uno schema simile a quello utilizzato nella simulazione di un modello di regressione (vedi Box 1.2) definito dall'equazione (2):

1. generiamo i valori della variabile latente ξ con media 0 e deviazione standard 1, utilizzando la funzione `rnorm()`
2. generiamo la matrice degli errori con distribuzione normale e matrice di covarianza $\mathbf{\Theta}_\delta$, utilizzando la funzione `mvrnorm()` disponibile nella libreria `MASS` (Venables e Ripley, 2002)
3. costruiamo la matrice dei valori osservati $\mathbf{X}$ con l'equazione (2)

```
> N <- 1000; nx <- 4; nk <- 1
> td <- .15; lam <- sqrt(1-td)
> LX <- matrix(rep(lam,nx),nx,nk)
> TD <- diag(td,nx)
>
> ksi <- rnorm(N)
> library(MASS)
> delta <- mvrnorm(N,rep(0,nrow(TD)),TD)
> X <- ksi%*%t(LX)+delta
```

Box 2.1: Generazione di variabili osservate in un modello con una variabile latente.

Nel Box 2.1 vediamo un esempio di come procedere. Definiamo la numerosità del campione (`N = 1000`), il numero di variabili x (`nx = 4`) e di variabili ξ (`nk = 1`). Poi fissiamo i valori scelti di θ_δ (`td = 0.15`) e λ (dalla relazione 6) e sulla base di tali valori costruiamo le matrici $\mathbf{\Lambda}$ (`LX`) e $\mathbf{\Theta}_\delta$ (`TD`). Infine generiamo i valori della variabile latente (`ksi`), degli errori (`delta`) e delle x.
Nel Box 2.2 sono riportate le stime ottenute dei parametri[2] in relazione ai dati generati nel Box 2.1. Per la verifica abbiamo utilizzato il pacchetto `sem` (Fox, 2006) disponibile per l'ambiente R. Possiamo vedere che, in generale, i valori dei parametri sono approssimativamente quelli che avevamo imposto.

[2]Dato che esula dagli obiettivi di questo contributo, non riportiamo i dettagli del comando per ottenere tali stime, gli interessati possono vedere Fox 2006.

```
 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
lam1 0.92730  0.0241786 38.352  0        x1 <--- ksi1
lam2 0.92401  0.0244443 37.801  0        x2 <--- ksi1
lam3 0.93726  0.0247408 37.883  0        x3 <--- ksi1
lam4 0.93079  0.0244390 38.086  0        x4 <--- ksi1
td1  0.14229  0.0088477 16.082  0        x1 <--> x1
td2  0.15783  0.0094217 16.752  0        x2 <--> x2
td3  0.15977  0.0095958 16.650  0        x3 <--> x3
td4  0.15139  0.0092169 16.425  0        x4 <--> x4
```

Box 2.2: Stime dei parametri ottenute con la generazione di variabili del Box 2.1.

2.2 Modello a due fattori

Prendiamo ora in considerazione il modello in figura 2. Si tratta di un modello in cui ipotizziamo l'esistenza di due variabili latenti (ξ_i; $i = 1, 2$) ciascuna delle quali rispettivamente con quattro variabili osservate (x_i; $i = 1, ..., 8$). Rispetto al modello precedente, qui abbiamo anche un parametro che definisce la correlazione tra le due latenti (ϕ_{21}).
Seguendo lo stesso schema logico del modello a fattore singolo, definiamo le matrici $\mathbf{\Lambda}$ e $\mathbf{\Theta}_\delta$ nel seguente modo:

$$\mathbf{\Lambda} = \begin{bmatrix} 0.922 & 0 \\ 0.922 & 0 \\ 0.922 & 0 \\ 0.922 & 0 \\ 0 & 0.922 \\ 0 & 0.922 \\ 0 & 0.922 \\ 0 & 0.922 \end{bmatrix} \quad \mathbf{\Theta}_\delta = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.15 \end{bmatrix}$$

Definiamo poi la matrice di correlazione tra le variabili latenti fissando a 0.3 il parametro ϕ_{21}:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

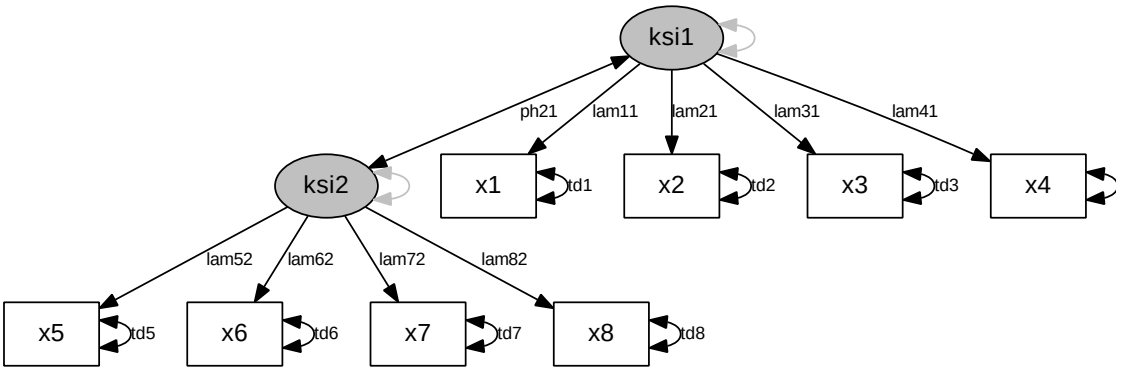

Figura 2: Modello fattoriale con 8 variabili osservate (x) e due variabili latenti (ξ). Le frecce grigie indicano i parametri del modello fissati ad uno.

L'algoritmo per la generazione dei dati rispetto a questo modello sarà sostanzialmente identico a quello già illustrato nel Box 2.1, con una sola differenza: dato che le variabili latenti sono due, con matrice di correlazione $\mathbf{\Phi}$, ci servirà il comando `mvrnorm()` anche per generare tali variabili.
Il Box 2.3 illustra come procedere per generare un campione di `N = 1000` osservazioni di 8 variabili ciascuna (`nx = 8`) e due variabili latenti (`nk = 2`). I valori dei θ_δ sono posti a 0.15, i valori dei λ a circa 0.92. Si creano le matrici dei parametri `LX`, `TD` e `PH` sulla base dei valori definiti precedentemente. Infine, generiamo le variabili latenti (`ksi`), gli errori (`delta`) e le variabili osservate (`X`) utilizzando la (2).

```
> N <- 1000; nx <- 8; nk <- 2
> td <- .15; lam <- sqrt(1-td)
> LX <- matrix(c(rep(lam,4),rep(0,nx),rep(lam,4)),nx,nk)
> TD <- diag(td,nx)
> PH <- matrix(c(1,.3,.3,1),nk,nk)
>
> ksi <- mvrnorm(N,rep(0,nk),PH)
> delta <- mvrnorm(N,rep(0,nx),TD)
> X <- ksi%*%t(LX)+delta
```

Box 2.3: Generazione di variabili osservate in un modello con due variabili latenti con correlazione 0.3.

Un metodo alternativo può essere quello di creare i dati direttamente dalla matrice di covarianza $\mathbf{\Sigma}$, come descritto nel Box 1.4, ottenendo quest'ultima dalla relazione (3).

3 Simulazioni con modelli strutturali completi

Vediamo ora come procedere nel caso di modelli strutturali completi, ovverosia che si compongono di tutti i tipi possibili di variabili endogene ed esogene, osservate e latenti. Nella classica formulazione le variabili osservate esogene sono indicate con la lettera x, le osservate endogene con la y, le latenti esogene con la lettera greca ξ e le latenti endogene con la lettera greca η. Un modello completo, sempre nella formulazione JKW, si compone di tre equazioni:

$$\begin{cases} \mathbf{X} = \mathbf{\Lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta} \\ \mathbf{Y} = \mathbf{\Lambda}_y \boldsymbol{\eta} + \boldsymbol{\epsilon} \\ \boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \mathbf{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta} \end{cases}$$

Le prime due equazioni definiscono i modelli di misura rispettivamente per le x e le y, la terza definisce il modello strutturale.
Dato il nostro obiettivo di generare dati sulla base dei parametri specificati a priori, è utile conoscere il legame tra i parametri e la matrice di covarianza tra le variabili osservate. Tale matrice si definisce come segue (Jöreskog & Sörbom, 1996a):

$$\mathbf{\Sigma}_{yx} = \mathrm{Cov}\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}_y(\mathbf{\Pi\Phi\Pi'} + \mathbf{\Psi}^*)\mathbf{\Lambda}_y' + \mathbf{\Theta}_\epsilon & \\ \mathbf{\Lambda}_x\mathbf{\Phi\Pi'}\mathbf{\Lambda}_y' & \mathbf{\Lambda}_x\mathbf{\Phi}\mathbf{\Lambda}_x' + \mathbf{\Theta}_\delta \end{bmatrix} \tag{7}$$

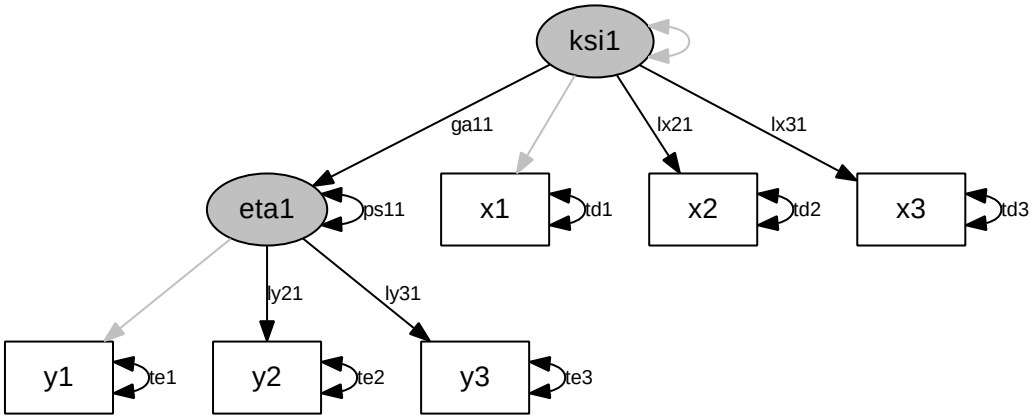

Figura 3: Modello strutturale completo. Le frecce grigie indicano i parametri del modello fissati ad uno.

in cui $\mathbf{\Pi} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Gamma}$ e $\mathbf{\Psi^*} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Psi}(\mathbf{I} - \mathbf{B}')^{-1}$.
Sfruttando la (7) possiamo calcolare la matrice $\mathbf{\Sigma}_{yx}$ in funzione dei parametri definiti del modello e, a partire da questa, generare i dati come descritto nella sezione 1.3.

3.1 Esempio 1

Consideriamo il modello rappresentato in figura 3. Tale modello si compone di sei variabili osservate (tre x e tre y) e due latenti. Definiamo i valori per le matrici di parametri: nella parte relativa ai modelli di misura utilizziamo i criteri già descritti (vedi sezione 2.1), fissando ad uno λ^y_{11}, λ^x_{11} e la varianza di ξ; per la parte struttuale assegniamo 0.3 al parametro γ_{11} e 0.15 al parametro ψ_{11}. Le matrici complete sono illustrate di seguito (8):

$$
\begin{aligned}
\mathbf{\Lambda}_y &= \begin{bmatrix} 1 \\ 0.922 \\ 0.922 \end{bmatrix} \quad \mathbf{\Theta}_\epsilon = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} \\
\mathbf{\Lambda}_x &= \begin{bmatrix} 1 \\ 0.922 \\ 0.922 \end{bmatrix} \quad \mathbf{\Theta}_\delta = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} \quad \mathbf{\Phi} = [1] \\
\mathbf{\Gamma} &= [0.3] \qquad\qquad \mathbf{\Psi} = [0.15]
\end{aligned}
\tag{8}
$$

A partire dalle matrici dei parametri, sfruttando la (7), calcoliamo la matrice di covarianza $\mathbf{\Sigma}_{yx}$. Successivamente, utilizzando la procedura descritta nella sezione 1.3 generiamo i dati.
Nel Box 3.1 è riportato il codice che permette di generare una matrice di 1000 osservazioni e 6 variabili. Esso si compone di tre parti: 1. Definizione dei parametri del modello e delle relative matrici sulla base dei valori prestabiliti; 2. Calcolo della matrice di covarianza tra le variabili osservate $\mathbf{\Sigma}_{yx}$ in base alla formula (7); 3. Generazione della matrice dei dati X con l'algoritmo descritto nel Box 1.4.

```
> # definizione dei parametri
> N <- 1000; nx <- 3; nk <- 1; ny <- 3; ne <- 1
> td <- .15; lx <- sqrt(1-td)
> te <- .15; ly <- sqrt(1-te)
> LY <- matrix(c(1,ly,ly),ny,ne)
> TE <- diag(td,ny)
> LX <- matrix(c(1,lx,lx),nx,nk)
> TD <- diag(td,nx)
> PH <- matrix(1,nk,nk)
> GA <- matrix(rep(.3,nk),ne,nk)
> PS <- diag(.15,ne)
> BE <- matrix(rep(0,ne),ne,ne)
>
> # calcolo di Sigmayx
> ID <- diag(1,nrow(BE))
> PI <- solve(ID-BE)%*%GA
> Syx <- LX%*%PH%*%t(PI)%*%t(LY)
> PS <- solve(ID-BE)%*%PS%*%solve(ID-t(BE))
> Syy <- LY%*%(PI%*%PH%*%t(PI)+PS)%*%t(LY)+TE
> Sxx <- LX%*%PH%*%t(LX)+TD
> Sigmayx <- cbind(rbind(Syy,Syx),rbind(t(Syx),Sxx))
>
> # generazione dati
> mi <- rep(0,nrow(Sigmayx))
> ev <- eigen(Sigmayx,symmetric=TRUE)
> delta <- ev$values
> V <- ev$vectors
> R <- V %*% diag(sqrt(delta)) %*% t(V)
> Z <- matrix(rnorm(N*length(mi)),N,length(mi))
> X <- Z%*%R + matrix(mi,N,length(mi),byrow=TRUE)
```

Box 3.1: Generazione di variabili osservate relative al modello in figura 3. Nella prima sezione vengono definite le matrici dei parametri: $\mathbf{\Lambda}_y$ (LY), $\mathbf{\Theta}_\epsilon$ (TE), $\mathbf{\Lambda}_x$ (LX), $\mathbf{\Theta}_\delta$ (TD), $\mathbf{\Phi}$ (PH), $\mathbf{\Gamma}$ (GA), $\mathbf{\Psi}$ (PS) sulla base dei valori illustrati nella (8). In più si definisce anche la matrice $\mathbf{B}$ (BE) che in questo modello è una matrice di zeri. Nella seconda sezione si calcola la matrice di covarianza $\mathbf{\Sigma}_{yx}$ (Sigmayx) sfruttando la relazione (7). Infine, nella terza sezione, si utilizza lo schema già presentato nel Box 1.4 per generare una matrice di dati X costituita da 1000 righe (le unità statistiche) e 9 colonne (le variabili osservate del modello).

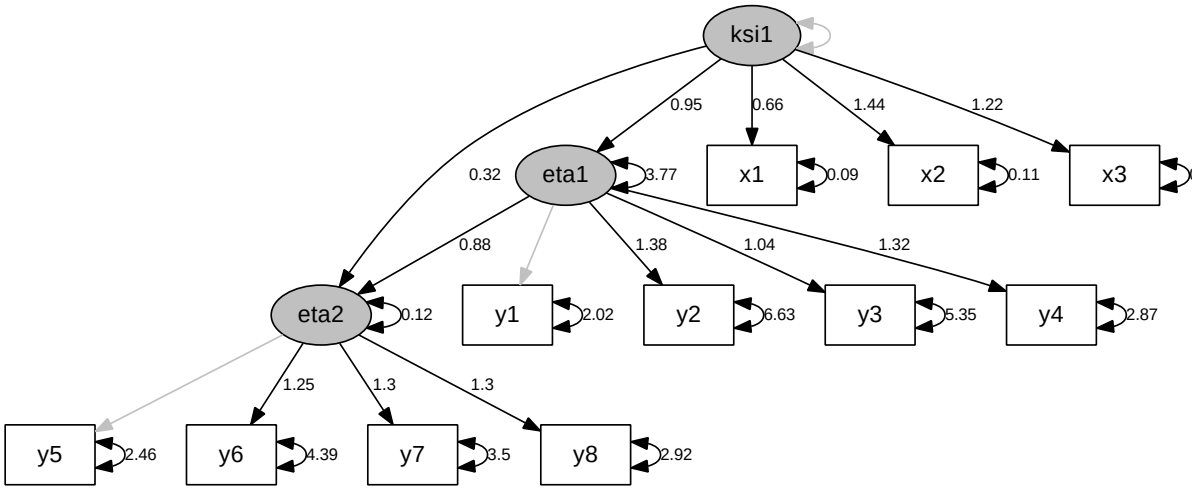

Figura 4: Modello strutturale completo. I valori riportati indicano le stime dei parametri ottenuti in un campione empirico. Le frecce grigie indicano i parametri del modello fissati ad uno.

3.2 Esempio 2

Consideriamo ora un esempio di riproduzione dei dati a partire da un modello empirico, ovverosia, piuttosto che decidere a priori i valori da assegnare ai parametri, utilizziamo quelli ottenuti sulla base di un campione osservato. Supponiamo di avere un modello come quello rappresentato in figura 4 con relativi parametri stimati e di voler riprodurre un campione di dati sulla base del modello in esame.
Le matrici dei parametri sono le seguenti:

$$\boldsymbol{\Lambda}_y = \begin{bmatrix} 1 & 0 \\ 1.38 & 0 \\ 1.04 & 0 \\ 1.32 & 0 \\ 0 & 1 \\ 0 & 1.25 \\ 0 & 1.30 \\ 0 & 1.30 \end{bmatrix} \quad \boldsymbol{\Theta}_\epsilon = \begin{bmatrix} 2.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.63 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.35 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.87 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.46 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.39 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.92 \end{bmatrix}$$

$$\boldsymbol{\Lambda}_x = \begin{bmatrix} 0.66 \\ 1.44 \\ 1.22 \end{bmatrix} \quad \boldsymbol{\Theta}_\delta = \begin{bmatrix} 0.09 & 0 & 0 \\ 0 & 0.11 & 0 \\ 0 & 0 & 0.49 \end{bmatrix} \quad \boldsymbol{\Phi} = [1]$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.95 \\ 0.32 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0.88 & 0 \end{bmatrix} \quad \boldsymbol{\Psi} = \begin{bmatrix} 3.78 & 0 \\ 0 & 0.12 \end{bmatrix}$$

A partire da tali parametri, sfruttando la 7, possiamo calcolare la matrice di covarianza attesa tra le variabili osservate:

$$\boldsymbol{\Sigma}_{yx} = \begin{bmatrix} 6.70 & 6.51 & 4.87 & 6.18 & 4.42 & 5.53 & 5.75 & 5.75 & 0.63 & 1.37 & 1.16 \\ 6.51 & 15.68 & 6.77 & 8.59 & 6.15 & 7.69 & 8.00 & 8.00 & 0.87 & 1.90 & 1.61 \\ 4.87 & 6.77 & 10.41 & 6.43 & 4.60 & 5.75 & 5.98 & 5.98 & 0.65 & 1.42 & 1.21 \\ 6.18 & 8.59 & 6.43 & 11.03 & 5.84 & 7.30 & 7.59 & 7.59 & 0.83 & 1.81 & 1.53 \\ 4.42 & 6.15 & 4.60 & 5.84 & 6.84 & 5.48 & 5.70 & 5.70 & 0.76 & 1.66 & 1.41 \\ 5.53 & 7.69 & 5.75 & 7.30 & 5.48 & 11.24 & 7.12 & 7.12 & 0.95 & 2.08 & 1.76 \\ 5.75 & 8.00 & 5.98 & 7.59 & 5.70 & 7.12 & 10.91 & 7.41 & 0.99 & 2.16 & 1.83 \\ 5.75 & 8.00 & 5.98 & 7.59 & 5.70 & 7.12 & 7.41 & 10.33 & 0.99 & 2.16 & 1.83 \\ 0.63 & 0.87 & 0.65 & 0.83 & 0.76 & 0.95 & 0.99 & 0.99 & 0.53 & 0.95 & 0.81 \\ 1.37 & 1.90 & 1.42 & 1.81 & 1.66 & 2.08 & 2.16 & 2.16 & 0.95 & 2.18 & 1.76 \\ 1.16 & 1.61 & 1.21 & 1.53 & 1.41 & 1.76 & 1.83 & 1.83 & 0.81 & 1.76 & 1.98 \end{bmatrix}$$

Una volta calcolata questa matrice, possiamo generare i dati utilizzando l'algoritmo già descritto nel Box 1.4.

4 Simulazioni con dati discreti

Nei casi presi in esame fino ad ora abbiamo considerato dei modelli di simulazione che producono come risultato delle variabili continue. Sappiamo però che spesso le variabili utilizzate come input per modelli strutturali nelle discipline psicologiche e sociali hanno piuttosto una natura discreta. Si pensi ad esempio alle scale di tipo Likert, in cui i soggetti interpellati rispondono ad una serie di item indicando il loro grado di accordo su una scala con un numero limitato di possibilità di scelta.
In linea generale possiamo assumere che una variabile discreta x (espressa con k categorie ordinate) rappresenti una approssimazione di una variabile latente continua ξ con distribuzione normale e media 0 (Jöreskog e Sörbom, 1996b, p. 4). Pertanto, quando osserviamo $x = i$ intendiamo che il vero valore corrispondente ξ risulta compreso tra due estremi ossia $\alpha_{i-1} < \xi \leq \alpha_i$, in cui $\alpha_0 = -\infty$, $\alpha_1 < \alpha_2 < ... < \alpha_{k-1}$ e $\alpha_k = +\infty$ sono i parametri soglia. Come conseguenza di questa associazione avremo che, data una variabile discreta (ordinata) con k valori possibili, esistono $k - 1$ soglie incognite.
Se vogliamo simulare dei dati che riproducano questo tipo di struttura ci troviamo di fronte quindi ad un livello maggiore di complessità. In particolare, la struttura discreta dei dati rende più problematica la gestione delle correlazioni tra le variabili da generare. In primo luogo, non possiamo più ragionare in termini di covarianza; questo concetto infatti risulta strettamente legato alla continuità delle variabili. Di conseguenza dobbiamo fare riferimento ad un altro tipo di correlazione, che tenga conto della natura discreta dei dati. Pertanto la struttura base di riferimento non sarà più la matrice di covarianza ma la tabella delle frequenze congiunte. Su tale tabella possiamo utilizzare la correlazione policorica (Drasgow, 1986) definita come una stima della correlazione tra le variabili latenti di cui le variabili ordinali sono approssimazione.

4.1 Generazione di due variabili discrete con correlazione definita

Supponiamo di voler generare due variabili discrete x_1 e x_2 che assumono rispettivamente k_1 e k_2 categorie di valori possibili ordinati. Possiamo assumere che tali variabili siano una

espressione approssimata di due variabili latenti corrispondenti ξ_1 e ξ_2 continue e normalmente distribuite.
Il primo passaggio per generare variabili discrete consiste quindi nel definire le soglie relative alle due variabili, siano $\alpha_1, \alpha_2, ..., \alpha_{k_1-1}$ per x_1 e $\beta_1, \beta_2, ..., \beta_{k_2-1}$ per x_2, e la correlazione voluta, sia ρ.
Successivamente si calcolano le probabilità congiunte $\pi_{ij} = Pr(x_1 = i, x_2 = j)$ definite come

$$\pi_{ij} = \int_{\alpha_i-1}^{\alpha_i} \int_{\beta_j-1}^{\beta_j} \phi(u, v)\, du\, dv \tag{9}$$

in cui ϕ è la densità della normale bivariata con correlazione ρ.
Una volta ottenute le probabilità congiunte π_{ij} si utilizzano tali probabilità per produrre i valori delle variabili x_1 e x_2.
Nel Box 4.1 abbiamo riportato un esempio di come procedere per generare due variabili discrete su una scala a quattro valori e con correlazione pari a $\rho = 0.6$.
L'algoritmo si compone di 4 passaggi:

1. Definizione dei parametri: `N`, numerosità campionaria; `tsholds`, valori delle soglie, definiti arbitrariamente; `R`, matrice di correlazione tra le variabili.

2. Calcolo delle probabilità π_{ij}, inserite in una matrice `P` di dimensione 4×4. Per il calcolo si utilizza la funzione `pmvnorm()` (disponibile nella libreria `mvtnorm`; Genz, Bretz, Miwa, Mi, Leisch, Scheipl e Hothorn, 2011) che calcola l'integrale relativo alla normale bivariata in base alle soglie ed alla correlazione definite.

3. Generazione della tabella di frequenze bivariata moltiplicando la numerosità voluta `N` per la tabella delle probabilità congiunte `P`.

4. Verifica della correlazione policorica ottenuta. Per ottenere tale coefficiente utilizziamo la funzione `polychor()` della libreria `polycor` (Fox, 2010).

Come si vede la matrice di frequenze congiunte ottenuta (`Fn`) è perfettamente simmetrica. Calcolando le frequenze marginali di riga e di colonna di tale tabella si ottengono due distribuzioni uguali (101, 201, 400, 300), dato che le soglie utilizzate sono le stesse per entrambe le variabili. È evidente che, con la stessa logica, possiamo produrre anche variabili con diverse soglie, o con un numero diverso di valori possibili (es. variabili dicotomiche) e, di conseguenza, con diverse distribuzioni di frequenza marginali.

4.2 Modello a fattore singolo

Vogliamo simulare il modello già presentato nella sezione 2.1 (vedi figura 1) in modo da ottenere però delle variabili discrete su una scala a 4 valori. In pratica si tratterà di generare prima delle variabili continue e poi renderle discrete con una opportuna trasformazione.
Di conseguenza riprendiamo quanto già illustrato precedentemente, a partire dalle matrici dei parametri definite nella sezione 2.1 e dall'algoritmo descritto nel Box 2.1 il cui risultato finale è una matrice `X` costituita da 4 variabili continue. A questo punto, dobbiamo definire opportunamente le soglie per ricodificare i valori da continui a discreti.

```
> ## 1) definizione dei parametri
> N <- 1000
> tsholds <- c(-Inf,c(-1.282,-0.524,0.524),Inf)
> R <- matrix(c(1,0.6,0.6,1),2,2)
>
> ## 2) calcolo le probabilita'
> P <- matrix(NA,4,4)
> for (i in 1:4) {
+     for (j in 1:4) {
+         P[i,j] <- pmvnorm(c(tsholds[i],tsholds[j]),
+             c(tsholds[i+1],tsholds[j+1]),corr=R)
+     }
+ }
>
> ## 3) generazione della tabella di frequenze
> (Fn <- round(P*N))
     [,1] [,2] [,3] [,4]
[1,]   39   35   24    3
[2,]   35   65   81   20
[3,]   24   81  191  104
[4,]    3   20  104  173
>
> ## 4) calcolo della correlazione
> library(polycor)
> polychor(Fn)
[1] 0.5988229
```

Box 4.1: Generazione di due variabili ordinali con quattro categorie di valori e correlazione $\rho = 0.6$. Nella sezione 1 si definiscono nell'ordine: la numerosità campionaria (`N`), le soglie (`tsholds`) e la matrice di correlazione (`R`). Nella sezione 2 vengono calcolate le probabilità della distribuzione congiunta con la (9). Nella sezione 3 si produce la tabella di frequenze bivariata. Nella sezione 4 si calcola la correlazione policorica sulla tabella di frequenze per verificare che corrisponda circa al valore desiderato (0.6).

Un criterio indicato da Jöreskog e Sörbom (1996b, p. 6) per la stima delle soglie si basa sulla distribuzione marginale empirica di ciascuna variabile nel seguente modo:

$$\alpha_i = \Phi^{-1}\left(\sum_{j=1}^{i}\frac{n_j}{N}\right) \qquad i = 1, 2, ..., k-1 \tag{10}$$

in cui Φ^{-1} è l'inversa della funzione di ripartizione della normale standard, N il numero complessivo di osservazioni della variabile e n_j la frequenza di osservazioni nella classe j-ma. Dato che, nel nostro caso, non disponiamo di una distribuzione empirica delle variabili osservate non possiamo calcolare le frequenze relative n_j/N ma dovremo piuttosto definirle sulla base di un criterio appropriato. Nel caso più semplice possiamo ottenerle dalle probabilità di una distribuzione binomiale con parametro $\pi = 0.5$. La scelta della binomiale può considerarsi un buon criterio in quanto tale distribuzione è approssimazione discreta della normale. Pertanto, volendo ottenere tre soglie, calcoliamo tre probabilità così definite

$$F(x; h, \pi) = Pr(X \leq x) = \sum_{j=1}^{x}\binom{h}{j}\pi^j(1-\pi)^{h-j}$$

con $x = 0, 1, 2$, $h = (k-1) = 3$ e $\pi = 0.5$. I valori che si ottengono sono 0.125, 0.5 e 0.875; sostituendoli nella 10 al posto della distribuzione cumulata empirica avremo le seguenti soglie: $\alpha_1 = -1.15$, $\alpha_2 = 0.00$ e $\alpha_3 = 1.15$. L'ultimo passaggio consisterà nel ricodificare i valori continui ottenuti e riportati nella matrice `X` con il seguente criterio:

- i valori minori o uguali ad α_1 diventano 1
- i valori maggiori di α_1 e minori o uguali ad α_2 diventano 2
- i valori maggiori di α_2 e minori o uguali ad α_3 diventano 3
- i valori maggiori di α_3 diventano 4

```
> ## 1) calcolo delle soglie
> prob <- pbinom(0:2,3,.5)
> tsholds <- qnorm(prob)
>
> ## 2) ciclo di ricodifica
> Dx <- X
> for (j in 1:4) Dx[,j] <- cut(Dx[,j],c(-Inf,tsholds,Inf))
```

Box 4.2: Seguito dell'algoritmo descritto nel Box 2.1 per rendere discreti i valori della matrice `X`.

Il Box 4.2 riporta i passaggi per trasformare la matrice `X` con quattro variabili continue (ottenuta nel Box 2.1) nella matrice `Dx` costituita da quattro variabili discrete. Nella prima parte vengono calcolate, con la funzione `pbinom()`, le probabilità cumulate della distribuzione binomiale, e, con la funzione `qnorm()` che calcola i quantili della distribuzione normale, i valori delle soglie (`tsholds`). Nella seconda parte dell'algoritmo viene applicato il criterio di ricodifica. Al termine della procedura si otterrà una nuova matrice `Dx` con valori discreti compresi tra 1 e 4.

4.3 Caso dicotomico

Un caso particolare di variabili discrete si ha quando i valori possibili sono solo due. Adottando lo schema già descritto nella sezione precedente dovremo definire un unico valore soglia α_1, che nel caso più semplice di totale simmetria avrà valore zero. Pertanto, preso come riferimento l'algoritmo descritto nel Box 2.1, una volta ottenuta la matrice `X` dei dati continui, si può trasformare in una matrice `Dx` di valori dicotomici con il semplice ciclo di ricodifica descritto nel Box 4.3.

```
> ## 2) ciclo di ricodifica
> Dx <- X
> for (j in 1:4) Dx[,j] <- cut(Dx[,j],c(-Inf,0,Inf))
```

Box 4.3: Seguito dell'algoritmo descritto nel Box 2.1 per rendere dicotomici i valori della matrice `X`.

5 Un esempio di esperimento Monte Carlo

In questa ultima sezione vogliamo illustrare un esempio di come eseguire un esperimento Monte Carlo. Immaginiamo di voler studiare il comportamento degli indici di adattamento in un semplice caso di *model misspecification*, ossia l'errata specificazione delle relazioni che intercorrono tra le variabili di un modello e, di conseguenza, dei parametri che lo caratterizzano. In altri termini, si produce errata specificazione quando si omettono parametri significativi oppure si inseriscono parametri non rilevanti creando di fatto un modello errato. Questo problema è stato molto trattato nella letteratura (si veda ad esempio Beauducel e Wittmann, 2005; Fan e Sivo, 2005; Fan e Sivo, 2007; Jackson, 2007; Kaplan, 1989; Lei, 2009; Olsson et. al, 2000; Yuan, Kouros e Kelley, 2008; Yuan, Marshall e Bentler, 2003) e ci offre innumerevoli esempi possibili.
Consideriamo un caso di analisi fattoriale confermativa prendendo come riferimento una versione semplificata di un modello proposto da Fan e Sivo (2007). Il modello scelto è presentato in figura 5 e consiste di 10 item e due variabili latenti. Nel modello vero il quarto item (x_4) presenta una saturazione significativa con entrambe le latenti le quali sono tra loro correlate (ϕ_{21}). Alla base della nostra simulazione poniamo la seguente domanda: cosa accade agli indici di adattamento se nell'analisi del modello omettiamo uno dei due parametri rappresentati in figura con la linea tratteggiata? Inoltre, vogliamo considerare se vi sia un effetto su tali indici legato anche alla dimensione campionaria. In particolare, ci attenderemmo che un indice di adattamento adeguato risulti sensibile all'errata specificazione, producendo dei valori che indicano un peggiore fit nel modello, ma non sia sensibile alla numerosità campionaria.
Per definire i parametri del modello utilizziamo i seguenti valori suggeriti da Fan e Sivo (2007):

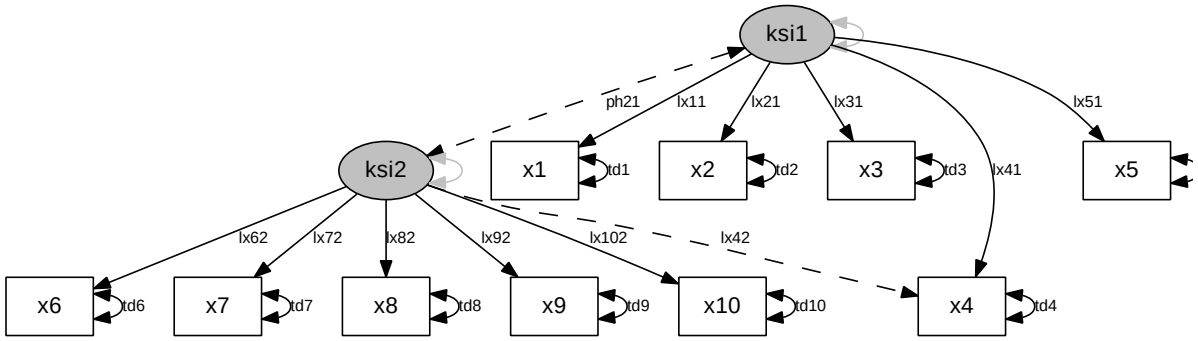

Figura 5: Modello di analisi fattoriale confermativa. Le frecce tratteggiate indicano i parametri soggetti ad errata specificazione.

$$\mathbf{\Lambda}_x = \begin{bmatrix} 0.7 & 0 \\ 0.7 & 0 \\ 0.75 & 0 \\ 0.8 & 0.5 \\ 0.8 & 0 \\ 0 & 0.7 \\ 0 & 0.7 \\ 0 & 0.75 \\ 0 & 0.8 \\ 0 & 0.8 \end{bmatrix} \quad \mathbf{\Theta}_\delta = \begin{bmatrix} 0.51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.36 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.36 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.44 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

Notiamo che la quarta variabile del modello presenta valori diversi da zero in entrambe le colonne della matrice $\mathbf{\Lambda}_x$, inoltre la correlazione tra le due variabili latenti viene posta al valore di 0.5.
Il nostro disegno sperimentale sarà caratterizzato dai due fattori seguenti:

1. il tipo di errata specificazione, con tre livelli (nessuna, omissione del parametro ϕ_{21}, omissione del parametro λ^x_{42})

2. la dimensione campionaria, con tre livelli ($N = 50, 100, 1000$)

Per tutte le 9 combinazioni dei livelli di questi fattori sperimentali calcoleremo i seguenti 8 indici di adattamento: *Goodness-of-fit index* (GFI), *Adjusted goodness-of-fit index* (AGFI), *Normed fit index* (NFI), *Non-normed fit index* (NNFI), *Comparative fit index* (CFI), *Root-mean-square error of approximation* (RMSEA), *Standardized root mean square residual* (SRMR) e *Bayesian information criterion* (BIC).
L'algoritmo che utilizzeremo per l'esperimento sarà articolato come segue:

- Generazione di un set di dati di dimensione N ($N = 50, 100, 1000$) sulla base dei parametri definiti.

	GFI	AGFI	NFI	NNFI	CFI	RMSEA	SRMR	BIC
MI	14	12	14	93	86	90	95	37
N	86	87	86	7	13	5	4	11
$MI * N$	0	0	0	0	1	5	1	52

Tabella 1: $\eta^2 \times 100$ ossia percentuale di variabilità relativa agli indici di adattamento attribuibile ai differenti fattori del disegno sperimentale: MI = Errata specificazione (*MIsspecification*), N = numerosità campionaria.

- Calcolo degli indici di adattamento utilizzando il modello corretto (condizione di nessuna omissione dei parametri).
- Calcolo degli indici di adattamento utilizzando un modello che non prevede la correlazione tra le variabili latenti ξ_1 e ξ_2 (condizione di omissione del parametro ϕ_{21}).
- Calcolo degli indici di adattamento utilizzando un modello che non prevede la relazione tra la variabile latente ξ_2 e la variabile osservata x_4 (condizione di omissione del parametro λ^x_{42}).

Ciascuno step viene replicato 1000 volte per ogni condizione sperimentale; al termine dell'esperimento avremo prodotto $1000 \times 3 \times 3 = 9000$ set di dati e di conseguenza 9000 valori per ogni indice di adattamento considerato.

5.1 Analisi dei risultati

Al fine di valutare quanto ciscun indice risulti influenzato dai fattori sperimentali, possiamo utilizzare η^2. Tale statistica viene calcolata in base alla scomposizione delle devianze ed è definita come $\eta^2_F = \frac{SS_F}{SS_{Totale}}$, in cui SS_F è la devianza attribuibile al fattore F e SS_{Totale} la devianza totale. Pertanto $\eta^2 \times 100$ esprime la percentuale di variabilità campionaria nell'indice attribuibile ad un determinato fattore.
In tabella 1 sono riportati i valori di $\eta^2 \times 100$ ottenuti per ciascun indice dalla nostra simulazione. Dalla lettura della tabella possiamo rilevare che l'errata specificazione del modello ha una non trascurabile influenza su tutti gli indici considerati: si va dal 12-14% relativo a GFI, AGFI e NFI sino al 93-95% di NNFI e SRMR. La numerosità del campione influisce molto, circa per l'86%, su GFI, AGFI e NFI, molto meno sugli altri indici. Infine osserviamo che solamente il BIC risulta molto influenzato anche dall'interazione tra errata specificazione e numerosità, negli altri indici l'effetto di interazione è trascurabile.
In figura 6 sono rappresentate con grafici a scatola e baffi le distribuzioni campionarie degli indici di adattamento ottenute con il Monte Carlo. Per ciascun indice i grafici sono divisi in tre parti, a sinistra i risultati nella condizione di corretta specificazione del modello, al centro nella condizione di omissione del parametro ϕ_{21}, a destra nella condizione di omissione del parametro λ^x_{42}. In ogni parte sono presenti tre scatole relative alle diverse numerosità campionarie utilizzate (bianco per $N = 50$; grigio chiaro per $N = 100$, grigio scuro per $N = 1000$).
L'analisi dei grafici ci permette di interpretare con maggior grado di dettaglio i risultati della tabella 1. GFI, AGFI e NFI hanno delle distribuzioni praticamente identiche: l'effetto dell'errata specificazione (ricordiamo intorno al 14%) si traduce in una riduzione dei valori di tali

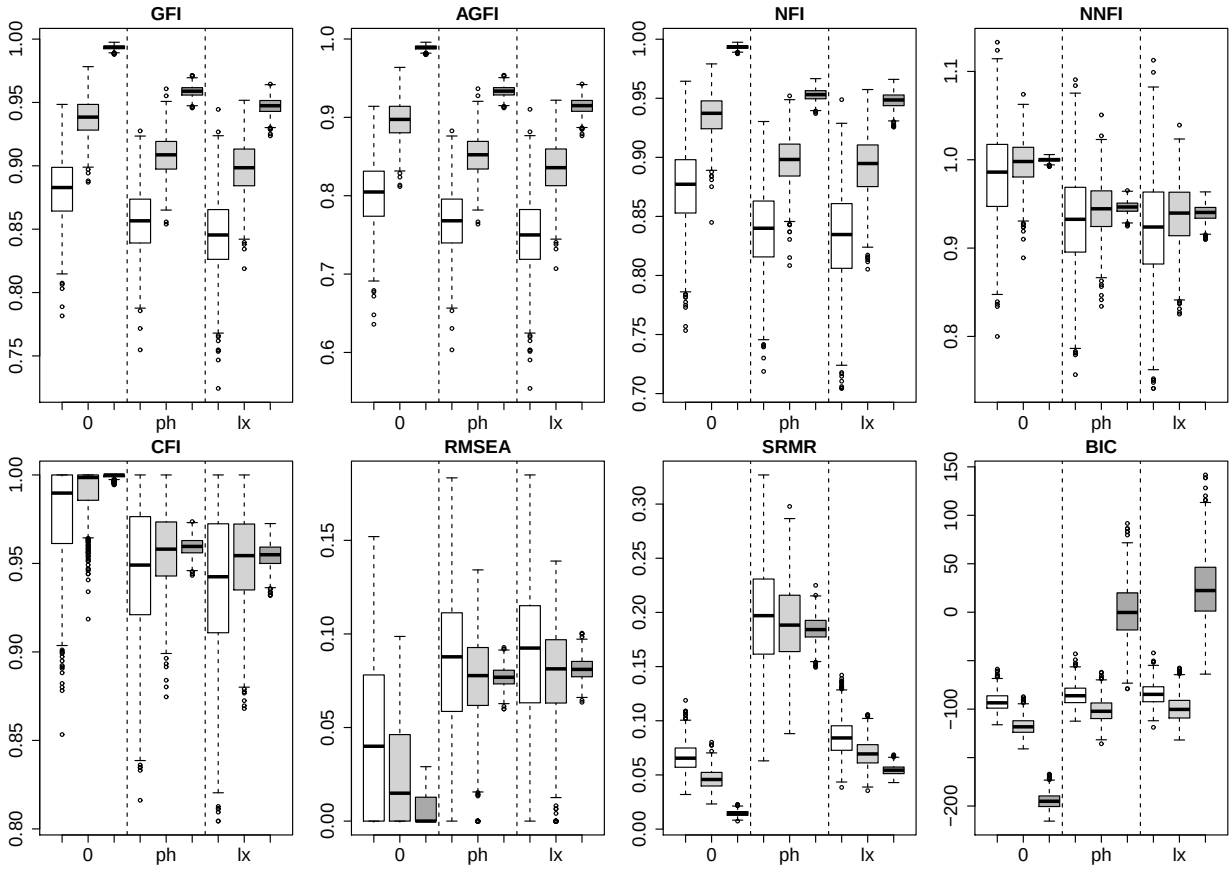

Figura 6: Distribuzioni Monte Carlo relative agli otto indici considerati in funzione del tipo di errata specificazione del modello e della numerosità campionaria. 0 indica corretta specificazione, `ph` indica omissione del parametro di correlazione tra i fattori, `lx` indica omissione del parametro λ. Le scatole bianche sono relative ai campioni di numerosità 50, le scatole grigio-chiaro sono relative alla numerosità 100, le scatole grigio-scuro relative alla numerosità 1000.

indici; in particolare l'errata omissione del parametro λ comporta un maggiore peggioramento rispetto all'errata omissione del parametro ϕ. Più forte l'effetto della numerosità campionaria (circa 86%) in base alla quale l'aumento di N implica un miglioramento dei valori degli indici. NNFI risulta fortemente influenzato dall'errata specificazione del modello (93%) ma entrambi i tipi di omissione comportano circa la stessa riduzione nella media dei valori di tale indice (la media nella condizione omissione di ϕ è 0.94, nella condizione omissione di λ è 0.93). Poco rilevante l'effetto della numerosità campionaria (7%), come confermato graficamente dal fatto che le mediane (rappresentate dalla linea nera interna alle scatole) risultano molto vicine. Per CFI l'effetto relativo all'errata specificazione è più ridotto (86%), ma anche in questo caso non emergono differenze tra le due tipologie di omissioni (le medie in entrambe le condizioni sono di 0.95). In questo indice risulta più evidente l'effetto della numerosità (13%) nel senso che campioni più ampi tendono ad avere migliori indici di adattamento.
A differenza dei cinque indici precedenti, RMSEA, SRMR e BIC esprimono un migliore grado di adattamento quanto più sono bassi. RMSEA è molto influenzato dall'omissione dei parametri del modello (90%) nel senso di un peggioramento del grado di adattamento, ma senza particolari differenze tra i due tipi di errore (le medie in entrambe le condizioni sono

di 0.80). Per questo indice osserviamo anche un lieve effetto della numerosità campionaria (5%) e di interazione tra errata specificazione e numerosità (5%). SRMR risulta, tra quelli considerati, l'indice più sensibile alle omissioni dei parametri (95%). Dalla lettura del grafico rileviamo però una situazione inversa al caso degli indici GFI, AGFI e NFI ossia che vi è un peggioramento più evidente nel caso di omissione del parametro ϕ (il valore medio dell'indice in questa condizione è di 0.19) rispetto all'omissione del parametro λ (valore medio 0.07). L'effetto della numerosità campionaria è molto ridotto (5%).
Diversamente dagli altri, che sono indici ad-hoc per i modelli di equazioni strutturali, BIC può essere applicato in molti contesti diversi per il confronto tra modelli (Raftery, 1995). In particolare, valori negativi indicano che il modello risulta più supportato dai dati rispetto ad un modello saturo, il cui valore di BIC è zero. Osserviamo che tale indice presenta un comportamento piuttosto diverso dagli altri, in particolare risulta molto sensibile all'interazione tra le condizioni sperimentali (52%). Nelle condizioni di numerosità 50 e 100 non emergono particolari differenze tra i valori medi di BIC nella condizione di modello corretto (la media è -92.58 per $N = 50$ e -117.51 per $N = 100$) verso quelle di errata specificazione (rispettivamente -85.64 per $N = 50$ e -101.28 per $N = 100$ in caso di omissione di ϕ, -84.24 per $N = 50$ e -99.21 per $N = 100$ in caso di omissione di λ). Questa interazione si esprime nella differenza esistente tra i valori medi di BIC per $N = 1000$. Nella condizione con $N = 1000$ abbiamo invece una differenza rilevante tra il modello corretto (media -194.65) e le condizioni di errata specificazione (rispettivamente 0.89 in caso di omissione di ϕ e 24.47 nel caso di omissione di λ).

6 Conclusioni

In questo capitolo abbiamo descritto i principi generali per le simulazioni Monte Carlo con i modelli di equazioni strutturali. Va ribadito però che l'approccio Monte Carlo ricopre una vasta gamma di possibili contesti di applicazione, tutti caratterizzati dalla presenza di processi casuali. Con i metodi Monte Carlo è possibile studiare le distribuzioni di variabili casuali, confrontare diverse procedure statistiche, analizzare il comportamento di sistemi complessi (Gentle, 2005). Volendo schematizzare, le simulazioni Monte Carlo hanno essenzialmente due tipi di applicazione in campo statistico: nello studio dei metodi statistici e nell'analisi dei dati. Un esempio del primo caso è quello che abbiamo presentato nella sezione precedente per valutare le prestazioni degli indici di adattamento nel caso di errata specificazione del modello. Un esempio del secondo caso si può ritrovare nell'approccio Bayesiano, in particolare nella valutazione delle distribuzioni a posteriori (si veda ad esempio Albert, 2008; Gelman, Carlin, Stern e Rubin, 2004). In relazione alle equazioni strutturali, le simulazioni Monte Carlo risultano particolarmente utili in due casi: nella valutazione delle conseguenze legate alla violazione degli assunti e nel fornire utili informazioni sulle statistiche di cui non è nota la distribuzione campionaria (Fan e Fan, 2005). O ancora, come propongono Muthén e Muthén (2002), per stabilire la numerosità campionaria necessaria per disporre di un certo livello di potenza.

Di fatto, l'importanza delle simulazioni Monte Carlo è riconosciuta anche nella presenza sempre più frequente di modaltità assistite per il loro utilizzo in molti software dedicati; per esempio LISREL (Jöreskog e Sörbom, 1996b), EQS (Bentler, 2004), Mplus (Muthén e Muthén, 2010) permettono di generare dati per studi di tipo Monte Carlo. Amos (Arbuckle, 2009) permette di utilizzare un tipo di generazione dati Monte Carlo nelle analisi di tipo Bayesiano.

In questo capitolo abbiamo presentato solo alcuni semplici esempi di come implementare alcuni algoritmi in ambiente R. Sono già disponibili alcune librerie per varie tipologie di utilizzo, dai modelli più semplici (`mc2d`; Pouillot, Delignette-Muller, Kelly, Denis, 2011) a quelli più complessi, ad esempio con Catene di Markov (`MCMCpack`; Martin, Quinn e Park, 2011). Vi sono anche alcuni casi di applicazioni relative ai modelli di equazioni strutturali, per esempio `MBESS` (Kelley e Lai, 2011) consente di studiare preventivamente un modello per valutare quale possa essere la dimensione campionaria ottimale, o più semplicemente `psych` (Revelle, 2011) consente di simulare dati a partire dai coefficienti di misura e strutturali di un modello. Infine vogliamo citare la libreria `lavaan` (Rosseel, 2012) che, oltre ad offrire un sistema molto flessibile per l'analisi di varie tipologie di modelli di equazioni strutturali, permette di simulare dati a partire da un modello specificato.

Riferimenti bibliografici

Albert, J. (2008). *Bayesian Computation with R*. Springer, NY.

Arbuckle, J. L. (1996). Full information estimation in the presence of incomplete data. In G. A. Marcoulides & R. E. Schumacker (Eds.), *Advanced structural equation modeling: Issues and techniques* pp. 243-277. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Arbuckle, J. L. (2009). *Amos 18 User's Guide*. Chicago, IL: SPSS Inc.

Beauducel, A., Wittmann, W. W. (2005). Simulation Study on Fit Indexes in CFA Based on Data With Slightly Distorted Simple Structure. *Structural Equation Modeling*, 12, 41-75.

Bentler, P. M. (2004). *EQS 6 Structural Equations Program Manual*. Encino, CA: Multivariate Software, Inc.

Bollen, K. A. (1989). *Structural equations with latent variables*. Wiley, NY.

Boomsma, A. (1985). Nonconvergence, improper solutions, and starting values in LISREL maximum likelihood estimation, *Psychometrika*, 50, 229-242.

Dalgaard, P. (2002). *Introductory Statistics with* R. Springer, NY.

Davey, A., Savla, J., Luo, A. (2005). Issues in Evaluating Model Fit With Missing Data. *Structural Equation Modeling*, 12, 578-597.

Drasgow, F. (1986). Polychoric and polyserial correlations. In S. Kotz & N. Johnson (Eds.), *The Encyclopedia of Statistics*, Volume 7, pp. 68-74. Wiley.

Enders, C. K., Bandalos, D. L. (2001). The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural Equation Modeling*, 8, 430-457.

Fan, X., Fan, X. (2005). Using SAS for Monte Carlo Simulation Research in SEM, *Structural Equation Modeling*, 12, 299-333.

Fan, X., Sivo, S. A. (2005). Sensitivity of Fit Indexes to Misspecified Structural or Measurement Model Components: Rationale of Two-Index Strategy Revisited, *Structural Equation Modeling*, 12, 343-367.

Fan, X., Sivo, S. A. (2007). Sensitivity of Fit Indices to Model Misspecification and Model Types, *Multivariate Behavioral Research*, 42, 509-529.

Fan, X., Thompson, B., Wang, L. (1999). Effects of Sample size, Estimation Methods, and Model Specification on Structural Equation Modeling Fit Indexes. *Structural Equation Modeling*, 6, 56-83.

Flora, D. B., Curran, P. J. (2004). An Empirical Evaluation of Alternative Methods of Estimation for Confirmatory Factor Analysis With Ordinal Data, *Psychological Methods*, 9, 466-491.

Fox, J. (2006). Structural Equation Modeling With the `sem` Package in R, *Structural Equation Modeling*, 13, 465-486.

Fox, J. (2009). Aspects of the Social Organization and Trajectory of the R Project, *The R Journal*, 1-2, 5-13.

Fox, J. (2010). *polycor: Polychoric and Polyserial Correlations*. R package version 0.7-8. `http://CRAN.R-project.org/package=polycor`

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (2004). *Bayesian Data Analysis (Second Edition)*. Chapman & Hall/CRC, FL.

Gentle, J. E. (2005). Monte Carlo Simulation. In B. S. Everitt & D. C. Howell (Eds.), *Encyclopedia of Statistics in Behavioral Science*, Volume 3, pp. 1264-1271. Wiley.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., Hothorn, T. (2011). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-999. `http://CRAN.R-project.org/package=mvtnorm`

Gerbing, D.W., Anderson, J.C. (1993). Monte Carlo evaluations of goodness-of-fit indices for structural equation models. In K.A. Bollen, J.S. Long (Eds.), *Testing Structural Equation Modeling*, pp. 40-65. Newbury Park, CA: SAGE.

Green, S. B., Yang, Y. (2009). Reliability of Summed Item Scores Using Structural Equation Modeling: An Alternative To Coefficient Alpha, *Psychometrika*, 74, 155-167.

Hu, L., Bentler, P. M. (1998). Fit Indices in Covariance Structure Modeling: Sensitivity to Underparameterized Model Misspecification, *Psychological Methods*, 3, 424-453.

Hu, L., Bentler, P. M. (1999). Cutoff Criteria for Fit Indices in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives, *Structural Equation Modeling*, 6, 1-55.

Iacus, S., Masarotto, G. (2003). *Laboratorio di statistica con* R. McGraw-Hill, Milano.

Jackson, D. L. (2007). The Effect of the Number of Observations per Parameter in Misspecified Confirmatory Factor Analytic Models. *Structural Equation Modeling*, 14, 48-76.

Jöreskog, K., Sörbom, D. (1996a). *LISREL 8: User's Reference Guide.* Scientific Software International, Inc., Lincolnwood, IL.

Jöreskog, K., Sörbom, D. (1996b). *PRELIS 2: User's Reference Guide.* Scientific Software International, Inc., Lincolnwood, IL.

Kaplan, D. (1989). A Study of the Sampling Variability and z-Values of Parameter Esitmates From Misspecified Structural Equation Models. *Multivariate Behavioral Research*, 24, 41-57.

Kelley, K., Lai, K. (2011). *MBESS: MBESS*. R package version 3.2.1. `http://CRAN.R-project.org/package=MBESS`

Lei, P. W. (2009). Evaluating estimation methods for ordinal data in structural equation modeling. *Quality & Quantity*, 43, 495-507.

Martin, A. D., Quinn, K. M., Park, J. H. (2011). *MCMCpack: Markov chain Monte Carlo (MCMC).* R package version 1.1-1. `http://mcmcpack.wustl.edu`

Matsumoto, M., Nishimura, T. (1998). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Transactions on Modeling and Computer Simulation*, 8, 3-30.

Merriam-Webster, Inc. (1994). *Merriam-Webster's Collegiate Dictionary.* 10th ed. Springfield, MA: Merriam-Webster, Inc.

Muenchen, R. A. (2010). *The Popularity of Data Analysis Software*, disponibile on-line `http://r4stats.com/popularity`.

Muthén, B., Kaplan, D., Hollis, M. (1987). On structural equation modeling with data that are not missing completely at random. *Psychometrika*, 52, 431-462.

Muthén, L. K., Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling*, 4, 599-620.

Muthén, L. K., Muthén, B. O. (2010). *Mplus User's Guide.* Sixth Edition. Los Angeles, CA: Muthén & Muthén.

Olsson, U. H., Foss, T., Troye, S. V., Howell, R. D. (2000). The Performance of ML, GLS, and WLS Estimation in Structural Equation Modeling Under Conditions of Misspecification and Nonnormality. *Structural Equation Modeling*, 7, 557-595.

Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., Chen, F. (2001). Monte Carlo Experiments: Design and Implementation, *Structural Equation Modeling*, 8, 287-312.

Pouillot, R., Delignette-Muller, M. L., Kelly, D. L., Denis, J. B. (2011). *mc2d: Tools for Two-Dimensional Monte-Carlo Simulations.* R package version 0.1-11. `http://CRAN.R-project.org/package=mc2d`

R Development Core Team (2010). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`.

Raftery, A. E. (1995). Bayesian Model Selection in Social Research. *Sociological Methodology*, 25, 111-163.

Revelle, W. (2011) *psych: Procedures for Personality and Psychological Research*. R package version 1.0-97. `http://personality-project.org/r/psych.manual.pdf`

Rizzo, M. L. (2008). *Statistical Computing with* R. Chapman & Hall, Boca Raton, FL.

Robert, C. P., Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, NY.

Rosseel, Y. with contributions from many people (2012). *lavaan: Latent Variable Analysis*. R package version 0.4-12. `http://CRAN.R-project.org/package=lavaan`

Schafer, J. L., Graham, J. E. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 7, 147-177.

Sivo, S. A., Fan, X., Witta, E. L., Willse, J. T. (2006). The Search for "Optimal" Cutoff Properties: Fit Index Criteria in Structural Equation Modeling, *The Journal of Experimental Education*, 74, 267-288.

Venables, W. N., Ripley, B. D. (2002). *Modern Applied Statistics with S.* Fourth Edition. Springer, New York.

Ximénez, C., (2006). A Monte Carlo Study of Recovery of Weak Factor Loadings in Confirmatory Factor Analysis, *Structural Equation Modeling*, 13, 587-614.

Yang, Y., Green, S. B. (2010). A Note on Structural Equation Modeling Estimates of Reliability, *Structural Equation Modeling*, 17, 66-81.

Yuan, K. H., Kouros, C. D., Kelley, K. (2008). Diagnosis for Covariance Structure Models by Analyzing the Path. *Structural Equation Modeling*, 15, 564-602.

Yuan, K. H., Marshall, L. L., Bentler, P. M. (2003). Assessing the Effect of Model Misspecifications on Parameter Estimates in Structural Equation Models. *Sociological Methodology*, 33, 241-265.