

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI SCIENZE STATISTICHE
CORSO DI LAUREA TRIENNALE IN
STATISTICA PER L'ECONOMIA E L'IMPRESA



RELAZIONE FINALE

**Errata specificazione della distribuzione degli effetti
casuali in modelli misti:
conseguenze e strumenti per la diagnosi**

Relatore Prof. Alessandra Salvan
Dipartimento di Scienze Statistiche

Laureando Francesco Bonato
Matricola N° 1220250

Anno Accademico 2021/2022

Indice

Introduzione	1
1 I modelli misti	3
1.1 I modelli per risposte correlate	3
1.1.1 I modelli marginali	4
1.1.2 I modelli con effetti individuali	4
1.2 I modelli con effetti casuali	5
1.2.1 Modelli con effetti casuali per risposte normali	5
1.2.2 Modelli con effetti casuali per risposte non normali	9
1.2.3 Previsione degli effetti casuali	10
2 La specificazione della distribuzione degli effetti casuali	11
2.1 Problemi relativi a un'errata specificazione	11
2.2 Proprietà asintotiche del test rapporto di verosimiglianza	13
2.3 Strumenti di diagnostica	14
2.3.1 Test di Tchetgen & Coull (2006)	14
2.3.2 Test del determinante di Alonso et al. (2008)	15
2.3.3 Test della traccia-determinante di Alonso et al. (2008)	16
2.3.4 La funzione gradiente di Verbeke & Molenberghs (2013)	18
2.3.5 Test di Drikvandi et al. (2017)	18
3 Studio di simulazione	21
3.1 Disegno dello studio	21
3.2 Risultati dello studio	24
3.3 Implementazione della funzione gradiente	28
3.4 Implementazione della statistica di Drikvandi et al. (2017)	31
Conclusione	32
Appendice A	35
Appendice B	37
Appendice C	39

Appendice D	41
D.1 Grafici	41
D.2 Convergenza nelle stime	48
Appendice E	51
E.1 Studio di simulazione	51
E.2 Funzione gradiente	69
E.3 Statistica test di Drikvandi et al. (2017)	72
Bibliografia	73

Introduzione

In molte aree di ricerca, (ad esempio in medicina, nel campo economico o delle scienze sociali) ci si può trovare in presenza di dati longitudinali, con più osservazioni ripetute sugli stessi soggetti nel corso di un arco temporale, oppure si può misurare una stessa variabile su cluster di soggetti (si pensi ad interviste ai componenti di un nucleo familiare). La variabile risposta in questi casi è multivariata con componenti correlate. I modelli misti sono una classe di modelli comunemente usata per le analisi di tali dati, perché permettono un approccio flessibile e sono facilmente implementabili (in R, ad esempio, sono disponibili le librerie `nlme` e `lme4` per modelli misti rispettivamente per risposte normali e non).

Per catturare l'eterogeneità dovuta alle singolari caratteristiche degli individui o gruppi di individui si incorporano nel modello effetti casuali, i quali spiegano una parte della variabilità che si può osservare nei dati. Si assume generalmente che gli effetti casuali siano generati da una distribuzione della famiglia normale.

L'obiettivo di questo lavoro è analizzare come i risultati inferenziali sulle varie componenti del modello possano essere distorti quando tale assunzione non è valida. Si desidera inoltre fornire una rassegna degli strumenti per la diagnosi che permettono di valutare la coerenza di un'ipotesi distributiva per gli effetti casuali con i dati osservati.

Questa relazione è basata principalmente sui contenuti degli articoli di Hui et al. (2021) e Verbeke & Molenberghs (2013). Nel Capitolo 1 si introdurranno i modelli misti e la notazione che verrà utilizzata. Nel Capitolo 2 si presenterà il dibattito tuttora aperto nella letteratura contemporanea sull'entità dell'impatto che un'errata specificazione della distribuzione degli effetti casuali può avere sull'inferenza e saranno illustrati i vari test che sono stati ideati per valutare la bontà di adattamento rispetto all'ipotesi che si assume per gli effetti casuali, con particolare attenzione alla funzione gradiente introdotta da Verbeke & Molenberghs (2013). Nel Capitolo 3 saranno analizzate le procedure inferenziali in condizioni di errata specificazione e sarà presentato un esempio

di implementazione della funzione gradiente e della relativa statistica test di Drikvandi et al. (2017) con R.

Capitolo 1

I modelli misti

Introduzione

L'obiettivo di questo capitolo è di introdurre i modelli con effetti casuali, passando prima per la cornice generale dei modelli per risposte correlate per poi presentare più nel dettaglio i modelli misti e la notazione che verrà utilizzata nel resto del lavoro. I contenuti sono basati su Salvani et al. (2020), Capitolo 7.

1.1 I modelli per risposte correlate

In molte situazioni la variabile risposta Y_i per ogni unità, $i = 1, \dots, n$, è multivariata. Si pensi a dati *panel*, risultato di studi osservazionali longitudinali, in cui si misura una caratteristica di un soggetto ripetutamente a istanti temporali diversi (per esempio, in campo clinico, la risposta dei pazienti a un farmaco registrata a istanti temporali diversi); o alle serie storiche, che registrano l'andamento di una variabile nel corso di un arco temporale; oppure ancora a quando l'unità statistica non è il singolo soggetto ma piuttosto un cluster di soggetti che si suppongono essere accomunati da determinate caratteristiche.

Si hanno dunque in questi casi n unità e m_i osservazioni su ciascuna unità

$$y_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, m_i,$$

dove y_{ij} , valore osservato per la risposta dell'unità i all'occasione j , è una realizzazione

della variabile casuale Y_{ij} , il cui valore atteso è $\mu_{ij} = \mathbb{E}(Y_{ij})$. Inoltre,

$$y_i = (y_{i1}, \dots, y_{im_i})^T, \quad i = 1, \dots, n,$$

è il vettore che contiene le osservazioni relative all'unità i , realizzazione di Y_i ; il numero di osservazioni complessive è quindi $N = \sum_{i=1}^n m_i$. Ci si aspetta che le variabili casuali Y_{ij} e Y_{ih} , con $j \neq h$, relative alla medesima unità i , siano correlate; i modelli che si prendono in considerazione per studiare relazioni tra le Y_i e possibili variabili esplicative devono tenere in conto tale correlazione. Si assumerà invece indipendenza tra osservazioni relative a unità distinte. Si distinguono due possibili approcci per modellare una risposta multivariata correlata, i modelli marginali e i modelli con effetti individuali.

1.1.1 I modelli marginali

I modelli marginali descrivono la relazione tra le variabili esplicative e i valori attesi marginali μ_{ij} . Nei modelli marginali, la correlazione tra osservazioni sulla stessa unità non è un aspetto centrale ma è piuttosto una componente di disturbo, di cui è necessario tener conto per valutare correttamente gli *standard error* degli stimatori. Si possono avere modelli marginali per risposte normali (ovvero un modello lineare normale multivariato nel quale si andrà ad assumere una determinata struttura per la matrice di varianze e covarianze delle risposte Y_i) o per risposte non normali (come nel caso di dati binari o di conteggio) attraverso le *Generalized Estimating Equations*, (*GEE*) proposte da Liang & Zeger (1986), assieme alla cosiddetta stima *sandwich* della matrice di varianze e covarianze. Per una trattazione più approfondita dei modelli marginali si rimanda a Salvan et al. (2020), Capitolo 7.

1.1.2 I modelli con effetti individuali

Nei modelli con effetti individuali si assume che vi siano caratteristiche non osservabili comuni a tutte le osservazioni relative alla stessa unità. Una prima tipologia è data dai **modelli con effetti fissi**, nei quali viene introdotto nel modello un parametro, detto incidentale, per ogni unità $i = 1, \dots, n$; ne consegue direttamente che lo spazio parametrico ha dimensione che cresce al crescere di n e sono quindi violate le condizioni di regolarità per la consistenza degli stimatori di massima verosimiglianza. Una seconda soluzione è data dai **modelli con effetti casuali**, in cui le caratteristiche comuni alle osservazioni sulla stessa unità sono invece modellate come realizzazioni di una variabile casuale, che viene chiamata **effetto casuale**; l'assunzione sulla distribuzione degli effetti

casuali (*random-effects distribution* o *mixing distribution*), costituirà l'oggetto di questo lavoro. I modelli con effetti casuali sono chiamati anche **modelli misti**, in virtù del fatto che comprendono congiuntamente effetti fissi e casuali. Si possono avere modelli misti per risposte normali e non.

1.2 I modelli con effetti casuali

1.2.1 Modelli con effetti casuali per risposte normali

Quando la variabile risposta è continua e ha come supporto l'intero asse reale, si possono considerare **modelli lineari con effetti misti** (LMM, *Linear Mixed Models*), i quali assumono che la distribuzione di Y condizionatamente agli effetti casuali segua una normale. Un primo esempio di modello (lineare normale) con effetto casuale assume che le osservazioni y_{ij} siano realizzazioni di Y_{ij} , con Y_{ij} del tipo

$$Y_{ij} = \mu + a_i + \epsilon_{ij}, \quad (1.1)$$

dove μ è un effetto fisso, "globale" di intercetta, a_i è un effetto casuale, condiviso da tutte le Y_{i1}, \dots, Y_{im} che spiega l'eterogeneità tra le osservazioni sull'unità i e le osservazioni sulle altre unità $j \neq i$ dovuta alle particolari caratteristiche di i , mentre ϵ_{ij} è il termine di errore casuale. Per semplicità si assume, qui e per tutto il resto del lavoro, che $m_1 = \dots = m_n = m$, ovvero che il numero di osservazioni per ogni unità sia costante. Tutti le variabili a_i e ϵ_{ij} sono assunte indipendenti per ogni $i = 1, \dots, n$ e per ogni $j = 1, \dots, m$. Si assume usualmente che entrambi questi due termini casuali provengano dalla famiglia normale

$$a_i \sim N(0, \sigma_a^2), \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2).$$

In questo modello la risposta ha dunque distribuzione normale, con valore atteso

$$\mathbb{E}(Y_{ij}) = \mu$$

valori attesi condizionati

$$\mathbb{E}(Y_{ij}|a_i) = \mu + a_i$$

e varianza

$$\text{Var}(Y_{ij}) = \sigma_a^2 + \sigma_\epsilon^2.$$

È facile verificare che una tale struttura induce inoltre correlazione tra le osservazioni relative alla stessa unità. Si osservi infatti che

$$Cov(Y_{ij}, Y_{ih}) = Cov(a_i + \epsilon_{ij}, a_i + \epsilon_{ih}) = \mathbb{E}((a_i + \epsilon_{ij})(a_i + \epsilon_{ih})) = \sigma_a^2,$$

dove è sfruttata la proprietà $Cov(X + a, Y + b) = Cov(X, Y)$ (con a, b scalari qualsiasi). La correlazione è dunque data da

$$Cor(Y_{ij}, Y_{ih}) = \frac{Cov(Y_{ij}, Y_{ih})}{\sqrt{Var(Y_{ij})}\sqrt{Var(Y_{ih})}} = \frac{\sigma_a^2}{(\sigma_a^2 + \sigma_\epsilon^2)} = \rho.$$

Si osservi che all'aumentare della varianza delle intercette casuali σ_a^2 aumenta la correlazione tra osservazioni sulla stessa unità. Osservazioni relative a unità differenti sono invece indipendenti: $Cov(Y_{ij}, Y_{i'j}) = Cov(a_i + \epsilon_{ij}, a_{i'} + \epsilon_{i'j}) = 0$ se $i \neq i'$. La correlazione tra osservazioni distinte relative alla medesima unità è dunque ρ , detto anche in questo caso **coefficiente di correlazione intra-classe**, positivo e costante. Si ha quindi una struttura di **equicorrelazione** per la matrice di varianze e covarianze di Y_i

$$Var(Y_i) = (\sigma_a^2 + \sigma_\epsilon^2) \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{pmatrix}.$$

Il modello misto (1.1) prevede un unico effetto casuale per il termine di intercetta, ma è possibile anche includere molteplici effetti casuali. Si consideri ad esempio il modello

$$Y_{ij} = \beta_1 + a_i + (\beta_2 + b_i)j + \epsilon_{ij} \quad (1.2)$$

dove j indica l'istante temporale nel quale abbiamo osservato la caratteristica sull'unità i ; si sta dunque assumendo una relazione lineare tra tempo e variabile risposta in cui β_1 e β_2 sono effetti fissi e rappresentano intercetta e coefficiente angolare globali, a livello di popolazione, mentre a_i e b_i sono effetti casuali che spiegano la deviazione dal livello di intercetta e coefficiente angolare di popolazione che si osserva nelle misurazioni relative all'unità i dovute alle particolari caratteristiche di tale unità. $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$ è il termine di errore casuale. Una semplice assunzione per gli effetti casuali è

$$a_i \sim N(0, \sigma_a^2), \quad b_i \sim N(0, \sigma_b^2),$$

con a_i, b_i, ϵ_{ij} indipendenti. In tale modello vale

$$\mathbb{E}(Y_{ij}) = \beta_1 + \beta_2 j,$$

$$\mathbb{E}(Y_{ij}|a_i, b_i) = \beta_1 + a_i + (\beta_2 + b_i)j,$$

$$\text{Var}(Y_{ij}) = \sigma_a^2 + \sigma_b^2 j^2 + \sigma_\epsilon^2,$$

$$\text{Cov}(Y_{ij}, Y_{ih}) = \sigma_a^2 + \sigma_b^2 jh.$$

Si osservi inoltre che non vi è più struttura di equicorrelazione per $\text{Var}(Y_i)$ (ma vi è comunque correlazione non nulla e positiva tra osservazioni sulla stessa unità i). In (1.2) si hanno rette di regressione che descrivono la relazione lineare tra variabile dipendente e risposta per ogni unità, $i = 1, \dots, n$, che differiscono non solo per il termine di intercetta ma anche per la loro pendenza.

Una forma generale per i modelli misti lineari normali è data da

$$Y_{ij} = x_{ij}\beta + z_{ij}u_i + \epsilon_{ij}, \quad (1.3)$$

dove β è il vettore p -dimensionale che contiene gli effetti fissi mentre u_i è il vettore q -dimensionale che contiene gli effetti casuali, che si assume essere realizzazione di una variabile casuale q -dimensionale con distribuzione $G(\cdot)$ (la *mixing distribution*), parametrizzata da ψ , vettore i cui elementi sono ignoti. L'assunzione più comune su $G(\cdot)$ è che sia una normale multivariata con media nulla e matrice di varianze e covarianze data da Σ_u . È frequente inoltre assumere che Σ_u sia una matrice diagonale (e quindi effetti casuali indipendenti), ma è possibile anche prevedere una struttura di correlazione tra effetti casuali. Gli elementi ignoti di Σ_u sono detti **componenti di varianza** (*variance components*). Si indica con x_{ij} il vettore delle esplicative associate all'osservazione Y_{ij} mentre le componenti di z_{ij} sono in genere un sottoinsieme di quelle di x_{ij} . Il termine $z_{ij}u_i$ spiega la variabilità tra unità, mentre ϵ_{ij} spiega la variabilità entro le unità. I parametri contenuti in β possono essere associati a variabili esplicative che dipendono solo dall'unità i (**effetti fissi tra unità**, *between-cluster*) come anche da j (**effetti fissi entro le unità**, *within-cluster*). Si osservi che il modello (1.2) può essere scritto nella forma (1.3) dove

$$x_{ij}\beta = \begin{pmatrix} 1 & j \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$$

e

$$z_{ij}u_i = \begin{pmatrix} 1 & j \end{pmatrix} \begin{pmatrix} a_i \\ b_i \end{pmatrix}$$

e si assume dunque

$$u_i \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{pmatrix}\right).$$

È possibile riscrivere (1.3) nella forma matriciale

$$Y_i = X_i\beta + Z_i u_i + \epsilon_i, \quad (1.4)$$

dove X_i è una matrice $m \times p$ con generica riga x_{ij} e Z_i è una matrice $m \times q$ con generica riga z_{ij} ; si assume $\epsilon_i \sim N_m(0, \sigma_\epsilon^2 I_m)$. Il valore atteso condizionato è dato dunque da

$$\mathbb{E}(Y_i|u_i) = X_i\beta + Z_i u_i$$

e la matrice di varianze e covarianze da

$$\text{Var}(Y_i) = Z_i \Sigma_u Z_i^T + \sigma_\epsilon^2 I_m \quad (1.5)$$

Tale forma evidenzia come la matrice di varianze e covarianze di Y_i possa essere decomposta in una componente relativa alla variabilità tra unità (il primo addendo) e un'altra relativa alla variabilità entro le unità. È possibile estendere (1.5) assumendo più in generale $\epsilon_i \sim N_m(0, R)$ dove con R si dà una struttura di correlazione (ad esempio autoregressiva) agli errori casuali relativi alla stessa unità

$$\text{Var}(Y_i) = Z_i \Sigma_u Z_i^T + R \quad (1.6)$$

essendo l'assunzione di indipendenza tra errori su osservazioni sulla stessa unità infatti in molte situazioni poco realistica.

Per quanto riguarda l'inferenza, il modello (1.4) ha in generale parametro $\theta = (\beta, \psi, R)$; l'inferenza su β si ottiene generalmente stimando prima le componenti ψ e R con il metodo della **massima verosimiglianza residua o ristretta (REML, REstricted Maximum Likelihood estimation)**, che sfrutta una trasformazione AY tale che la distribuzione non dipenda più da β . Per una trattazione più approfondita del metodo REML, si rimanda a Davison (2003, paragrafo 12.2). In R, la stima di modelli lineari normali misti si può ottenere con la funzione `lme` contenuta nella libreria `nlme` (Pinheiro et al., 2017).

1.2.2 Modelli con effetti casuali per risposte non normali

È possibile estendere il modello lineare normale con effetti misti anche a risposte non normali (come ad esempio nel caso di dati binari o di conteggio) assumendo che condizionatamente al vettore di effetti casuali u_i , Y_{ij} segua una distribuzione appartenente alla famiglia di dispersione esponenziale, con densità

$$p(y_{ij}; \alpha_{ij}, \phi | u_i) = \exp \left\{ \frac{\alpha_{ij} y_{ij} - b(\alpha_{ij})}{a_{ij}(\phi)} + c(y_{ij}, \phi) \right\},$$

dove $a_{ij}(\cdot)$, $b(\cdot)$ e $c(\cdot)$ sono funzioni note mentre α_{ij} è detto parametro **naturale** e ϕ parametro **di dispersione**. Un tale modello è detto **modello lineare generalizzato con effetti misti** (GLMM, *Generalized Linear Mixed Model*). Per il valore atteso condizionato delle Y_{ij} vale

$$g(\mathbb{E}(Y_{ij}|u_i)) = x_{ij}\beta + z_{ij}u_i \quad (1.7)$$

dove $g(\cdot)$ è la funzione legame (*link function*). La distribuzione condizionata è data dunque da

$$Y_{ij}|u_i \sim DE_1(\mathbb{E}(Y_{ij}|u_i), a_{ij}(\phi)v(\mathbb{E}(Y_{ij}|u_i)))$$

dove $\mathbb{E}(Y_{ij}|u_i) = g^{-1}(x_{ij}\beta + z_{ij}u_i)$, e $v(\cdot)$ è la funzione di varianza e dipende dalla distribuzione della variabile risposta Y di interesse. Per approfondimenti sulla famiglia di dispersione esponenziale si rimanda a Salvani et al. (2020, paragrafo 2.1). Il valore atteso marginale di Y_{ij} è dato da

$$\mathbb{E}(Y_{ij}) = \mu_{ij} = \mathbb{E}(\mathbb{E}(Y_{ij}|u_i)) = \int g^{-1}(x_{ij}\beta + z_{ij}u_i)p(u_i; \psi)du_i,$$

dove $p(u_i; \psi)$ è la densità degli effetti casuali; è ottenuto dunque integrando rispetto alla distribuzione di u_i . Le stime dei parametri del modello sono ottenute a partire dalla funzione di **verosimiglianza marginale**, data da

$$L(\beta, \psi) = \int \prod_{i=1}^n \prod_{j=1}^m p(y_{ij}; \beta | u_i) p(u_i; \psi) du_i \quad (1.8)$$

e l'integrale viene calcolato con metodi numerici. In R, per la stima di modelli lineari generalizzati con effetti misti è disponibile la funzione `glmer` della libreria `lme4` (Bates et al., 2015).

1.2.3 Previsione degli effetti casuali

Poiché gli effetti casuali sono variabili casuali e non parametri, si parla di previsione e non di stima. La previsione degli effetti casuali può essere utile in molti contesti; ad esempio, trovare che un'unità è associata a un'intercetta casuale particolarmente lontana dal livello di popolazione potrebbe essere un fatto rilevante e da tenere in considerazione. Per i modelli misti lineari normali (1.4) si usa il predittore **BLUP** (*Best Linear Unbiased Predictor*, miglior predittore lineare non distorto), che restituisce le previsioni \tilde{u}_i per ogni unità $i = 1, \dots, n$, tali che $\mathbb{E}(a^T u_i - a^T \tilde{u}_i)^2$, l'errore quadratico medio per una qualunque combinazione lineare $a^T u_i$, sia minimo tra tutti i possibili previsori di u_i funzioni lineari di Y_i con media zero. Tale previsore coincide con il valore atteso condizionato $\mathbb{E}(u_i | Y_i = y_i)$ della distribuzione congiunta di Y_i e u_i , la quale, se si assume che $G(\cdot)$ sia una normale, è anch'essa una normale $m + q$ -variata.

$$\begin{pmatrix} Y_i \\ u_i \end{pmatrix} \sim N_{m+q} \left(\begin{pmatrix} X_i \\ 0 \end{pmatrix}, \begin{pmatrix} V_i & Z_i \Sigma_u \\ \Sigma_u Z_i^T & \Sigma_u \end{pmatrix} \right)$$

dove V_i è definita come in (1.6); la matrice di covarianza tra Y_i e u_i può essere facilmente calcolata

$$\text{Cov}(Y_i, u_i) = \mathbb{E}((X_i \beta + Z_i u_i + \epsilon_i) u_i^T) = Z_i \Sigma_u$$

in virtù dell'indipendenza tra u_i e ϵ_i . Per il risultato relativo alle distribuzioni condizionate in una normale multivariata richiamato nell'Appendice A, vale che

$$\mathbb{E}(u_i | Y_i = y_i) = \Sigma_u Z_i^T V_i^{-1} (y_i - X_i \beta)$$

e il previsore BLUP è dato quindi da

$$\tilde{u}_i = \Sigma_u Z_i^T V_i^{-1} (y_i - X_i \hat{\beta})$$

e se le matrici Σ_u e V_i sono ignote saranno sostituite con le loro rispettive stime. Anche per modelli misti lineari generalizzati (1.7) è possibile ottenere previsioni per gli effetti casuali stimando $\mathbb{E}(u_i | Y_i = y_i)$ con metodi numerici a partire dalla distribuzione congiunta $\begin{pmatrix} Y_i \\ u_i \end{pmatrix}$. In R, le previsioni degli effetti casuali sono fornite dalla funzione `ranef`, contenuta sia nella libreria `nlme` che nella libreria `lme4` (in `lme4` le previsioni si basano sulle mode di $u_i | Y_i = y_i$).

Capitolo 2

La specificazione della distribuzione degli effetti casuali

Introduzione

Assumendo la distribuzione $G(\cdot)$ per gli effetti casuali u_i , la funzione di verosimiglianza per i parametri del modello viene ottenuta integrando le densità condizionate agli effetti casuali rispetto alla loro distribuzione $G(\cdot)$. Si può infatti riscrivere la verosimiglianza trovata in (1.8) come

$$L(G) = \int \prod_{i=1} \prod_{j=1} p(y_{ij}|u_i) dG(u_i), \quad (2.1)$$

in cui viene messa in primo piano nella notazione la dipendenza della verosimiglianza dalla specificazione di $G(\cdot)$. È evidente dunque che un'errata specificazione può potenzialmente compromettere la bontà dei risultati inferenziali. Il capitolo presente ha il compito di presentare il dibattito sull'entità di questo problema e di illustrare alcuni strumenti di diagnosi per rilevare errori nella specificazione.

2.1 Problemi relativi a un'errata specificazione

Nella letteratura vi sono opinioni contrastanti riguardo al reale impatto di un'errata specificazione della distribuzione degli effetti casuali sulle procedure inferenziali. Verbeke & Lesaffre (1997) mostrano come gli stimatori di massima verosimiglianza degli effetti fissi e delle componenti di varianza sotto condizioni di regolarità sono consistenti e asintoticamente normali, anche se la velocità con la quale le stime convergono dipende dalla correttezza della specificazione di $G(\cdot)$. Chen et al. (2002) concludono che,

contrariamente a quanto si potrebbe intuire, sotto misspecificazione la distorsione nelle stime degli effetti fissi è trascurabile. McCulloch & Neuhaus (2011a) affermano che vi è elevata robustezza nella stima degli effetti fissi associati a variabili esplicative *within-cluster*, entro i soggetti (ovvero che variano tra osservazioni relative alla stessa unità, ad esempio il tempo), una modesta distorsione per le stime degli effetti fissi *between-cluster* nei casi estremi (ad esempio, si assume normalità quando la vera distribuzione degli effetti casuali è bimodale), mentre vi può essere distorsione per la stima dell'intercetta di popolazione e per le previsioni degli effetti casuali, ma comunque con buone proprietà in termini di errore quadratico medio (McCulloch & Neuhaus, 2011b). Tuttavia, poiché sono proprio le stime degli effetti fissi *within-cluster* in genere l'elemento di interesse primario per cui si conduce lo studio longitudinale, gli Autori concludono che la specificazione della distribuzione degli effetti casuali non è un aspetto di particolare preoccupazione. Agresti et al. (2004) consigliano invece una maggiore cautela, suggerendo di utilizzare congiuntamente un approccio parametrico e non parametrico per la specificazione della *mixing distribution*: la presenza di sostanziali differenze per i risultati inferenziali nei due casi costituisce un segnale che l'assunzione parametrica è problematica. Litière et al. (2008) mostrano attraverso uno studio di simulazione come ci possa essere considerevole distorsione nelle stime degli effetti fissi quando nel modello sono inseriti più di un effetto casuale e quando la varianza della distribuzione degli effetti casuali è elevata (cosa che accade spesso, ad esempio, nel caso di studi clinici con gruppo di controllo sottoposto a un placebo, in cui ci aspetta di osservare modesta variabilità nella risposta e gruppo di trattamento in cui invece la variabilità sarà più alta).

Riassumendo, in caso di non corretta specificazione di $G(\cdot)$ vi è concordanza nella letteratura riguardo a tre aspetti:

- la distorsione associata agli effetti fissi *within-cluster* è piccola, e i test su questi parametri hanno probabilità di errore di primo tipo vicina al livello nominale;
- l'intercetta fissa è invece soggetta a considerevole distorsione, e i test hanno probabilità di errore di primo tipo inflazionata;
- le previsioni per gli effetti casuali sono sensibili alla specificazione della *mixing distribution* e tenderanno a "mimare" la forma della distribuzione assunta, tuttavia le loro proprietà in termini di MSE (*Mean Squared Error*) rimangono in ogni caso buone.

Due questioni sono invece elementi di acceso dibattito:

- la distorsione (finita) per la stima degli effetti fissi *between-cluster*;

- la distorsione (finita) per la stima della matrice di varianze e covarianze di u_i nei modelli misti lineari.

2.2 Proprietà asintotiche del test rapporto di verosimiglianza

Hui et al. (2021) presentano due risultati importanti per l'inferenza nei modelli misti sotto errata specificazione della distribuzione degli effetti casuali. Sia $\theta = (\beta, \lambda, \sigma_\epsilon^2)$ il vettore dei parametri di un modello misto lineare normale; λ è la diagonale di Σ_u , matrice di varianze e covarianze degli effetti casuali, che si assume per semplicità avere elementi fuori dalla diagonale pari a zero. Sia $\hat{\theta}$ la stima di massima verosimiglianza per θ ottenuta massimizzando la (2.1). Sia $l(\theta) = \log(L(\theta))$ la funzione di log-verosimiglianza.

Risultato 1

Sia β_s un generico vettore i cui elementi sono un sottoinsieme degli elementi di β . Si vuole verificare $H_0 : \beta_s = \beta_s^0$ attraverso il test del rapporto di verosimiglianza, sfruttando la statistica test $T_\beta = 2\{l(\hat{\theta}) - l(\tilde{\theta})\}$ dove $\tilde{\theta}$ è la stima di massima verosimiglianza sotto l'ipotesi nulla. Sotto errata specificazione di $G(\cdot)$ vale che

$$P(T_\beta > \chi_{\dim(\beta_s)}^2\{1 - \alpha\} | H_0) \rightarrow \alpha \quad n \rightarrow \infty \quad (2.2)$$

dove $\chi_{\dim(\beta_s)}^2\{1 - \alpha\}$ indica il quantile $(1 - \alpha)$ di una distribuzione chi quadrato con $\dim(\beta_s)$ gradi di libertà. Il risultato (2.2) garantisce la validità dal punto di vista asintotico del test. Resta comunque da valutare l'accordo tra livello effettivo e livello nominale in presenza di piccoli campioni.

Risultato 2

Si voglia ora verificare $H_0 : \lambda_s = 0$ attraverso il test rapporto di verosimiglianza, dove λ_s indica un generico vettore i cui elementi sono un sottoinsieme degli elementi di λ . Sia $T_\lambda = 2\{l(\hat{\theta}) - l(\tilde{\theta})\}$ dove $\tilde{\theta}$ è la stima di massima verosimiglianza sotto l'ipotesi nulla. Si tenga in considerazione che le condizioni di regolarità sono violate, ci si pone infatti sulla frontiera dello spazio parametrico di λ (ovvero \mathbb{R}^{q+}). Sotto errata specificazione di $G(\cdot)$ vale che

$$P(T_\lambda > \chi_{\dim(\lambda_s)}^2\{1 - \alpha\} | H_0) \rightarrow k \quad n \rightarrow \infty$$

dove $k \in [0, 1]$ non necessariamente uguale ad α . Il test rapporto di verosimiglianza per verificare l'ipotesi di nullità delle varianze degli effetti casuali non è asintoticamente valido. Problemi riguardanti il vero errore di primo tipo in questo genere di test non svaniscono al crescere di n . Si noti comunque che, essendo questo un caso di test sulla frontiera dello spazio parametrico, la statistica test, anche quando il modello è correttamente specificato, non segue una χ^2 e il test rapporto di verosimiglianza non è in generale uno strumento affidabile per verificare ipotesi di questo tipo.

2.3 Strumenti di diagnostica

Gli effetti casuali sono variabili latenti, non osservabili, per cui fare valutazioni sulla bontà di una specificazione per $G(\cdot)$ non è banale. Nel presente paragrafo si presenta una rassegna di alcuni strumenti per la diagnosi di un'errata specificazione della distribuzione degli effetti casuali.

2.3.1 Test di Tchetgen & Coull (2006)

Gli Autori propongono un test per modelli lineari generalizzati con effetti misti basato sulla differenza tra la stima di massima verosimiglianza ottenuta con la verosimiglianza marginale (1.8) e quella ottenuta con la **verosimiglianza condizionata**. Quest'ultima è basata sulla densità condizionata a quella che è la statistica sufficiente per le componenti di varianza quando queste sono trattate come effetti fissi. L'argomento della verosimiglianza condizionata in modelli misti è affrontato nel dettaglio in Sartori & Severini (2004). Lo stimatore che si ottiene con questa seconda funzione di verosimiglianza è meno efficiente, ma robusto rispetto a un'errata specificazione di $G(\cdot)$. Sia β^W il vero valore degli effetti fissi *within-cluster*, $\hat{\beta}_M^W$ la stima ottenuta con la verosimiglianza marginale e $\hat{\beta}_C^W$ la stima ottenuta con la verosimiglianza condizionata. La statistica test è data da

$$D = n\hat{\delta}^T \hat{\Sigma}_\delta^{-1} \hat{\delta} \quad (2.3)$$

dove $\hat{\delta} = \hat{\beta}_C^W - \hat{\beta}_M^W$ e Σ_δ è la varianza asintotica di $\sqrt{n}\hat{\delta}$, che sotto l'ipotesi nulla coincide con $\Sigma_C - \Sigma_M$, matrice di varianze e covarianze dello stimatore di massima verosimiglianza con verosimiglianza condizionata e marginale rispettivamente; Σ_δ è garantita essere semidefinita positiva perchè Σ_M coincide con il limite inferiore di Cramér-Rao. D è dunque una forma quadratica della differenza standardizzata di due stime puntuali distinte per gli effetti fissi *within-cluster*. Sotto l'ipotesi nulla di corretta specificazione,

asintoticamente vale $\delta \sim \chi_{\dim(\delta)}^2$. Si rifiuta l'ipotesi nulla per valori grandi di D . Attraverso simulazioni, gli Autori mostrano che il test possiede una buona capacità di rilevare specificazioni errate, ma è applicabile solo a modelli con funzione legame canonica e con almeno un effetto fisso *within-cluster*.

2.3.2 Test del determinante di Alonso et al. (2008)

È un test basato sugli autovalori della matrice di varianze e covarianze degli stimatori degli effetti fissi. Sia $\theta = (\beta, \psi)$ il vettore dei parametri ignoti in un modello lineare generalizzato con effetti misti. Sia $f(y_i, \theta) = \int \prod_{j=1}^m p(y_{ij}; \beta | u_i) p(u_i; \psi) du_i$ il contributo dell' i -esima unità alla verosimiglianza marginale (1.8). Si consideri il vettore $S(y_i, \theta)$, la funzione punteggio per ogni unità $i = 1, \dots, n$, con generico elemento

$$S_k(y_i, \theta) = \left\{ \frac{\partial \log f(y_i, \theta)}{\partial \theta_k} \right\}, \quad k = 1, \dots, \dim(\theta). \quad (2.4)$$

Il generico elemento della matrice di varianze e covarianze di (2.4) cambiata di segno è dato da

$$A_{kl} = \mathbb{E} \left(\frac{\partial^2 \log f(Y_i, \theta)}{\partial \theta_k \partial \theta_l} \right), \quad k, l = 1, \dots, \dim(\theta),$$

con

$$\hat{A}_{kl} = \frac{1}{n} \sum_{i=1}^n \frac{\partial^2 \log f(y_i, \theta)}{\partial \theta_k \partial \theta_l}, \quad k, l = 1, \dots, \dim(\theta),$$

generico elemento del suo stimatore consistente. Siano A e \hat{A} le matrici corrispondenti.

Sia B una matrice con generico elemento

$$B_{kl} = \mathbb{E} \left(\frac{\partial \log f(Y_i, \theta)}{\partial \theta_k} \frac{\partial \log f(Y_i, \theta)}{\partial \theta_l} \right), \quad k, l = 1, \dots, \dim(\theta),$$

con

$$\hat{B}_{kl} = \frac{1}{n} \sum_{i=1}^n \frac{\partial \log f(y_i, \theta)}{\partial \theta_k} \frac{\partial \log f(y_i, \theta)}{\partial \theta_l}, \quad k, l = 1, \dots, \dim(\theta),$$

generico elemento del suo stimatore consistente. Si osservi che $n\hat{B} = \sum_{i=1}^n S(y_i, \theta) S^T(y_i, \theta)$.

Nel seguito, con \hat{A} e \hat{B} si intendono le matrici stesse calcolate in $\hat{\theta}$, stimatore di massima verosimiglianza di θ . Siano $\hat{\gamma}_{1A}, \dots, \hat{\gamma}_{\dim(\theta)A}$ gli autovalori di $-\hat{A}$. Siano $\hat{\gamma}_{1B}, \dots, \hat{\gamma}_{\dim(\theta)B}$ gli autovalori di \hat{B} . Si considerino ora le quantità

$$\hat{\delta}_1 = \log |\hat{B}[-\hat{A}^{-1}]| \quad (2.5)$$

e

$$\hat{\delta}_2 = |\hat{B}| \cdot |-\hat{A}^{-1}| \quad (2.6)$$

dove $|C|$ è il determinante di una generica matrice C . Sotto corretta specificazione del modello vale

$$\frac{n}{2 \dim(\theta)} (\hat{\delta}_1)^2 \sim \chi_1^2 \quad (2.7)$$

e

$$\frac{n}{2 \dim(\theta)} (\hat{\delta}_1 - 1)^2 \sim \chi_1^2 \quad (2.8)$$

Si può inoltre riscrivere la (2.5) come

$$\hat{\delta}_1 = \sum_{k=1}^{\dim(\theta)} (\log \hat{\gamma}_{kB} - \log \hat{\gamma}_{kA})$$

e la (2.6) come

$$\hat{\delta}_2 = \prod_{k=1}^{\dim(\theta)} \frac{\hat{\gamma}_{kB}}{\hat{\gamma}_{kA}}.$$

In sostanza, le due statistiche test (2.7) e (2.8) rilevano allontanamenti dall'ipotesi $B = -A$ e loro valori grandi suggeriscono di rifiutare l'ipotesi nulla di corretta specificazione. Si sta dunque verificando la validità dell'identità dell'informazione, attraverso il confronto dei due metodi con cui è possibile ottenere la matrice di informazione di Fisher del modello. L'*Information Matrix test* sviluppato da White (1982) è un test analogo spesso utilizzato in ambito econometrico per valutare la correttezza della specificazione di un modello di regressione.

2.3.3 Test della traccia-determinante di Alonso et al. (2008)

Un'altra possibile strada per valutare la distanza dall'ipotesi di uguaglianza tra B e $-A$ proposta dagli Autori incorpora nella statistica test oltre al determinante delle matrici \hat{B} e $-\hat{A}^{-1}$ anche la loro traccia. Si definisca

$$\hat{\delta}_t = \frac{\text{tr}(\hat{B})}{\text{tr}(-\hat{A})} - \frac{|\hat{B}|}{|-\hat{A}|} \quad (2.9)$$

e

$$\hat{\sigma}_t = \sum_{k=1}^{\dim(\theta)} \left(\frac{\hat{\gamma}_{kA}}{\sum_l \hat{\gamma}_{lA}} - 1 \right)^2$$

dove $\text{tr}(C)$ è la traccia di una generica matrice C . Sotto corretta specificazione del modello vale

$$\frac{n(\hat{\delta}_t)^2}{2\hat{\sigma}_t} \sim \chi_1^2$$

Analogamente, possiamo riscrivere (2.9) attraverso gli autovalori di $-\hat{A}$ e \hat{B} con

$$\hat{\delta}_t = \sum_{k=1}^{\dim(\theta)} \left(\frac{\hat{\gamma}_{kB}}{\sum_l \hat{\gamma}_{lA}} \right) - \prod_{k=1}^{\dim(\theta)} \frac{\hat{\gamma}_{kB}}{\hat{\gamma}_{kA}}$$

Proprietà dei test di Alonso et al. (2008)

Gli Autori hanno analizzato attraverso simulazione la potenza dei test proposti in tre scenari di possibile misspecificazione nella struttura dei modelli misti. Nel primo, la vera struttura prevede che gli effetti casuali siano realizzazioni di una mistura di due normali a seconda di una variabile dipendente contenuta nel modello (si pensi ad effetti casuali che provengono da una determinata normale con media nulla e varianza σ_1^2 per le unità del gruppo di trattamento e da un'altra normale con media nulla e varianza σ_2^2 per le unità del gruppo di controllo) e si assume (erroneamente) che provengano da un'unica distribuzione $N \sim (0, \sigma_0^2)$, i tre test mostrano potenza in generale bassa, salvo i casi in cui le due varianze σ_1^2 e σ_2^2 siano molte diverse. Analogamente, nel secondo scenario, nel quale la ragione della misspecificazione è l'ignorare un effetto casuale presente nel modello (si assume solo intercetta casuale quando nella vera struttura dei dati è presente anche un coefficiente angolare casuale) i casi in cui i test rilevano tale misspecificazione sono solo una piccola frazione. Lo scenario in cui si osservano le prestazioni migliori è quello nel quale vi è una struttura autoregressiva per gli effetti casuali non presente nella specificazione che si assume. Si pensi ad esempio a un GLMM per dati binari

$$\text{logit}(P(Y_{ij} = 1)|u_{ij}) = \beta_0 + \beta_1 x_i + \beta_2 t_j + \beta_3 x_i t_j + u_{ij}$$

dove x_i è una variabile dicotomica che distingue le unità del gruppo di controllo da quelle del gruppo di trattamento e t_j è il tempo al quale si è effettuata la misurazione j -esima, per il quale vale $\text{Cov}(u_{ij}, u_{ik}) = \rho^{|t_{ij} - t_{ik}|} \sigma^2$, mentre si assume (erroneamente), $\rho = 1$. I tre test mostrano risultati incoraggianti e portano a un rifiuto dell'ipotesi nulla di corretta specificazione in buona parte dei casi, specie quando σ è sufficientemente grande. Ciò è particolarmente interessante perché il test di Tchetgen & Coull (2006) dato dalla (2.3) non è applicabile in questo caso specifico.

2.3.4 La funzione gradiente di Verbeke & Molenberghs (2013)

Un utile strumento per valutare la bontà di adattamento della *mixing distribution* $G(\cdot)$ è la cosiddetta **funzione gradiente**, costruita a partire dalla derivata direzionale della log-verosimiglianza $l(\cdot)$ in $G(\cdot)$ in una direzione $H(\cdot)$. La funzione gradiente è così definita

$$\Delta(G, u) = \frac{1}{n} \sum_{i=1}^n \frac{f(y_i|u)}{f(y_i|G)} \quad (2.10)$$

dove u è un generico punto del supporto di $G(\cdot)$ e $f(y_i|G)$ è la densità marginale di y_i con la quale otteniamo la verosimiglianza marginale (2.1). $\Delta(G, u)$ può essere interpretato come una media di n rapporti di verosimiglianza, dove ogni rapporto misura quanto è più verosimile che si osservi y_i per l'unità i se il corrispondente effetto casuale u_i è uguale a u rispetto ad essere stato campionato dalla distribuzione $G(\cdot)$.

Sia $\hat{G}(\cdot)$ la stima di $G(\cdot)$ ottenuta massimizzando la (2.1). Se non esiste alcuna specificazione alternativa $H(\cdot)$ in grado di fornire un adattamento migliore di $\hat{G}(\cdot)$ (ovvero nessuna $H(\cdot)$ porta a una verosimiglianza marginale con un massimo maggiore) deve valere che $\Delta(G, u)$ non supera mai 1 ed è esattamente uguale ad 1 in ogni punto del supporto di $G(\cdot)$. Deviazioni da 1 (in qualunque direzione) della funzione gradiente in punti del supporto di $G(\cdot)$ indicano dunque che esiste una distribuzione alternativa $H(\cdot)$ con un adattamento migliore. In tal caso, dal punto di vista grafico la funzione gradiente dà anche un suggerimento su come migliorare la distribuzione degli effetti casuali per un migliore adattamento, scegliendo una distribuzione $H(\cdot)$ che rispetto a $G(\cdot)$ sposti la sua massa di probabilità dalle aree in cui la funzione di gradiente è piccola alle aree in cui è grande. Tuttavia, la funzione gradiente resta di per sé uno strumento grafico, che non permette di svolgere test formali sulle assunzioni del modello. Un test formale basato sulla funzione gradiente è presentato nel paragrafo successivo.

2.3.5 Test di Drikvandi et al. (2017)

Si consideri la quantità

$$T(\hat{\theta}) = \int_{\mathbb{R}^q} (\hat{\Delta}(\hat{G}, b) - 1)^2 d\hat{G}(b)$$

dove θ è lo stimatore di massima verosimiglianza del vettore θ che contiene tutti i parametri ignoti in un GLMM, ottenuto massimizzando la (2.1); $\hat{\Delta}(\hat{G}, b)$ è la stima della funzione gradiente ottenuta sostituendo $\hat{\theta}$ a θ nelle densità $f(y_i|u)$ e $f(Y_i|G)$. Si osservi che $T(\hat{\theta})$ coincide con la misura della distanza di $\hat{\Delta}(\hat{G}, b)$ da 1 secondo il criterio di Cramèr-von Mises. La distribuzione finita sotto l'ipotesi nulla della statistica test non è facilmente ottenibile. Gli Autori propongono dunque tre strade. La prima è di

guardare alla distribuzione asintotica di $T(\hat{\theta})$ sotto H_0 , ottenibile analiticamente. La seconda è quella di usare la statistica test modificata

$$T^* = \frac{T(\hat{\theta})}{\bar{\lambda}}$$

dove $\bar{\lambda}$ è la media di $\hat{\lambda}_1, \dots, \hat{\lambda}_L$, stime degli autovalori di $E^T Q(\theta) E$ (Appendice B), che ha il vantaggio di avere distribuzione approssimata χ_L^2 . La terza è quella di utilizzare un *bootstrap* parametrico in grado di approssimare la distribuzione finita di $T(\hat{\theta})$ sotto H_0 , approccio particolarmente utile quando la dimensione del campione è piccola e quindi risultati asintotici non sono molto affidabili. Attraverso simulazione gli Autori hanno mostrato che T e T^* sono conservativi in termini di errore di primo tipo (hanno probabilità di errore di primo tipo più bassa del livello nominale) e sono più efficaci dei test basati su (2.9) e su (2.3) per rilevare specificazione errate, hanno infatti potenza maggiore nella maggior parte degli scenari considerati.

Capitolo 3

Studio di simulazione

Introduzione

Nella prima parte del capitolo si andranno a presentare i risultati dello studio di simulazione, ricostruzione di quanto fatto da Hui et al. (2021) nel paragrafo 4 dell'articolo, con l'aggiunta di un'ulteriore distribuzione, una mistura di normali, per gli effetti casuali. Nella seconda parte si andrà invece a implementare con software R la funzione gradiente di Verbeke & Molenberghs (2013) e la statistica test di Drikvandi et al. (2017), entrambe implementate dagli Autori attraverso SAS, su alcuni *dataset* presi dallo studio di simulazione della prima parte del capitolo.

3.1 Disegno dello studio

Il modello su cui si basa lo studio di simulazione, che Hui et al. (2021) hanno a loro volta ripreso da Neuhaus et al. (2013), è un modello lineare normale con effetti misti, definito in (1.4), nel quale X_i è una matrice $m \times 3$ e Z_i una matrice $m \times 2$. Si hanno dunque tre effetti fissi e due effetti casuali. La prima colonna di X_i è un vettore unitario e rappresenta il termine di intercetta, la seconda colonna è data da valori equispaziati tra -1 e 1 mentre la terza da valori generati da una variabile casuale uniforme sull'intervallo $[-1, 1]$; Z_i coincide con le prime due colonne di X_i . Il numero di unità n è un valore preso dall'insieme $\{50, 100\}$ e il numero di osservazioni per ciascuna unità m un valore preso dall'insieme $\{10, 20\}$. Il vero valore degli effetti fissi è imposto essere $\beta_0 = 2$, $\beta_1 = 1$, $\beta_2 = 1$, mentre per la varianza *within-cluster* si assume $\sigma_\epsilon^2 = 1$ e indipendenza tra errori casuali relativi a osservazioni distinte sulla stessa unità. I due effetti casuali sono generati indipendentemente e, come vera distribuzione, sono scelte sette distribuzioni possibili:

la normale (si ha dunque in questo caso assenza di misspecificazione) (1); la t di Student con 3 gradi di libertà (2); la Tukey gh con $g = 0.25$ e $h = 0.05$ (si veda l'Appendice C per una panoramica sulla distribuzione Tukey gh) (3); la Tukey gh con $g = 0.5$ e $h = 0.05$ (4); la log-normale con media e varianza del logaritmo rispettivamente 0 e 1 (5); la chi quadrato con 3 gradi di libertà (6); la beta con $\alpha = 2$ e $\beta = 4$ (7). Nel presente studio oltre a queste sette distribuzioni si è considerato anche una mistura di due normali, una con media 2 e l'altra con media -2 , con uguale peso (8). È stata presa in esame questa ulteriore distribuzione in modo da poter verificare cosa succede quando $G(\cdot)$ è bimodale. Tutte le distribuzioni sono state trasformate in modo da avere media nulla e vera deviazione standard che assume valori nell'insieme $\{0.5, 1, \sqrt{2}, 2, 3\}$. Le densità sono rappresentate nella Figura 3.1 in forma grafica, nella loro versione standardizzata, con media nulla e varianza unitaria. Si può osservare che sono distribuzioni che coprono un'ampia gamma di forme, e si distinguono per simmetria, curtosi e supporto.

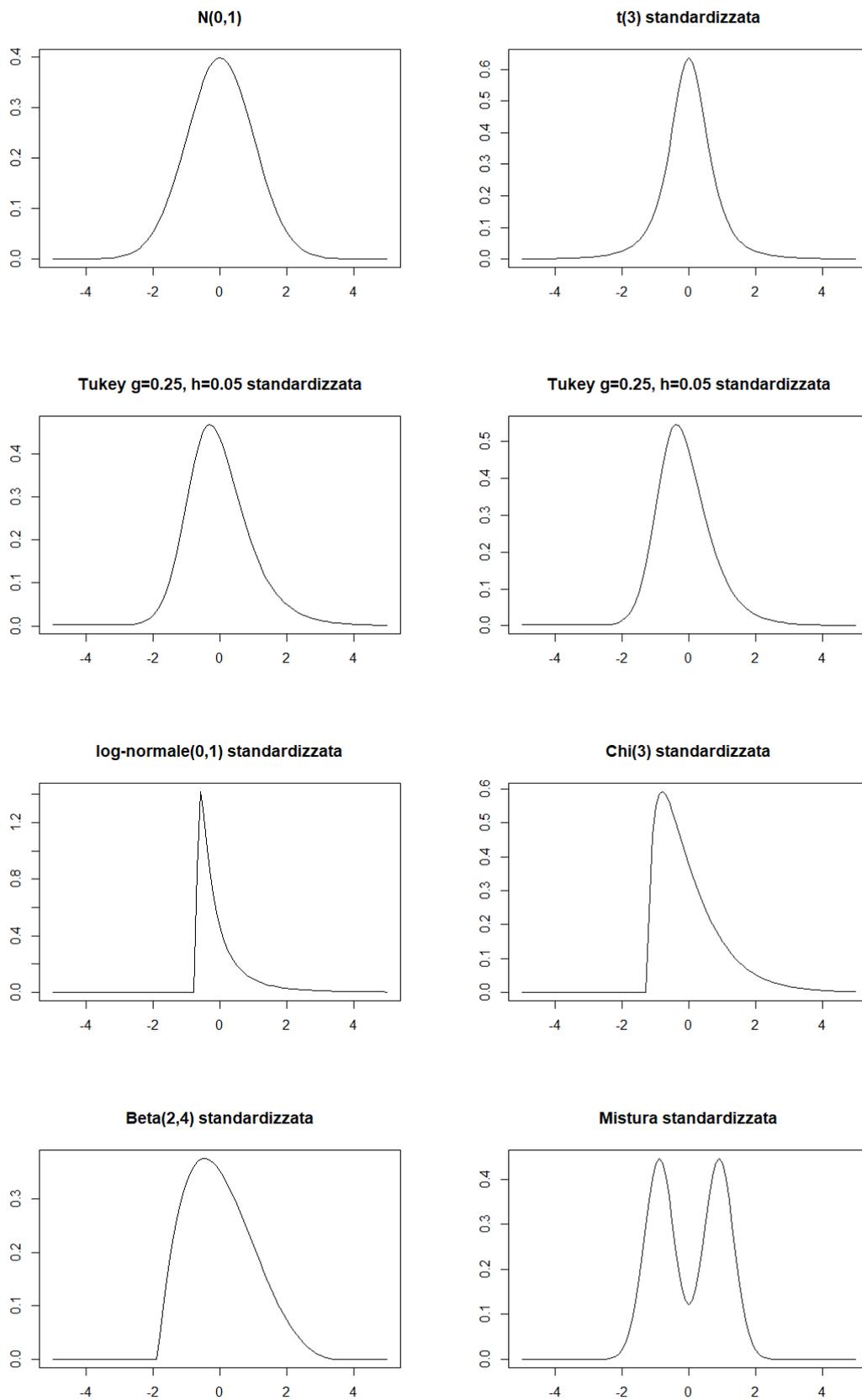


FIGURA 3.1: Densità delle distribuzioni standardizzate

Fissati n e m , sono stati generati 500 *dataset* per ogni combinazione di distribuzione degli effetti casuali e vero valore della componente di deviazione standard, chiamata c_0 , e su ognuno di questi è stato stimato un LMM attraverso la funzione `lme`, che assume normalità per gli effetti casuali. Le stime dei parametri sono state ottenute attraverso massimizzazione della verosimiglianza marginale (1.8), invece che con la REML, metodo di default della funzione, in modo da avere risultati coerenti con quelli degli Autori. Per ogni combinazione è poi stata calcolata la distorsione empirica, definita come la media campionaria delle stime meno il vero valore, e la relativa copertura effettiva degli intervalli di confidenza, definita come la proporzione di volte in cui l'intervallo di confidenza contiene il vero valore del parametro a un livello nominale del 95% per i tre effetti fissi, β_0 , β_1 e β_2 , e per le due componenti di deviazione standard, chiamate λ_1 e λ_2 . Per gli effetti fissi, sono stati considerati intervalli di confidenza di Wald, mentre per le componenti di deviazione standard gli intervalli sono stati costruiti attraverso la funzione `intervals`, contenuta nella libreria `nlme`. Quest'ultima calcola anch'essa intervalli di confidenza di Wald, ma a partire dalla verosimiglianza ristretta. È stata fatta questa scelta a causa della difficoltà a reperire con R gli *standard error* per le componenti di deviazione standard basandosi sulla verosimiglianza marginale. Il codice R relativo alla simulazione è contenuto nell'Appendice E.1.

3.2 Risultati dello studio

Di seguito sono riportati i risultati dello studio quando $n = 50$ e $m = 10$. I grafici per le altre combinazioni di n e m sono contenuti nell'Appendice D.1. Nel caso presente, quando c_0 è fissato a valori piccoli, specialmente 0.5 e 1, ma occasionalmente anche con valori più grandi, si possono avere *dataset* per i quali, per problemi di convergenza, non è possibile trovare le stime dei parametri oppure si riescono a trovare le stime ma `intervals` non riesce a trovare gli estremi degli intervalli di confidenza per le componenti di deviazione standard. Nel calcolo delle distorsioni e delle coperture effettive, tali *dataset* sono stati ignorati. Il problema tende a scomparire all'aumentare di c_0 . Il numero di *dataset* problematici è riportato nella Tabella 3.1.

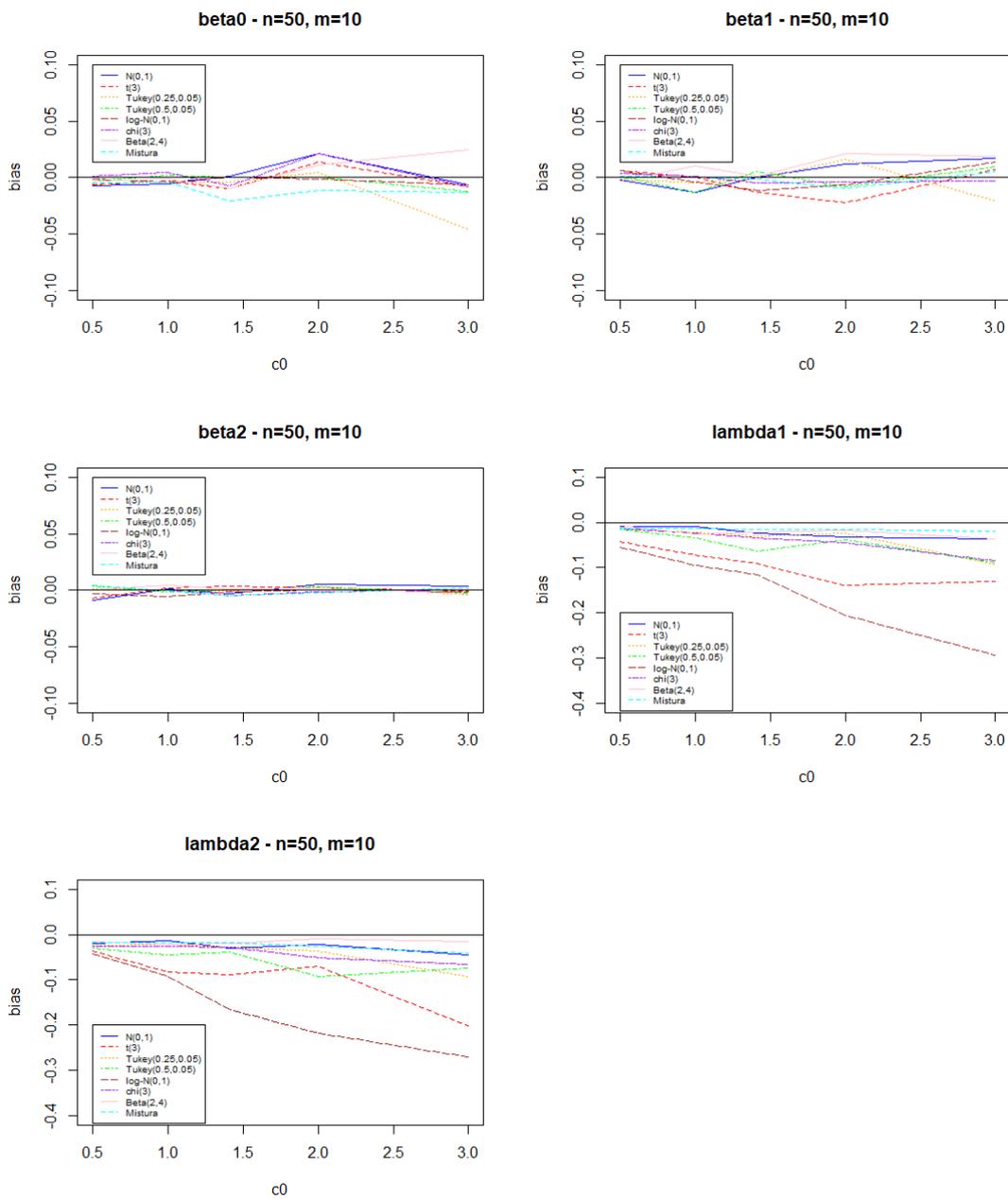


FIGURA 3.2: Distorsioni empiriche nel caso $n = 50$ e $m = 10$.

Si osserva una moderata distorsione per l'intercetta fissa, inferiore al 2% del vero valore del parametro, e una distorsione di entità leggermente inferiore per β_1 . Man mano che aumenta la deviazione standard della distribuzione degli effetti casuali, si osserva in una buona parte dei casi un andamento non monotono per la distorsione. Questa caratteristica è presente anche in Hui et al. (2021), mentre su ulteriori simulazioni compiute prendendo un numero di *dataset* per ciascuna combinazione di distribuzione e vera componente di deviazione standard più grande questo fatto è leggermente meno marcato, per cui si è portati a pensare che ciò possa dipendere dalla limitatezza del

numero di simulazioni. I grafici delle distorsioni empiriche quando il numero di *dataset* per ogni combinazione è fissato a 3000 invece di 500, per il caso $n = 50$ e $m = 10$, sono contenuti nell'appendice D.1. Si può notare inoltre che non vi sono grandi differenze tra le distorsioni sotto corretta specificazione del modello, date dalle linee blu nei grafici, e le altre linee, specialmente per quanto riguarda β_1 , e ciò porta a pensare che non sia un'errata specificazione di $G(\cdot)$ la causa principale di tale distorsione. Per β_2 , effetto fisso a cui non è associato alcun effetto casuale, si hanno invece distorsioni prossime a zero.

Non vale lo stesso per λ_1 e λ_2 . Ad eccezione della distribuzione beta e della mistura di normali, che hanno distorsione comparabile a quella che si osserva sotto corretta specificazione del modello, le restanti producono stime negativamente distorte in maniera piuttosto considerevole, specialmente nel caso della t di Student e della log-normale, con quest'ultima che arriva ad avere una distorsione del $\approx -30\%$ per $c_0 = 3$.

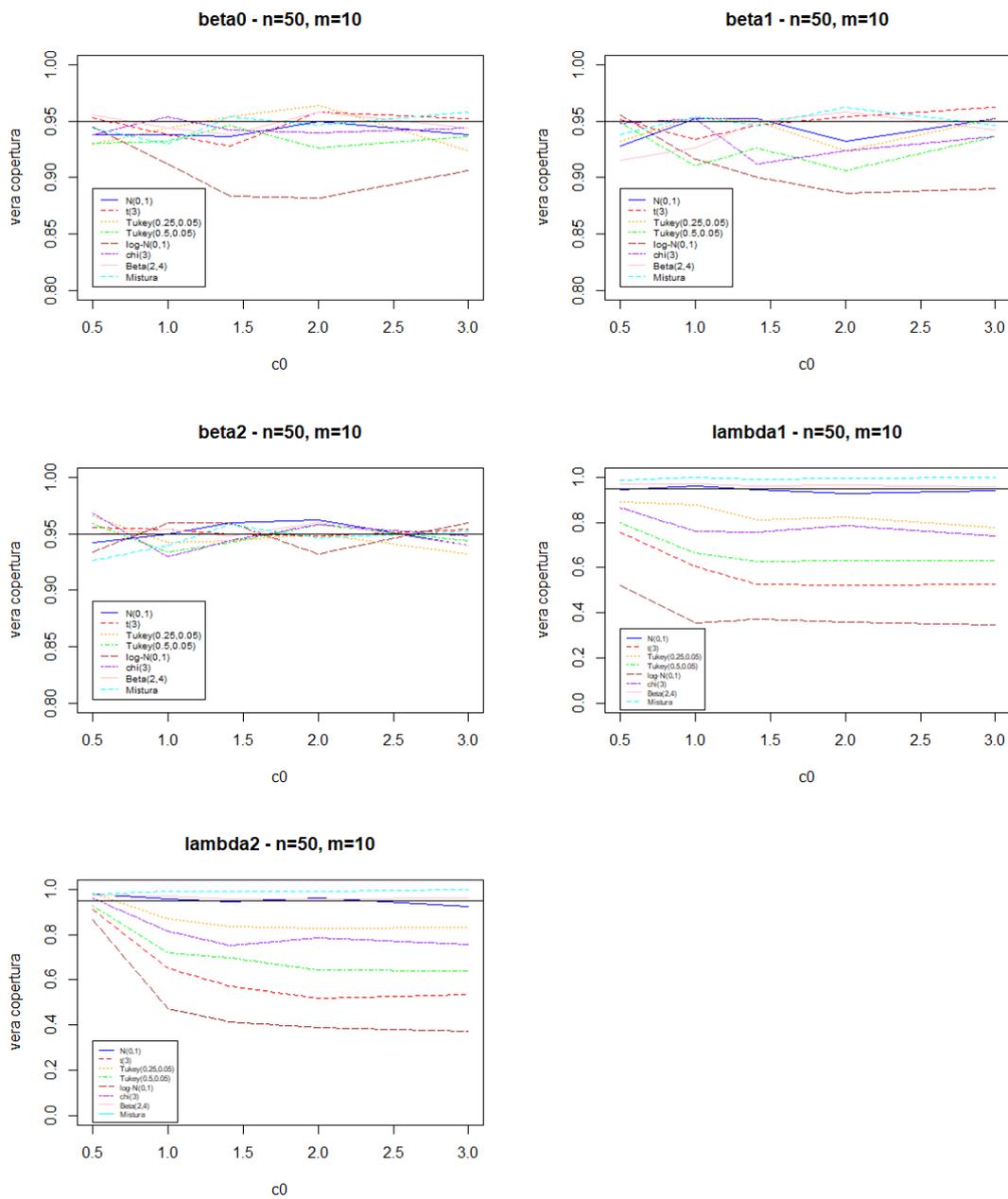


FIGURA 3.3: Vere coperture degli intervalli di confidenza nel caso $n = 50$ e $m = 10$.

Per quanto riguarda gli intervalli di confidenza, i risultati sono riassunti nella Figura 3.3. Si hanno coperture vicine al livello nominale del 95% per β_0 e β_1 ; con l'unica eccezione della distribuzione log-normale, infatti, le restanti hanno livello reale $> 90\%$. Non sembrano esserci particolari problemi riguardo la copertura effettiva degli intervalli di confidenza per l'effetto fisso β_2 . Vi è invece considerevole sottocopertura per le componenti di deviazione standard. Escluse la beta e la mistura di normali, le restanti producono intervalli di confidenza che contengono il vero valore del parametro in una frazione di volte di molto inferiore rispetto al livello nominale, che arriva al $\approx 40\%$

nel caso della log-normale quando $c_0 = 3$. Sotto errata specificazione della distribuzione degli effetti normali, le procedure inferenziali sulle componenti di varianza possono essere pesantemente compromesse, come evidenziato anche da Hui et al. (2021).

Distribuzione	stime - 0.5	I.C. - 0.5	stime - 1	I.C. - 1	stime - $\sqrt{2}$	I.C. - $\sqrt{2}$
(1)	0	8	0	4	0	1
(2)	15	31	0	6	0	1
(3)	4	16	0	4	0	1
(4)	9	21	0	3	0	1
(5)	23	46	0	7	0	5
(6)	7	16	0	4	0	2
(7)	0	10	0	4	0	0
(8)	0	16	0	2	0	0

Distribuzione	stime - 2	I.C. - 2	stime - 3	I.C. - 3
(1)	0	0	0	1
(2)	0	1	0	0
(3)	0	1	0	0
(4)	0	1	0	1
(5)	0	1	0	1
(6)	0	1	0	2
(7)	0	0	0	0
(8)	0	0	0	0

TABELLA 3.1: Numero di *dataset*, su un totale di 500, per i quali non vi è convergenza nel calcolo delle stime o degli estremi degli intervalli di confidenza delle componenti di deviazione standard nel caso $n = 50$, $m = 10$.

3.3 Implementazione della funzione gradiente

Si illustra ora un esempio di calcolo con R della funzione gradiente di Verbeke & Molenberghs (2013), definita in (2.10), vista come funzione di u con $G(\cdot)$ fissata e assunta essere una normale, su *dataset* simulati a partire dal LMM del paragrafo precedente, considerando però un unico effetto casuale, per il termine di intercetta. Una tale scelta rende più immediate l'implementazione e le analisi grafiche. Le vere distribuzioni $G(\cdot)$ prese in considerazione per la generazione delle intercette casuali sono la normale standard e la log-normale con media e varianza del logaritmo rispettivamente 0 e 1, standardizzata. Quest'ultima distribuzione ha mostrato nello studio di simulazione precedente di portare ad avere, assumendo erroneamente l'ipotesi di normalità, modelli lineari misti con criticità riguardo l'inferenza sulle componenti di varianza e, in parte

minore, anche sugli effetti fissi. L'obiettivo che ci si prepone è verificare se la funzione gradiente sia in grado in questi casi di fornire segnali riguardo una possibile errata specificazione della distribuzione degli effetti casuali nel modello.

Le quantità necessarie sono le densità condizionate e marginali. In questo caso specifico, entrambe queste densità sono delle normali multivariate. Si è quindi sfruttata la funzione `dmvnorm` della libreria `mvtnorm` (Genz et al., 2021). Il codice è contenuto nell'Appendice E.2.

Funzione gradiente nel caso di corretta specificazione

Quando le intercette casuali sono effettivamente generate da una normale, si può osservare nella Figura 3.4 una funzione gradiente che oscilla attorno alla retta di ordinata uno parallela all'asse delle ascisse u , segnale di una corretta specificazione del modello. Anche provando a simulare più volte, si tende a ottenere sempre un risultato simile. Per valori di u indicativamente maggiori in modulo di due, la funzione gradiente, invece, va talvolta verso lo zero e talvolta assume valori grandi. Tuttavia è utile ricordare che la vera distribuzione, che è una normale standard, ha densità molto basse per quei valori. Oltre il 95% della sua massa di probabilità, infatti, è contenuta nell'intervallo $[-2, 2]$. Valutazioni su una corretta specificazione del modello devono basarsi quindi principalmente su questo sottointervallo del supporto di $G(\cdot)$.

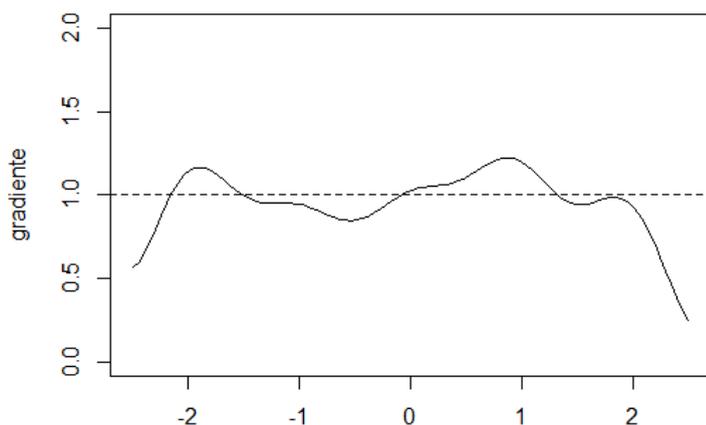


FIGURA 3.4: Funzione gradiente in un LMM stimato su dati simulati in cui l'intercetta casuale è generata da una normale standard ($n = 50, m = 10$).

Funzione gradiente nel caso di errata specificazione

Quando invece le intercette casuali sono generate dalla log-normale standardizzata e il modello è stimato sotto ipotesi di normalità, si può osservare nella Figura 3.5 che la funzione gradiente ottenuta mostra ampi allontanamenti da uno anche all'interno dell'intervallo $[-2, 2]$. Ciò costituisce un allarme riguardo la specificazione assunta per l'effetto casuale. Inoltre, il grafico fornisce un suggerimento su una specificazione alternativa che dia un adattamento migliore. La forma della funzione consiglia infatti di considerare una distribuzione che, rispetto alla normale assunta, abbia maggiore massa di probabilità attorno a -0.5 e minore massa di probabilità in $[-2, -1]$ e $[1, 2]$.

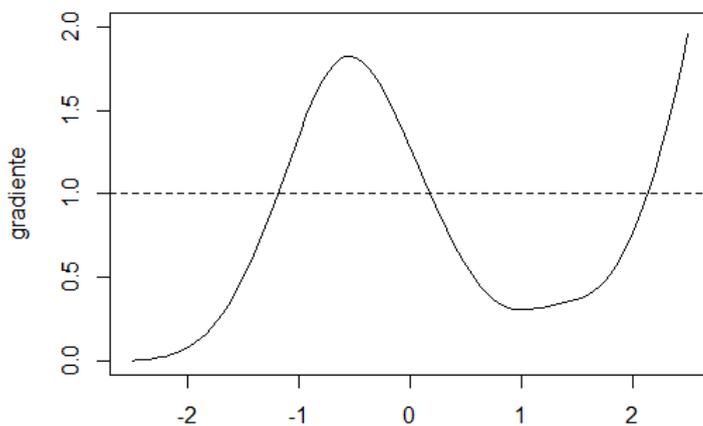


FIGURA 3.5: Funzione gradiente in un LMM simulato con intercetta casuale generata da una log-normale(0,1) standardizzata ($n=50$, $m=10$)

Andando a confrontare le densità della distribuzione assunta e della vera distribuzione, osservabili nella Figura 3.6, è effettivamente ciò che si dovrebbe fare.

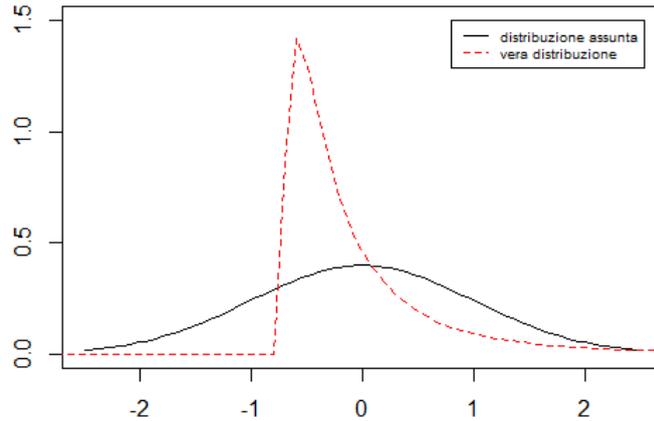


FIGURA 3.6: Densità di una normale (0,1) e di una log-normale (0,1) standardizzata

3.4 Implementazione della statistica di Drikvandi et al. (2017)

Si illustra nel presente paragrafo un'implementazione con R del calcolo della statistica test (2.3.5). Il relativo codice è contenuto nell'Appendice E.3.

L'integrale è approssimato con approccio quasi-Monte Carlo

$$T(\hat{\theta}) = \int (\hat{\Delta}(\hat{G}, b) - 1)^2 d\hat{G}(b) \approx \frac{1}{K} \sum_{k=1}^K (\hat{\Delta}(\hat{G}, b_k) - 1)^2$$

dove K è il numero di nodi di integrazione e $b_k = F^{-1}(c_k)$, in cui c_k , $k = 1, \dots, K$ sono i nodi di integrazione e $F^{-1}(\cdot)$ è l'inversa della funzione di ripartizione di $G(\cdot)$. I nodi c_k sono dati da

$$c_k = \frac{2k-1}{2K} \quad k = 1, \dots, K.$$

I valori osservati per la statistica test sono 0.349 nel primo caso, in cui si assume correttamente che gli effetti casuali siano generati da una normale, e 0.590 nel secondo, in cui la vera distribuzione è la log-normale mentre si assume, erroneamente, normalità.

I risultati confermano le aspettative: la statistica test, che è una misura della distanza da 1 della funzione gradiente, mostra un valore più alto nel caso in cui vi è misspecificazione.

Conclusione

In questa relazione si è affrontato il problematico tema della specificazione della distribuzione degli effetti casuali in modelli misti. È stata presentata una rassegna delle conclusioni a cui sono giunti vari autori della letteratura scientifica contemporanea occupatisi dell'argomento. Sono stati descritti alcuni strumenti di diagnostica utili a valutare la correttezza dell'ipotesi assunta. È stato riproposto lo studio di simulazione di Hui et al. (2021), con una distribuzione aggiuntiva tra quelle considerate. È stata infine presentata un'implementazione con R della funzione gradiente di Verbeke & Molenberghs (2013), e della relativa statistica test di Drikvandi et al. (2017). Per quest'ultima, si è solo calcolato il valore osservato in due modelli simulati, uno in assenza e uno in presenza di misspecificazione. Al fine di svolgere procedure formali di test, sarebbe interessante applicare il *bootstrap* parametrico descritto dagli Autori ai modelli in questione, in modo da avere una misura della significatività dei valori osservati. Si segnala tuttavia che un tale *bootstrap* avrebbe un costo computazionale piuttosto oneroso.

Si può concludere che la specificazione della distribuzione degli effetti casuali sia un problema quantomeno da tenere in considerazione nella stima dei modelli misti. Si è avuto modo di osservare che l'impatto sull'inferenza sugli effetti fissi è nella maggior parte dei casi di lieve entità; quando le componenti di varianza sono un elemento di interesse, tuttavia, un'errata specificazione può comprometterne notevolmente l'inferenza.

Potrebbe essere utile in futuro l'inclusione in software di utilizzo comune come R di funzioni per la stima di modelli misti che permettano una maggiore flessibilità per la specificazione della distribuzione degli effetti casuali. Una mistura di normali, ad esempio, permetterebbe di catturare un'ampia gamma di forme possibili per $G(\cdot)$.

Appendice A

Distribuzioni condizionate in una normale multivariata

Sia Y un vettore casuale n -dimensionale la cui distribuzione segue una normale multivariata con valore atteso il vettore μ e matrice di varianze e covarianze Σ . Si consideri la partizione

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$$
$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$
$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix},$$

dove Y_1 è un vettore p -dimensionale e Y_2 è un vettore $q = n - p$ dimensionale, con entrambi distribuzione normale multivariata. $(Y_1|Y_2)$ si distribuisce come una normale p -variata con media $\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(Y_2 - \mu_2)$ e matrice di varianze e covarianze data da $\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$. Per la verifica, si definisca il vettore p -dimensionale Z come

$$Z = Y_1 + AY_2$$

dove $A = -\Sigma_{12}\Sigma_{22}^{-1}$. Z in quanto combinazione di vettori che seguono una normale multivariata, ha anch'esso distribuzione normale multivariata. Si osservi che

$$Cov(Z, Y_2) = Cov(Y_1, Y_2) + Cov(AY_2, Y_2) = \Sigma_{12} + A\Sigma_{22} = \Sigma_{12} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22} = 0$$

ed essendo la distribuzione congiunta di Z e Y_2 una normale multivariata, l'incorrelazione implica anche l'indipendenza. Vale dunque

$$\begin{aligned}
\mathbb{E}(Y_1|Y_2) &= \mathbb{E}(Z - AY_2|Y_2) \\
&= \mathbb{E}(Z|Y_2) - \mathbb{E}(AY_2|Y_2) \\
&= \mathbb{E}(Z) - AY_2 \\
&= \mu_1 + A(\mu_2 - Y_2) \\
&= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(Y_2 - \mu_2)
\end{aligned}$$

e

$$\begin{aligned}
\text{Var}(Y_1|Y_2) &= \text{Var}(Z - AY_2|Y_2) \\
&= \text{Var}(Z|Y_2) + \text{Var}(AY_2|Y_2) - ACov(Z, -Y_2) - Cov(Z, -Y_2)A^T \\
&= \text{Var}(Z|Y_2) \\
&= \text{Var}(Z) \\
&= \text{Var}(Y_1 + AY_2) \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}
\end{aligned}$$

Appendice B

Autovalori necessari al calcolo della distribuzione asintotica del test di Drikvandi et al. (2017)

Si consideri la matrice $E^T Q E$, dove E è la radice quadrata dell'inversa della matrice di informazione di Fisher del modello, mentre Q è la matrice $L \times L$, dove L è la lunghezza del parametro del modello θ , con generico elemento

$$Q_{kl} = \int \left\{ \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbb{E} \left[\frac{\partial}{\partial \theta_k} \frac{f_i(y_i|u)}{\partial f_i(y_i|G)} \right] \right\} \left\{ \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbb{E} \left[\frac{\partial}{\partial \theta_l} \frac{f_i(y_i|u)}{f_i(y_i|G)} \right] \right\} dG(u)$$

$k, l = 1, \dots, L$

il cui stimatore consistente è dato da

$$\hat{Q}_{kl} = \int \left\{ \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \theta_k} \frac{f_i(y_i|u)}{\partial f_i(y_i|G)} \Big|_{\theta=\hat{\theta}} \right\} \left\{ \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \theta_l} \frac{f_i(y_i|u)}{f_i(y_i|G)} \Big|_{\theta=\hat{\theta}} \right\} d\hat{G}(u)$$

$k, l = 1, \dots, L$.

La matrice $E^T Q E$ ha generico autovalore λ_i . Un'approssimazione di \hat{Q} si può ottenere con un'integrazione con approccio Quasi-Monte Carlo attraverso

$$\hat{Q}_{kl} \approx \frac{1}{KN^2} \sum_{s=1}^S \sum_{i=1}^N \sum_{i'=1}^N \left(\frac{\partial}{\partial \theta_k} \frac{f_i(y_i|u_s)}{f_i(y_i|G)} \Big|_{\theta=\hat{\theta}} \right) \left(\frac{\partial}{\partial \theta_l} \frac{f_{i'}(y_{i'}|u_s)}{f_{i'}(y_{i'}|G)} \Big|_{\theta=\hat{\theta}} \right) \quad k, l = 1, \dots, L$$

dove S è il numero di nodi di integrazione.

Appendice C

Distribuzione Tukey gh

La famiglia delle Tukey gh è una classe di distribuzioni, introdotte per la prima volta da Tukey (1977), basata su una trasformazione di una variabile normale standard che permette di avere un ampio *range* di possibilità per quanto riguarda asimmetria e curtosi, governate dai due parametri g e h , rispettivamente. g può assumere valori sull'intero asse reale mentre h può assumere solo valori positivi. L'algoritmo per la generazione di valori da una Tukey gh è facilmente ottenibile, mentre è difficile ricavare la funzione di densità in forma esplicita (He & Raghunathan, 2006).

Sia Z una variabile normale standard. Sia Y una variabile con valore atteso μ e deviazione standard σ la cui distribuzione è data da una Tukey gh con generici parametri g e h . Y è definita dalla seguente trasformazione di Z

$$Y(Z) = \mu + \sigma \frac{e^{gZ} - 1}{g} e^{hZ^2/2}.$$

Due casi speciali della Tukey gh sono la normale standard, ottenuta con $g = 0$ e $h = 0$ e la log-normale con log-valore atteso 0 e log-varianza 1, ottenuta con $g = 1$ e $h = 0$. Di seguito, nella Figura C.1, sono riportate alcune delle possibili forme che si possono ottenere per la funzione di densità, ciascuna nella versione standardizzata a valore atteso nullo e varianza unitaria.

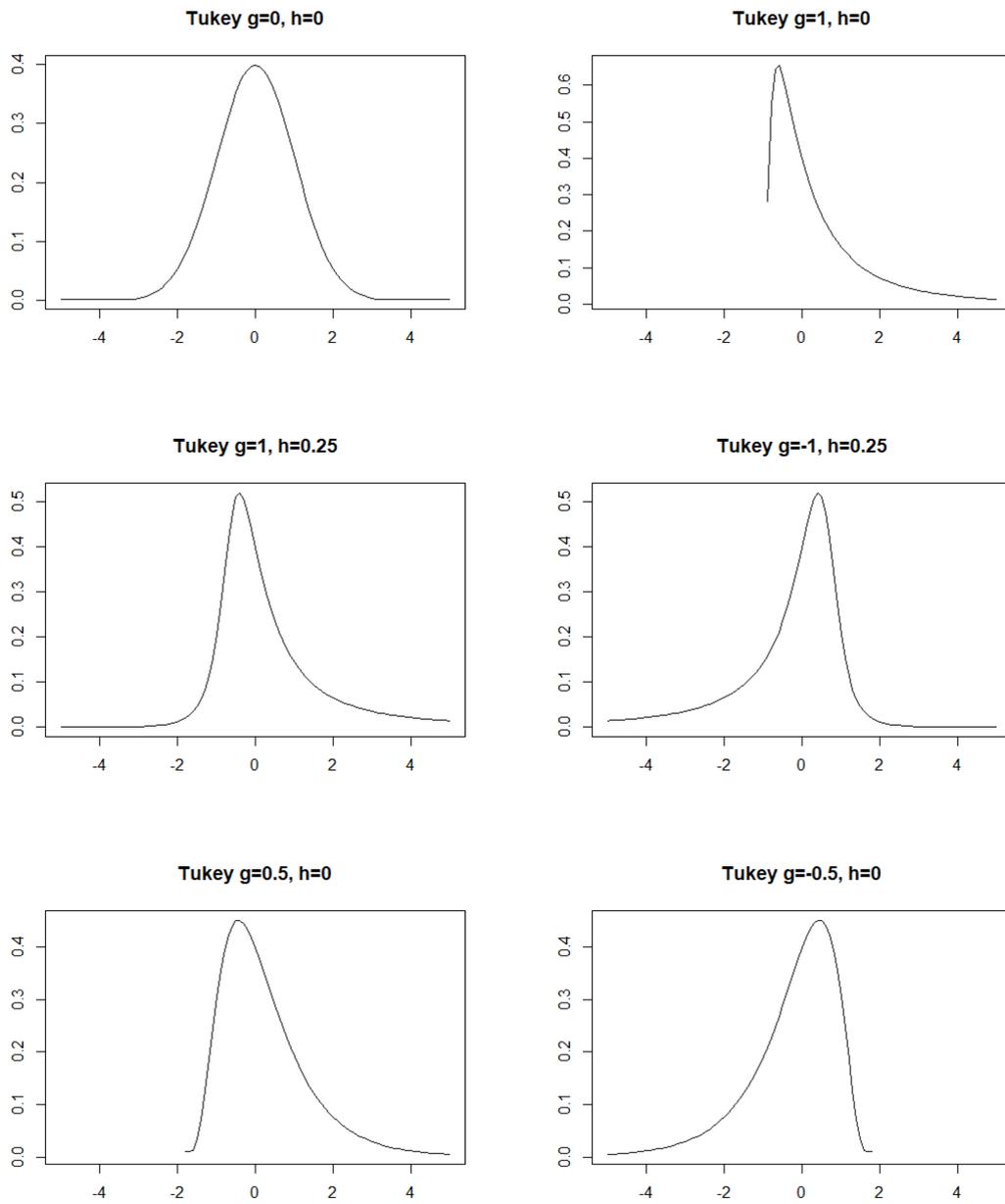
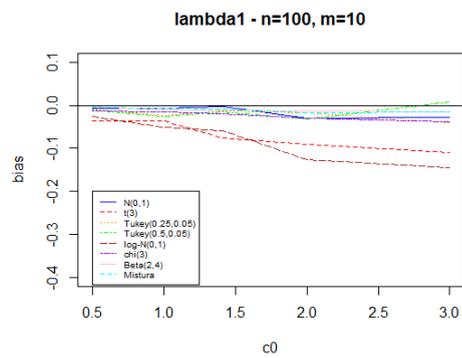
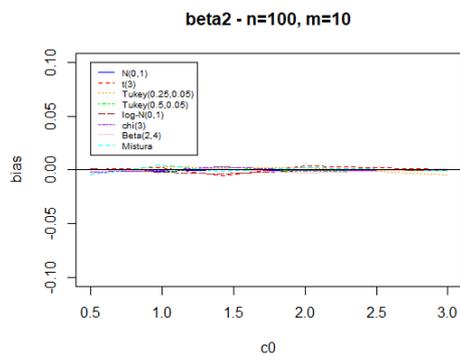
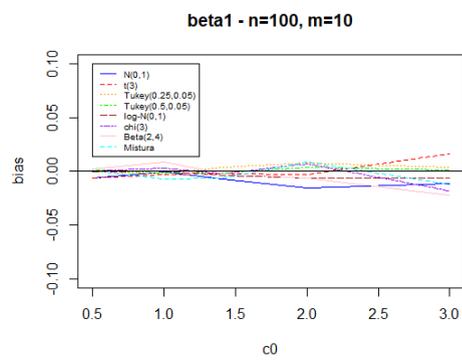
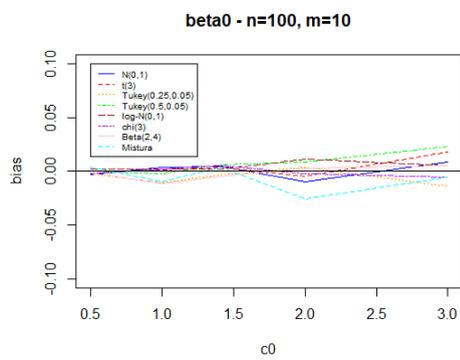


FIGURA C.1: Esempi di densità di Tukey gh standardizzate.

Appendice D

D.1 Grafici



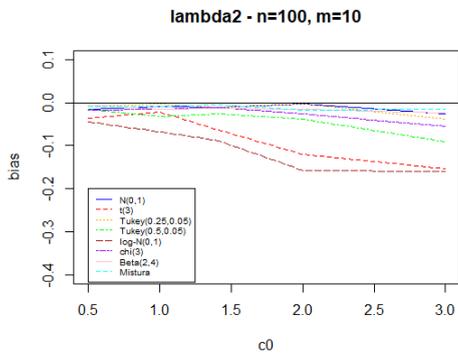


FIGURA D.1: Distorsioni empiriche nel caso $n = 100$ e $m = 10$.

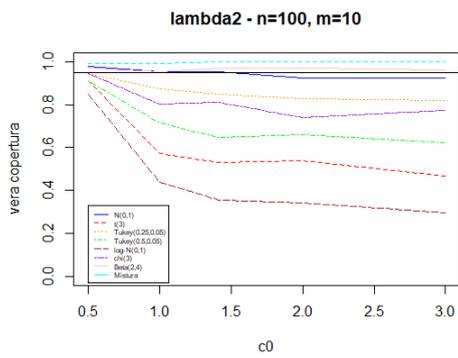
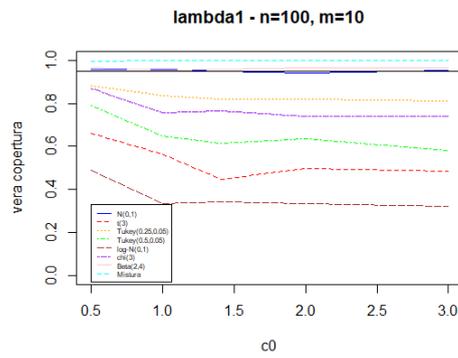
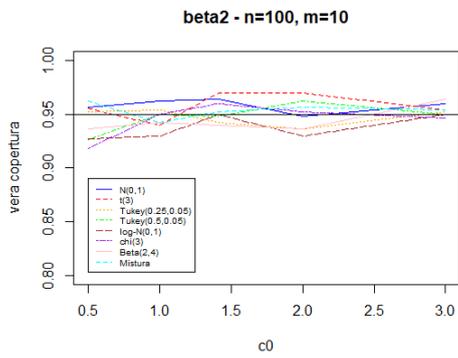
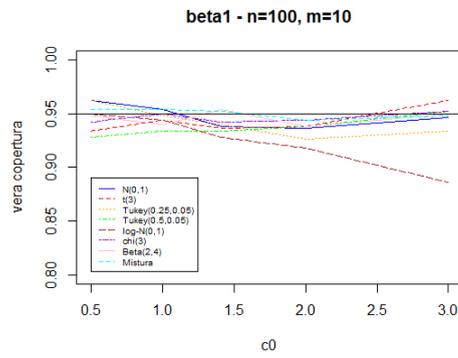
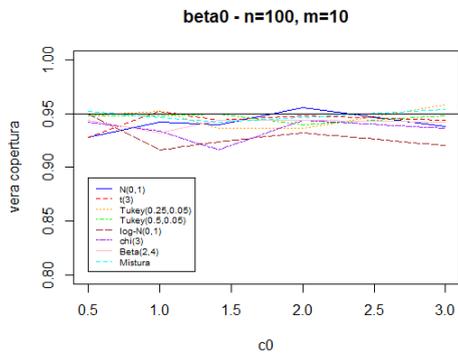


FIGURA D.2: Vere coperture degli intervalli di confidenza nel caso $n = 100$ e $m = 10$.

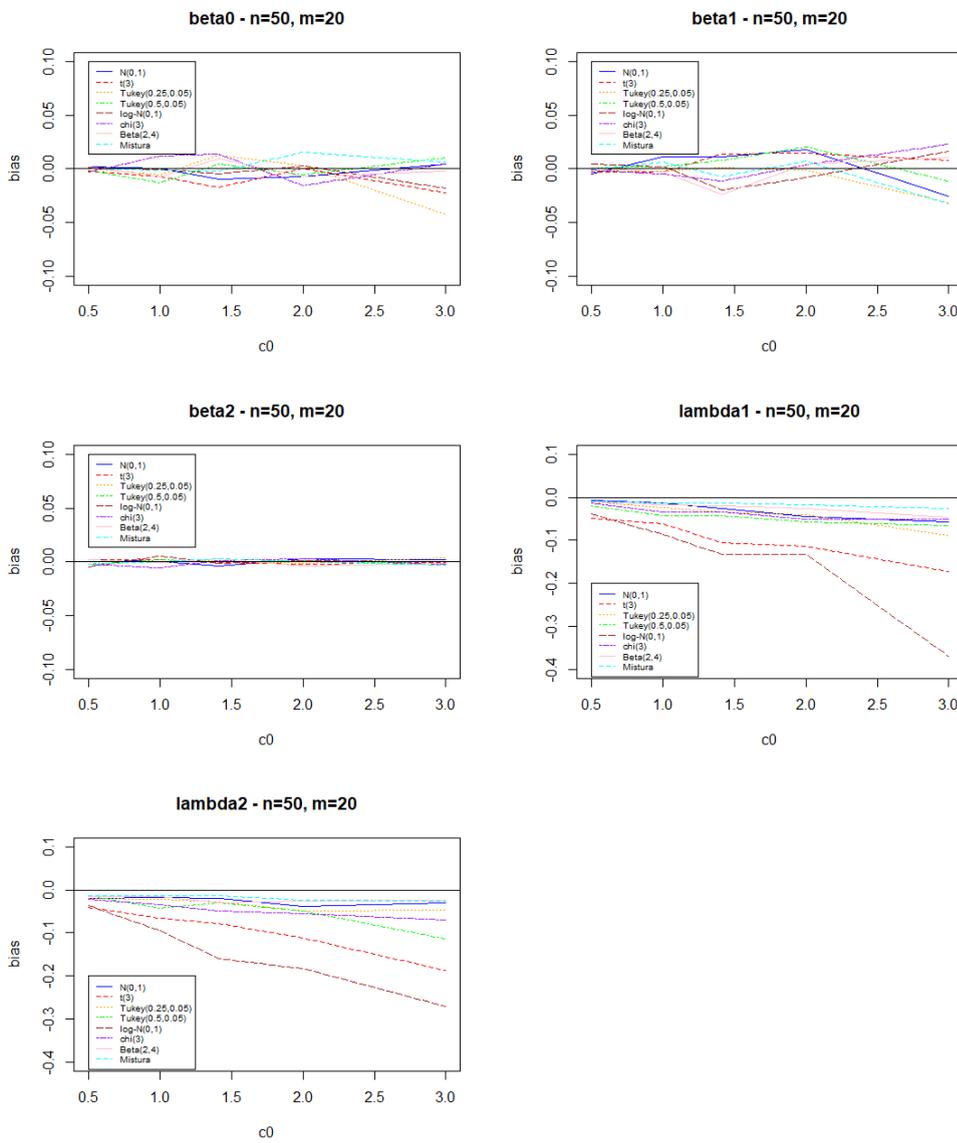


FIGURA D.3: Distorsioni empiriche nel caso $n = 50$ e $m = 20$.

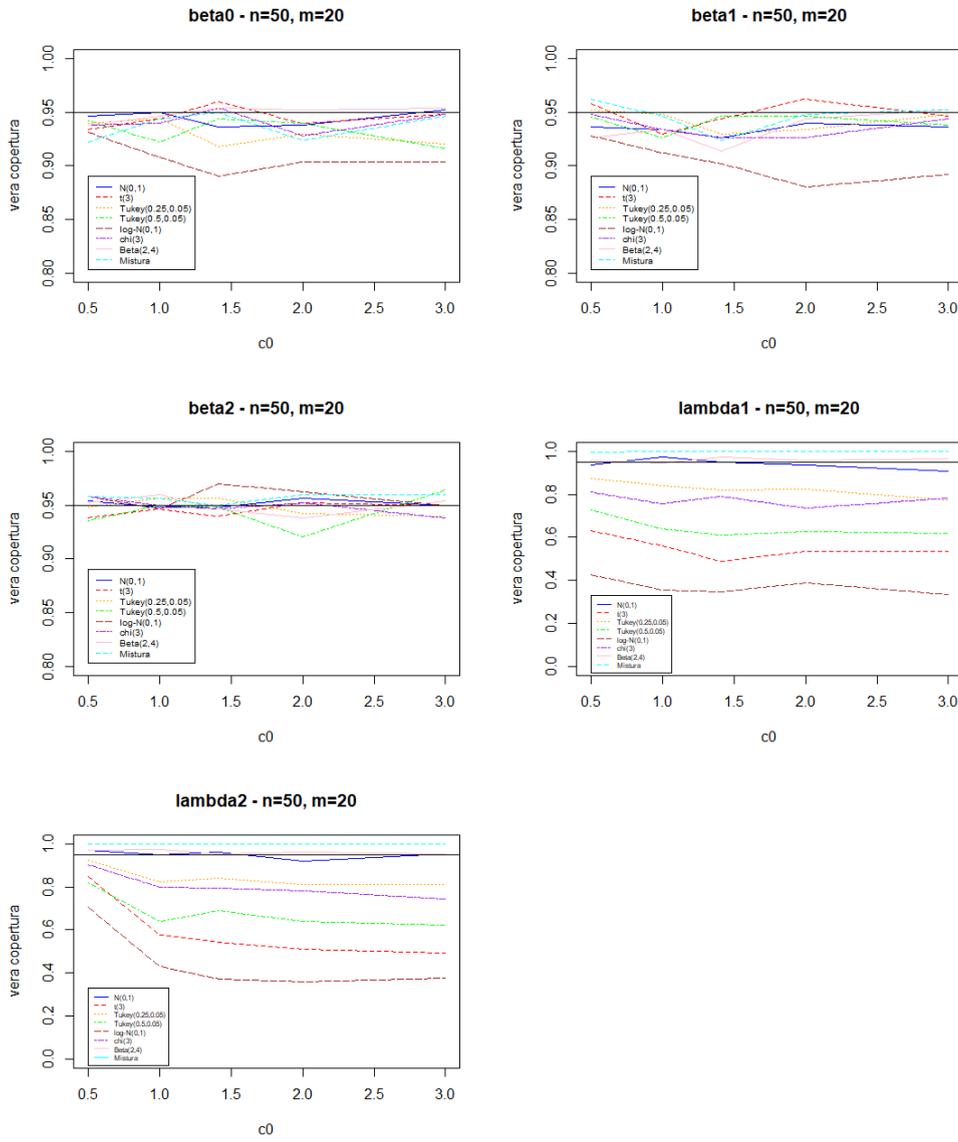


FIGURA D.4: Vere coperture degli intervalli di confidenza nel caso $n = 50$ e $m = 20$.

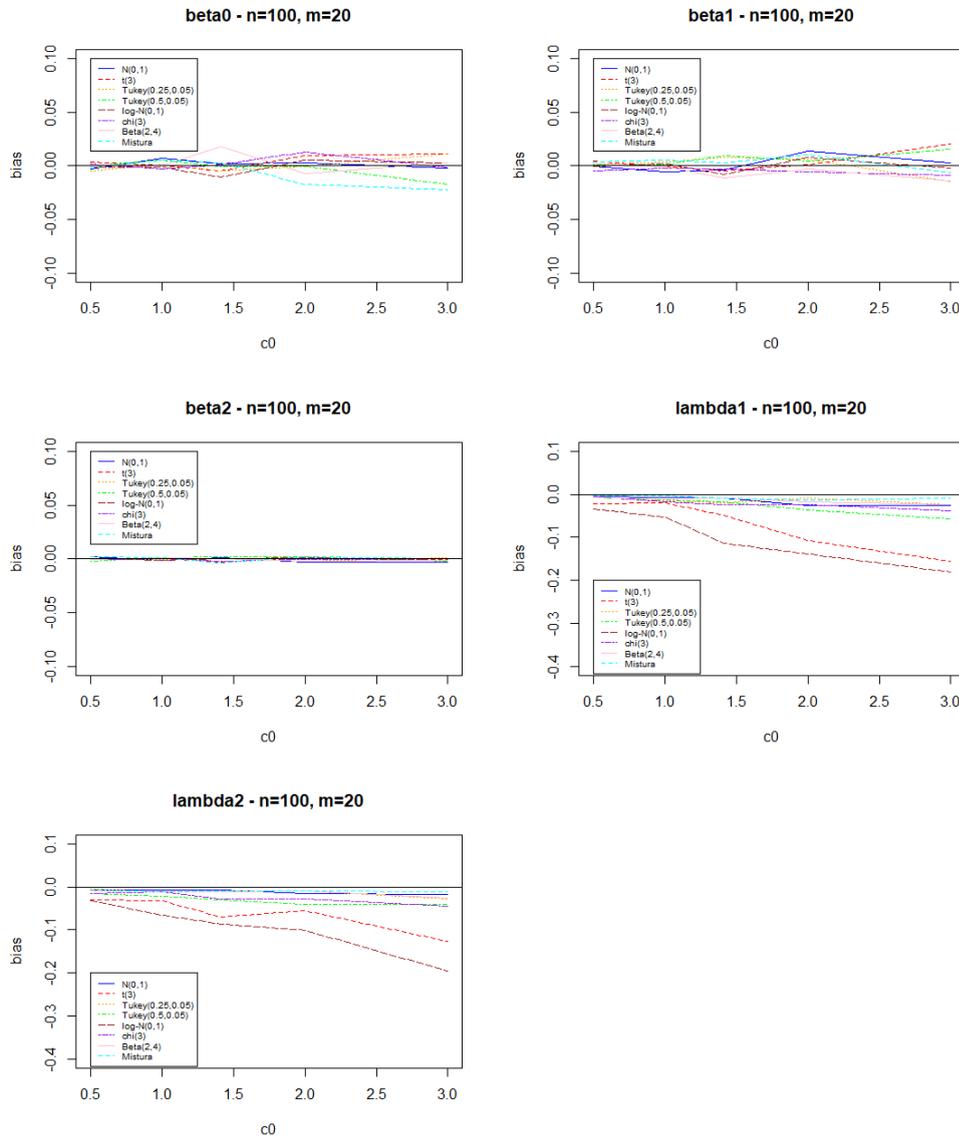


FIGURA D.5: Distorsioni empiriche nel caso $n = 100$ e $m = 20$.

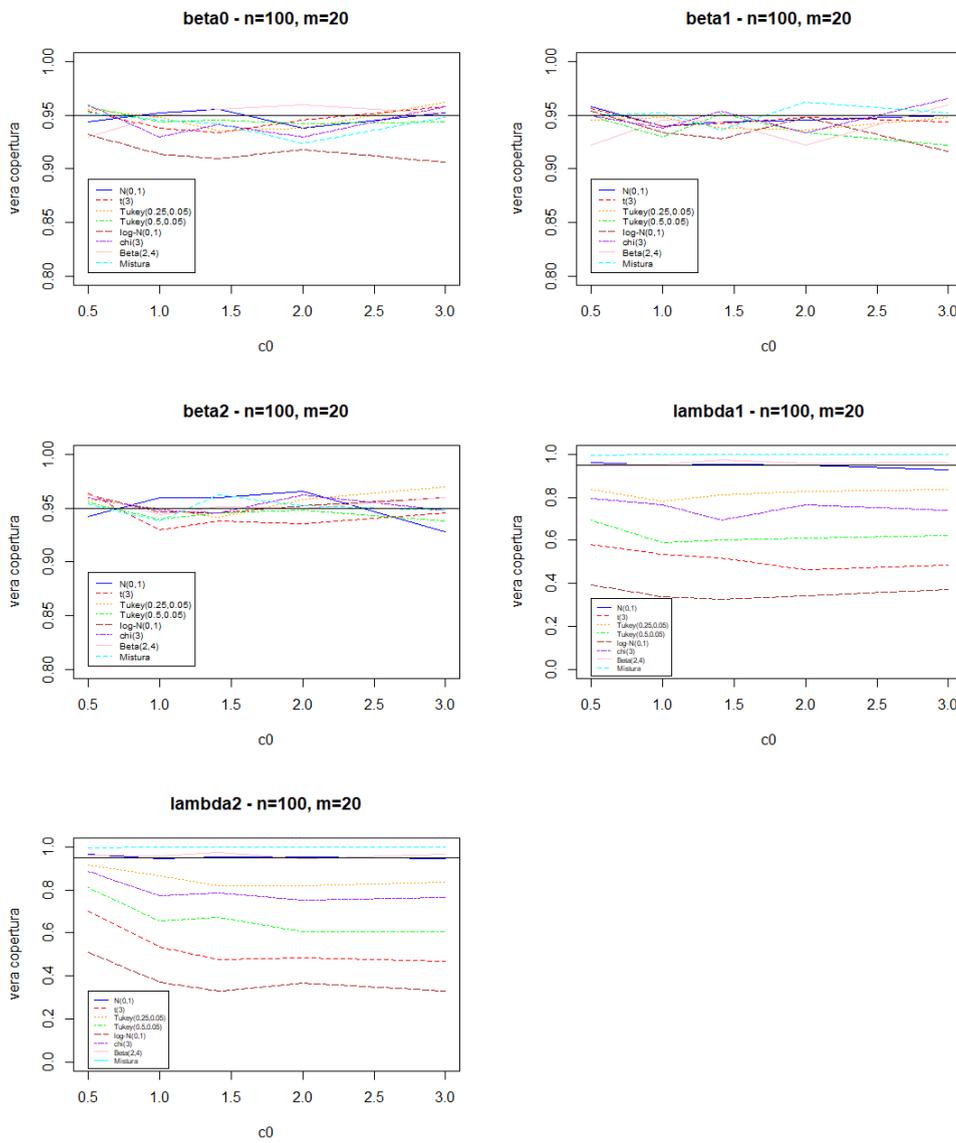


FIGURA D.6: Vere coperture degli intervalli di confidenza nel caso $n = 100$ e $m = 20$.

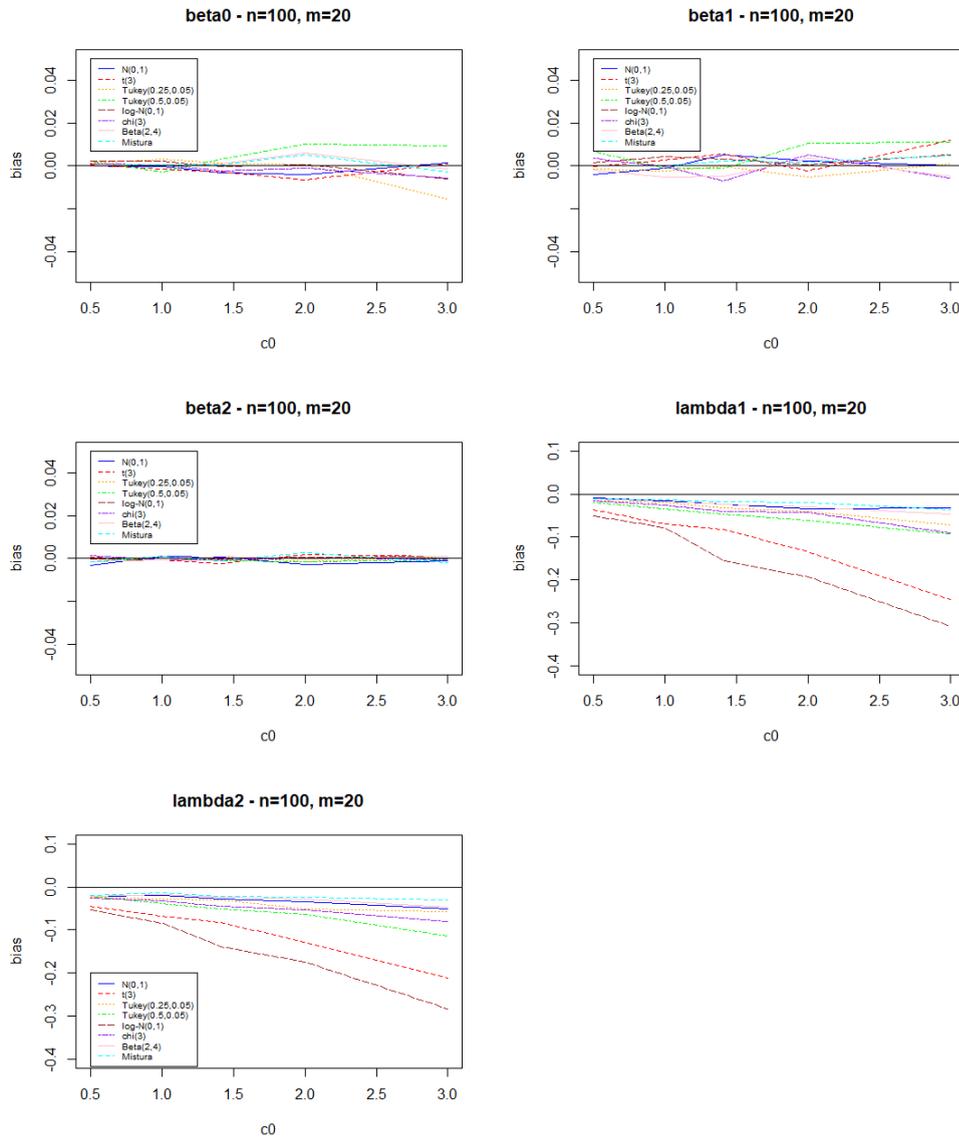


FIGURA D.7: Distorsioni empiriche nel caso $n = 50$ e $m = 10$, quando il numero di dataset per ogni combinazione è fissato a 3000.

D.2 Convergenza nelle stime

Distribuzione	stime - 0.5	I.C. - 0.5	stime - 1	I.C. - 1	stime - $\sqrt{2}$	I.C. - $\sqrt{2}$
(1)	0	31	0	10	0	6
(2)	0	24	0	9	0	11
(3)	0	28	0	8	0	10
(4)	0	39	0	13	0	7
(5)	2	40	0	17	0	11
(6)	0	30	0	15	0	7
(7)	0	22	0	11	0	5
(8)	0	26	0	8	0	5

Distribuzione	stime - 2	I.C. - 2	stime - 3	I.C. - 3
(1)	0	2	0	1
(2)	0	3	0	8
(3)	0	3	0	2
(4)	0	3	0	2
(5)	0	3	0	7
(6)	0	3	0	3
(7)	0	6	0	1
(8)	0	7	0	3

TABELLA D.1: Numero di *dataset*, su un totale di 500, per i quali non vi è convergenza nel calcolo delle stime o degli estremi degli intervalli di confidenza delle componenti di deviazione standard nel caso $n = 100$, $m = 10$.

Distribuzione	stime - 0.5	I.C. - 0.5	stime - 1	I.C. - 1	stime - $\sqrt{2}$	I.C. - $\sqrt{2}$
(1)	0	34	0	21	0	11
(2)	1	38	0	28	0	20
(3)	0	25	0	27	0	25
(4)	2	26	0	18	0	21
(5)	3	47	0	27	0	18
(6)	0	27	0	19	0	15
(7)	0	32	0	27	0	14
(8)	0	26	0	27	0	20

Distribuzione	stime - 2	I.C. - 2	stime - 3	I.C. - 3
(1)	0	20	0	18
(2)	0	16	0	13
(3)	0	10	0	11
(4)	0	16	0	13
(5)	0	15	0	18
(6)	0	14	0	11
(7)	0	16	0	15
(8)	0	18	0	24

TABELLA D.2: Numero di *dataset*, su un totale di 500, per i quali non vi è convergenza nel calcolo delle stime o degli estremi degli intervalli di confidenza delle componenti di deviazione standard nel caso $n = 50$, $m = 20$.

Distribuzione	stime - 0.5	I.C. - 0.5	stime - 1	I.C. - 1	stime - $\sqrt{2}$	I.C. - $\sqrt{2}$
(1)	1	20	1	10	3	12
(2)	0	13	2	11	0	8
(3)	1	15	1	12	3	12
(4)	1	17	2	22	6	11
(5)	1	25	0	3	3	12
(6)	0	20	3	15	3	14
(7)	1	18	3	14	5	21
(8)	1	14	2	10	0	8

Distribuzione	stime - 2	I.C. - 2	stime - 3	I.C. - 3
(1)	3	6	2	7
(2)	3	12	2	10
(3)	1	11	1	12
(4)	1	11	0	3
(5)	1	7	0	9
(6)	0	3	2	12
(7)	1	10	3	13
(8)	0	7	2	13

TABELLA D.3: Numero di *dataset*, su un totale di 500, per i quali non vi è convergenza nel calcolo delle stime o degli estremi degli intervalli di confidenza delle componenti di deviazione standard nel caso $n = 100$, $m = 20$.

Appendice E

Codice R

E.1 Studio di simulazione

```
#Studio di simulazione per analizzare la distorsione e l'accordo tra livello nominale  
#e copertura effettiva degli intervalli di confidenza per i parametri di un modello  
#misto lineare sotto errata specificazione della distribuzione degli effetti casuali.  
#Stime e standard error dei parametri sono trovati attraverso lme() con method="ML".
```

```
rm(list=ls())  
library(tukeyGH)  
library(nlme)
```

```
#####
```

```
#Grafici delle distribuzioni utilizzate trasformate in modo da avere media nulla e  
#varianza unitaria
```

```
#1. N(0,1) -> non c'e' misspecificazione.  
curve(dnorm(x), -5, 5, ylab="", xlab="", main="N(0,1)")
```

```
#2. t(3) t di Student con 3 gradi di liberta'  
curve(sqrt(3)*dt(sqrt(3)*x, df=3), -5, 5, ylab="", xlab="", main="t(3) standardizzata")
```

```
I<-rgh(1e6, g=0.25, h=0.05)  
mu_gh1<-mean(I)  
var_gh1<-var(I) #questi sono i valori saranno presi come media e varianza  
#di una Tukey (g=0.25, h=0.05)
```

```
#Allo stesso modo per la Tukey (g=0.5, h=0.05)  
I<-rgh(1e6, g=0.5, h=0.05)  
mu_gh2<-mean(I)  
var_gh2<-var(I)
```

```
#3. Tukey con g=0.25, h=0.05 (g governa l'asimmetria, h la curtosi)  
curve(sqrt(var_gh1)*dgh(sqrt(var_gh1)*x+mu_gh1, g=0.25, h=0.05), -5, 5, xlab="", ylab="",  
main="Tukey g=0.25, h=0.05 standardizzata")
```

```

#4. Tukey con g=0.5, h=0.05
curve(sqrt(var_gh2)*dgh(sqrt(var_gh2)*x+mu_gh2,g=0.25,h=0.05),-5,5,xlab="",ylab="",
main="Tukey g=0.25, h=0.05 standardizzata")

#5. log-normale a media 0 e log-varianza 1 (coincide con una Tukey(g=1,h=0))
curve(sqrt(exp(1)*(exp(1)-1))*dlnorm(sqrt(exp(1)*(exp(1)-1))*x+exp(0.5),0,1),-5,5,
xlab="",ylab="",main="log-normale(0,1) standardizzata")

#6.  $X^2_3$  chi quadro con 3 gradi di liberta'
curve(sqrt(6)*dchisq((sqrt(6)*x+3),df=3),-5,5,xlab="",ylab="",
main="Chi(3) standardizzata")

#7. beta(2,4) una beta con alfa=2 e beta=4
curve(sqrt(2/63)*dbeta(sqrt(2/63)*x+1/3,2,4),-5,5,
xlab="",ylab="",main="Beta(2,4) standardizzata")

#8. Mistura di due normali  $N(-3,1)$  e  $N(3,1)$  ciascuna con probabilita'  $p=0.5$ 
curve((dnorm(x*sqrt(10), -3)+dnorm(x*sqrt(10), 3))*sqrt(10)/2,-5,5,
xlab="",ylab="", main="Mistura standardizzata")

#####

#di seguito gli algoritmi per generare da ciascuna delle distribuzioni
#standardizzate

#per la normale la funzione c'e' gia', e' rnorm con valori di default.

#2. t di student con 3 g.d.l.:
r_t3<-function(n) {
  1/sqrt(3)*rt(n,df=3)
}

#3. Tukey(g=0.5,h=0.05)
r_gh1<-function(n) {
  1/sqrt(var_gh1)*rgh(n,g=0.25,h=0.05)-mu_gh1/sqrt(var_gh1)
}

#4. Tukey(g=0.25,h=0.05)
r_gh2<-function(n) {
  1/sqrt(var_gh2)*rgh(n,g=0.5,h=0.05)-mu_gh2/sqrt(var_gh2)
}

#5. log-normale (0,1)
r_lN<-function(n) {
  1/sqrt(exp(1)*(exp(1)-1))*rlnorm(n)-exp(0.5)/sqrt(exp(1)*(exp(1)-1))
}

#6. chi quadro 3 con 3 g.d.l.
r_chi3<-function(n) {
  1/sqrt(6)*rchisq(n,df=3)-3/sqrt(6)
}

#7. Beta(2,4)
r_beta<-function(n) {
  sqrt(63/2)*rbeta(n,2,4)-1/3*sqrt(63/2)
}

```

```

}

#8. Mistura di normali
r_mist<-function(n) {
  xx<-rbinom(n,1,0.5)
  v<-rep(NA,n)
  for(i in 1:n) {
    if (xx[i]==1) {
      v[i]<-rnorm(1,-3,1)
    }
    else {
      v[i]<-rnorm(1,3,1)
    }
  }
  1/sqrt(10)*v
}

#####

#Sono fissati qui i valori che saranno costanti per tutti i dataset simulati.
M=500
n=50
m=10
beta0=2
beta1=1
beta2=1
id<-rep(c(1:n),each=m)
x1<-rep(seq(-1,1,length=m),n)

test.copertura<-function(theta,theta.hat,se) { #test per vedere se intervallo di
  #confidenza contiene il vero valore del parametro theta
  abs((theta.hat-theta)/se)<qnorm(0.975)
  #funzione ritorna 1 se lo contiene, 0 altrimenti
}

simula.stime<-function (M,rg,c0) {
  #M<-numero di dataset da simulare
  #rg<-algoritmo per generare dalla distribuzione standardizzata degli effetti
  #casuali
  #c0<-deviazione standard degli effetti casuali
  stime<-matrix(NA,5,M) #si prealloca la matrice che conterra' le stime
  coperture<-matrix(NA,5,M) #si prealloca la matrice che conterra' le coperture
  #(si/no) degli intervalli di confidenza
  non.converge.stima<-0 #variabile che conterra' il numero di volte che lme()
  #non riesce a trovare le stime del modello
  non.converge.ic<-0 #variabile che conterra' il numero di volte che intervals()
  #non riesce a calcolare gli estremi degli intervalli di confidenza per le s.d.
  #components
  for(q in 1:M) {
    non.convergenza<-FALSE #booleana, serve come controllo per saltare
    #iterazioni per cui lme() non riesce a trovare le stime
    a<-c0*rg(n)
    b<-c0*rg(n) #genero gli effetti casuali
    a<-rep(a,each=m)
    b<-rep(b,each=m) #si uniforma a e b al formato lungo
  }
}

```

```

x2<-runif(n*m,-1,1) #si genera l'altra var. dipendente
epsilon<-rnorm(n*m) #si generano gli errori casuali
y<-beta0+a+(beta1+b)*x1+beta2*x2+epsilon #si simulano le risposte
tryCatch({
  dati.lme<-lme(y~x1+x2,random=~1+x1|id,method="ML") #si stima il modello
},
error=function(e) {
  print("Non convergenza nelle stime dei parametri")
  non.converge.stima<<-non.converge.stima+1
  non.convergenza<<-TRUE
  #Quando lme() da' errore, lo si segnala a schermo e si aggiorna il
  #valore di non.converge.stima
})
if (non.convergenza) {
  next #si passa all'iterazione successiva quando non si ha convergenza
}
stime[1,q]=summary(dati.lme)$tTable[1,1] #beta0.hat
stime[2,q]=summary(dati.lme)$tTable[2,1] #beta1.hat
stime[3,q]=summary(dati.lme)$tTable[3,1] #beta2.hat
stime[4,q]=sqrt(as.numeric(VarCorr(dati.lme)[c(1,2),1]))[1] #lambda1.hat
stime[5,q]=sqrt(as.numeric(VarCorr(dati.lme)[c(1,2),1]))[2] #lambda2.hat
coperture[1,q]=test.copertura(beta0,summary(dati.lme)$tTable[1,1],
summary(dati.lme)$tTable[1,2])#copertura beta0
coperture[2,q]=test.copertura(beta1,summary(dati.lme)$tTable[2,1],
summary(dati.lme)$tTable[2,2])#copertura beta1
coperture[3,q]=test.copertura(beta2,summary(dati.lme)$tTable[3,1],
summary(dati.lme)$tTable[3,2])#copertura beta2
tryCatch({
  coperture[4,q]=(intervals(dati.lme)$reStruct$id[1,1]<=c0 &
intervals(dati.lme)$reStruct$id[1,3]>=c0)#copertura
#per lambda1
  coperture[5,q]=(intervals(dati.lme)$reStruct$id[2,1]<=c0 &
intervals(dati.lme)$reStruct$id[2,3]>=c0)#copertura
#per lambda2
},
error=function(e) {
  print("Non convergenza negli i.c. delle s.d. components")
  non.converge.ic<<-non.converge.ic+1
  #Quando intervals() da' errore, lo si segnala a schermo e
  #si aggiorna il valore di non.converge.ic
})
if (q %% 10 == 0)
  print(q) #per tenere il passo del punto nel ciclo
}
non.converge.ic<-non.converge.ic+non.converge.stima
#le volte in cui lme() non trova le stime, non si hanno ovviamente
#nemmeno gli i.c. (ma non viceversa)

cat("dataset per cui non si trovano stime parametri:",non.converge.stima)
cat("\ndataset per cui non si trovano estremi i.c.:",non.converge.ic)
cat("\n")
list("stime"=stime,"coperture"=coperture,
"non.converge.stima"=non.converge.stima,"non.converge.ic"=non.converge.ic)
#funzione ritorna una lista con le 2 matrici e il numero di dataset problematici
}

```

```

#Normale c0=0.5
c0=0.5
N_0.5<-simula.stime(M,rnorm,c0)
media.stime_N_0.5<-apply(N_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
#Si ignorano i valori NA, altrimenti nel momento in cui si ha una stima NA
# per una non convergenza, l'intera media sarebbe NA
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_N_0.5<-media.stime_N_0.5-truepar
coperture_N_0.5<-apply(N_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))
#per ogni riga si calcola la proporzione di volte in cui il vero valore
#del parametro e' contenuto nell'intervallo di confidenza

#Normale c0=1
c0=1
N_1<-simula.stime(M,rnorm,c0)
media.stime_N_1<-apply(N_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_N_1<-media.stime_N_1-truepar
coperture_N_1<-apply(N_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#Normale c0=sqrt(2)
c0=sqrt(2)
N_sqrt2<-simula.stime(M,rnorm,c0)
media.stime_N_sqrt2<-apply(N_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_N_sqrt2<-media.stime_N_sqrt2-truepar
coperture_N_sqrt2<-apply(N_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Normale c0=2
c0=2
N_2<-simula.stime(M,rnorm,c0)
media.stime_N_2<-apply(N_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_N_2<-media.stime_N_2-truepar
coperture_N_2<-apply(N_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Normale c0=3
c0=3
N_3<-simula.stime(M,rnorm,c0)
media.stime_N_3<-apply(N_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_N_3<-media.stime_N_3-truepar
coperture_N_3<-apply(N_3$coperture,1,function(x) mean(x,na.rm=TRUE))

#t(3) c0=0.5
c0=0.5
t3_0.5<-simula.stime(M,r_t3,c0)
media.stime_t3_0.5<-apply(t3_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_t3_0.5<-media.stime_t3_0.5-truepar
coperture_t3_0.5<-apply(t3_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#t(3) c0=1
c0=1

```

```

t3_1<-simula.stime(M,r_t3,c0)
media.stime_t3_1<-apply(t3_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_t3_1<-media.stime_t3_1-truepar
coperture_t3_1<-apply(t3_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#t(3) c0=sqrt(2)
c0=sqrt(2)
t3_sqrt2<-simula.stime(M,r_t3,c0)
media.stime_t3_sqrt2<-apply(t3_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_t3_sqrt2<-media.stime_t3_sqrt2-truepar
coperture_t3_sqrt2<-apply(t3_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#t(3) c0=2
c0=2
t3_2<-simula.stime(M,r_t3,c0)
media.stime_t3_2<-apply(t3_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_t3_2<-media.stime_t3_2-truepar
coperture_t3_2<-apply(t3_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#t(3) c0=3
c0=3
t3_3<-simula.stime(M,r_t3,c0)
media.stime_t3_3<-apply(t3_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_t3_3<-media.stime_t3_3-truepar
coperture_t3_3<-apply(t3_3$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.25, h=0.05) c0=0.5
c0=0.5
gh1_0.5<-simula.stime(M,r_gh1,c0)
media.stime_gh1_0.5<-apply(gh1_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh1_0.5<-media.stime_gh1_0.5-truepar
coperture_gh1_0.5<-apply(gh1_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.25, h=0.05) c0=1
c0=1
gh1_1<-simula.stime(M,r_gh1,c0)
media.stime_gh1_1<-apply(gh1_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh1_1<-media.stime_gh1_1-truepar
coperture_gh1_1<-apply(gh1_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.25, h=0.05) c0=sqrt(2)
c0=sqrt(2)
gh1_sqrt2<-simula.stime(M,r_gh1,c0)
media.stime_gh1_sqrt2<-apply(gh1_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh1_sqrt2<-media.stime_gh1_sqrt2-truepar
coperture_gh1_sqrt2<-apply(gh1_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.25, h=0.05) c0=2

```

```
c0=2
gh1_2<-simula.stime(M,r_gh1,c0)
media.stime_gh1_2<-apply(gh1_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh1_2<-media.stime_gh1_2-truepar
coperture_gh1_2<-apply(gh1_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.25, h=0.05) c0=3
c0=3
gh1_3<-simula.stime(M,r_gh1,c0)
media.stime_gh1_3<-apply(gh1_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh1_3<-media.stime_gh1_3-truepar
coperture_gh1_3<-apply(gh1_3$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.5, h=0.05) c0=0.5
c0=0.5
gh2_0.5<-simula.stime(M,r_gh2,c0)
media.stime_gh2_0.5<-apply(gh2_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh2_0.5<-media.stime_gh2_0.5-truepar
coperture_gh2_0.5<-apply(gh2_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.5, h=0.05) c0=1
c0=1
gh2_1<-simula.stime(M,r_gh2,c0)
media.stime_gh2_1<-apply(gh2_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh2_1<-media.stime_gh2_1-truepar
coperture_gh2_1<-apply(gh2_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.5, h=0.05) c0=sqrt(2)
c0=sqrt(2)
gh2_sqrt2<-simula.stime(M,r_gh2,c0)
media.stime_gh2_sqrt2<-apply(gh2_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh2_sqrt2<-media.stime_gh2_sqrt2-truepar
coperture_gh2_sqrt2<-apply(gh2_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.5, h=0.05) c0=2
c0=2
gh2_2<-simula.stime(M,r_gh2,c0)
media.stime_gh2_2<-apply(gh2_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh2_2<-media.stime_gh2_2-truepar
coperture_gh2_2<-apply(gh2_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Tukey (g=0.5, h=0.05) c0=3
c0=3
gh2_3<-simula.stime(M,r_gh2,c0)
media.stime_gh2_3<-apply(gh2_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_gh2_3<-media.stime_gh2_3-truepar
coperture_gh2_3<-apply(gh2_3$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```

#log-Normale(0,1) c0=0.5
c0=0.5
lN_0.5<-simula.stime(M,r_lN,c0)
media.stime_lN_0.5<-apply(lN_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_lN_0.5<-media.stime_lN_0.5-truepar
coperture_lN_0.5<-apply(lN_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#log-Normale(0,1) c0=1
c0=1
lN_1<-simula.stime(M,r_lN,c0)
media.stime_lN_1<-apply(lN_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_lN_1<-media.stime_lN_1-truepar
coperture_lN_1<-apply(lN_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#log-Normale(0,1) c0=sqrt(2)
c0=sqrt(2)
lN_sqrt2<-simula.stime(M,r_lN,c0)
media.stime_lN_sqrt2<-apply(lN_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_lN_sqrt2<-media.stime_lN_sqrt2-truepar
coperture_lN_sqrt2<-apply(lN_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#log-Normale(0,1) c0=2
c0=2
lN_2<-simula.stime(M,r_lN,c0)
media.stime_lN_2<-apply(lN_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_lN_2<-media.stime_lN_2-truepar
coperture_lN_2<-apply(lN_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#log-Normale(0,1) c0=3
c0=3
lN_3<-simula.stime(M,r_lN,c0)
media.stime_lN_3<-apply(lN_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_lN_3<-media.stime_lN_3-truepar
coperture_lN_3<-apply(lN_3$coperture,1,function(x) mean(x,na.rm=TRUE))

#Chi(3) c0=0.5
c0=0.5
chi3_0.5<-simula.stime(M,r_chi3,c0)
media.stime_chi3_0.5<-apply(chi3_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_chi3_0.5<-media.stime_chi3_0.5-truepar
coperture_chi3_0.5<-apply(chi3_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#Chi(3) c0=1
c0=1
chi3_1<-simula.stime(M,r_chi3,c0)
media.stime_chi3_1<-apply(chi3_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_chi3_1<-media.stime_chi3_1-truepar
coperture_chi3_1<-apply(chi3_1$coperture,1,function(x) mean(x,na.rm=TRUE))

```

```
#Chi (3) c0=sqrt(2)
c0=sqrt(2)
chi3_sqrt2<-simula.stime(M,r_chi3,c0)
media.stime_chi3_sqrt2<-apply(chi3_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_chi3_sqrt2<-media.stime_chi3_sqrt2-truepar
coperture_chi3_sqrt2<-apply(chi3_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Chi (3) c0=2
c0=2
chi3_2<-simula.stime(M,r_chi3,c0)
media.stime_chi3_2<-apply(chi3_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_chi3_2<-media.stime_chi3_2-truepar
coperture_chi3_2<-apply(chi3_2$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Chi (3) c0=3
c0=3
chi3_3<-simula.stime(M,r_chi3,c0)
media.stime_chi3_3<-apply(chi3_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_chi3_3<-media.stime_chi3_3-truepar
coperture_chi3_3<-apply(chi3_3$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Beta (2,4) c0=0.5
c0=0.5
beta_0.5<-simula.stime(M,r_beta,c0)
media.stime_beta_0.5<-apply(beta_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_beta_0.5<-media.stime_beta_0.5-truepar
coperture_beta_0.5<-apply(beta_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Beta (2,4) c0=1
c0=1
beta_1<-simula.stime(M,r_beta,c0)
media.stime_beta_1<-apply(beta_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_beta_1<-media.stime_beta_1-truepar
coperture_beta_1<-apply(beta_1$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Beta (2,4) c0=sqrt(2)
c0=sqrt(2)
beta_sqrt2<-simula.stime(M,r_beta,c0)
media.stime_beta_sqrt2<-apply(beta_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_beta_sqrt2<-media.stime_beta_sqrt2-truepar
coperture_beta_sqrt2<-apply(beta_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))
```

```
#Beta (2,4) c0=2
c0=2
beta_2<-simula.stime(M,r_beta,c0)
media.stime_beta_2<-apply(beta_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_beta_2<-media.stime_beta_2-truepar
```

```

coperture_beta_2<-apply(beta_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Beta(2,4) c0=3
c0=3
beta_3<-simula.stime(M,r_beta,c0)
media.stime_beta_3<-apply(beta_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_beta_3<-media.stime_beta_3-truepar
coperture_beta_3<-apply(beta_3$coperture,1,function(x) mean(x,na.rm=TRUE))

#Mistura di normali c0=0.5
c0=0.5
mist_0.5<-simula.stime(M,r_mist,c0)
media.stime_mist_0.5<-apply(mist_0.5$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_mist_0.5<-media.stime_mist_0.5-truepar
coperture_mist_0.5<-apply(mist_0.5$coperture,1,function(x) mean(x,na.rm=TRUE))

#Mistura di normali c0=1
c0=1
mist_1<-simula.stime(M,r_mist,c0)
media.stime_mist_1<-apply(mist_1$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_mist_1<-media.stime_mist_1-truepar
coperture_mist_1<-apply(mist_1$coperture,1,function(x) mean(x,na.rm=TRUE))

#Mistura di normali c0=sqrt(2)
c0=sqrt(2)
mist_sqrt2<-simula.stime(M,r_mist,c0)
media.stime_mist_sqrt2<-apply(mist_sqrt2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_mist_sqrt2<-media.stime_mist_sqrt2-truepar
coperture_mist_sqrt2<-apply(mist_sqrt2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Mistura di normali c0=2
c0=2
mist_2<-simula.stime(M,r_mist,c0)
media.stime_mist_2<-apply(mist_2$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_mist_2<-media.stime_mist_2-truepar
coperture_mist_2<-apply(mist_2$coperture,1,function(x) mean(x,na.rm=TRUE))

#Mistura di normali c0=3
c0=3
mist_3<-simula.stime(M,r_mist,c0)
media.stime_mist_3<-apply(mist_3$stime,1,function(x) mean(x,na.rm=TRUE))
truepar<-c(beta0,beta1,beta2,c0,c0)
bias_mist_3<-media.stime_mist_3-truepar
coperture_mist_3<-apply(mist_3$coperture,1,function(x) mean(x,na.rm=TRUE))

####

#Analisi grafiche dei risultati.

c0.values<-c(0.5,1,sqrt(2),2,3)

```

#Distorsioni:

```

beta0.biases_N<-c(bias_N_0.5[1], bias_N_1[1], bias_N_sqrt2[1], bias_N_2[1],
bias_N_3[1])
beta0.biases_t3<-c(bias_t3_0.5[1], bias_t3_1[1], bias_t3_sqrt2[1], bias_t3_2[1],
bias_t3_3[1])
beta0.biases_gh1<-c(bias_gh1_0.5[1], bias_gh1_1[1], bias_gh1_sqrt2[1], bias_gh1_2[1],
bias_gh1_3[1])
beta0.biases_gh2<-c(bias_gh2_0.5[1], bias_gh2_1[1], bias_gh2_sqrt2[1], bias_gh2_2[1],
bias_gh2_3[1])
beta0.biases_lN<-c(bias_lN_0.5[1], bias_lN_1[1], bias_lN_sqrt2[1], bias_lN_2[1],
bias_lN_3[1])
beta0.biases_chi3<-c(bias_chi3_0.5[1], bias_chi3_1[1], bias_chi3_sqrt2[1],
bias_chi3_2[1], bias_chi3_3[1])
beta0.biases_beta<-c(bias_beta_0.5[1], bias_beta_1[1], bias_beta_sqrt2[1],
bias_beta_2[1], bias_beta_3[1])
beta0.biases_mist<-c(bias_mist_0.5[1], bias_mist_1[1], bias_mist_sqrt2[1],
bias_mist_2[1], bias_mist_3[1])

beta1.biases_N<-c(bias_N_0.5[2], bias_N_1[2], bias_N_sqrt2[2], bias_N_2[2],
bias_N_3[2])
beta1.biases_t3<-c(bias_t3_0.5[2], bias_t3_1[2], bias_t3_sqrt2[2], bias_t3_2[2],
bias_t3_3[2])
beta1.biases_gh1<-c(bias_gh1_0.5[2], bias_gh1_1[2], bias_gh1_sqrt2[2], bias_gh1_2[2],
bias_gh1_3[2])
beta1.biases_gh2<-c(bias_gh2_0.5[2], bias_gh2_1[2], bias_gh2_sqrt2[2], bias_gh2_2[2],
bias_gh2_3[2])
beta1.biases_lN<-c(bias_lN_0.5[2], bias_lN_1[2], bias_lN_sqrt2[2], bias_lN_2[2],
bias_lN_3[2])
beta1.biases_chi3<-c(bias_chi3_0.5[2], bias_chi3_1[2], bias_chi3_sqrt2[2],
bias_chi3_2[2], bias_chi3_3[2])
beta1.biases_beta<-c(bias_beta_0.5[2], bias_beta_1[2], bias_beta_sqrt2[2],
bias_beta_2[2], bias_beta_3[2])
beta1.biases_mist<-c(bias_mist_0.5[2], bias_mist_1[2], bias_mist_sqrt2[2],
bias_mist_2[2], bias_mist_3[2])

beta2.biases_N<-c(bias_N_0.5[3], bias_N_1[3], bias_N_sqrt2[3], bias_N_2[3],
bias_N_3[3])
beta2.biases_t3<-c(bias_t3_0.5[3], bias_t3_1[3], bias_t3_sqrt2[3], bias_t3_2[3],
bias_t3_3[3])
beta2.biases_gh1<-c(bias_gh1_0.5[3], bias_gh1_1[3], bias_gh1_sqrt2[3], bias_gh1_2[3],
bias_gh1_3[3])
beta2.biases_gh2<-c(bias_gh2_0.5[3], bias_gh2_1[3], bias_gh2_sqrt2[3], bias_gh2_2[3],
bias_gh2_3[3])
beta2.biases_lN<-c(bias_lN_0.5[3], bias_lN_1[3], bias_lN_sqrt2[3], bias_lN_2[3],
bias_lN_3[3])
beta2.biases_chi3<-c(bias_chi3_0.5[3], bias_chi3_1[3], bias_chi3_sqrt2[3],
bias_chi3_2[3], bias_chi3_3[3])
beta2.biases_beta<-c(bias_beta_0.5[3], bias_beta_1[3], bias_beta_sqrt2[3],
bias_beta_2[3], bias_beta_3[3])
beta2.biases_mist<-c(bias_mist_0.5[3], bias_mist_1[3], bias_mist_sqrt2[3],
bias_mist_2[3], bias_mist_3[3])

lambda1.biases_N<-c(bias_N_0.5[4], bias_N_1[4], bias_N_sqrt2[4], bias_N_2[4],

```

```

bias_N_3[4])
lambda1.biases_t3<-c(bias_t3_0.5[4],bias_t3_1[4],bias_t3_sqrt2[4],bias_t3_2[4],
bias_t3_3[4])
lambda1.biases_gh1<-c(bias_gh1_0.5[4],bias_gh1_1[4],bias_gh1_sqrt2[4],bias_gh1_2[4],
bias_gh1_3[4])
lambda1.biases_gh2<-c(bias_gh2_0.5[4],bias_gh2_1[4],bias_gh2_sqrt2[4],bias_gh2_2[4],
bias_gh2_3[4])
lambda1.biases_lN<-c(bias_lN_0.5[4],bias_lN_1[4],bias_lN_sqrt2[4],bias_lN_2[4],
bias_lN_3[4])
lambda1.biases_chi3<-c(bias_chi3_0.5[4],bias_chi3_1[4],bias_chi3_sqrt2[4],
bias_chi3_2[4],bias_chi3_3[4])
lambda1.biases_beta<-c(bias_beta_0.5[4],bias_beta_1[4],bias_beta_sqrt2[4],
bias_beta_2[4],bias_beta_3[4])
lambda1.biases_mist<-c(bias_mist_0.5[4],bias_mist_1[4],bias_mist_sqrt2[4],
bias_mist_2[4],bias_mist_3[4])

lambda2.biases_N<-c(bias_N_0.5[5],bias_N_1[5],bias_N_sqrt2[5],bias_N_2[5],
bias_N_3[5])
lambda2.biases_t3<-c(bias_t3_0.5[5],bias_t3_1[5],bias_t3_sqrt2[5],bias_t3_2[5],
bias_t3_3[5])
lambda2.biases_gh1<-c(bias_gh1_0.5[5],bias_gh1_1[5],bias_gh1_sqrt2[5],bias_gh1_2[5],
bias_gh1_3[5])
lambda2.biases_gh2<-c(bias_gh2_0.5[5],bias_gh2_1[5],bias_gh2_sqrt2[5],bias_gh2_2[5],
bias_gh2_3[5])
lambda2.biases_lN<-c(bias_lN_0.5[5],bias_lN_1[5],bias_lN_sqrt2[5],bias_lN_2[5],
bias_lN_3[5])
lambda2.biases_chi3<-c(bias_chi3_0.5[5],bias_chi3_1[5],bias_chi3_sqrt2[5],
bias_chi3_2[5],bias_chi3_3[5])
lambda2.biases_beta<-c(bias_beta_0.5[5],bias_beta_1[5],bias_beta_sqrt2[5],
bias_beta_2[5],bias_beta_3[5])
lambda2.biases_mist<-c(bias_mist_0.5[5],bias_mist_1[5],bias_mist_sqrt2[5],
bias_mist_2[5],bias_mist_3[5])

plot(c0.values,beta0.biases_N,type="l", col="blue",xlab="c0",ylab="bias",
main="beta0 - n=100, m=20",ylim=c(-0.1,0.1))
lines(c0.values,beta0.biases_t3,col="red",lty=2)
lines(c0.values,beta0.biases_gh1,col="orange",lty=3)
lines(c0.values,beta0.biases_gh2,col="green",lty=4)
lines(c0.values,beta0.biases_lN,col="brown",lty=5)
lines(c0.values,beta0.biases_chi3,col="purple",lty=6)
lines(c0.values,beta0.biases_beta,col="pink",lty=1)
lines(c0.values,beta0.biases_mist,col="cyan",lty=2)
abline(h=0)
legend(0.5,0.1,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)",
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

plot(c0.values,beta1.biases_N,type="l", col="blue",xlab="c0",ylab="bias",
main="beta1 - n=100, m=20",ylim=c(-0.1,0.1))
lines(c0.values,beta1.biases_t3,col="red",lty=2)
lines(c0.values,beta1.biases_gh1,col="orange",lty=3)
lines(c0.values,beta1.biases_gh2,col="green",lty=4)
lines(c0.values,beta1.biases_lN,col="brown",lty=5)
lines(c0.values,beta1.biases_chi3,col="purple",lty=6)

```

```

lines(c0.values,beta1.biases_beta,col="pink",lty=1)
lines(c0.values,beta1.biases_mist,col="cyan",lty=2)
abline(h=0)
legend(0.5,0.1,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)","
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

plot(c0.values,beta2.biases_N,type="l",col="blue",xlab="c0",ylab="bias",
main="beta2 - n=100, m=20",ylim=c(-0.1,0.1))
lines(c0.values,beta2.biases_t3,col="red",lty=2)
lines(c0.values,beta2.biases_gh1,col="orange",lty=3)
lines(c0.values,beta2.biases_gh2,col="green",lty=4)
lines(c0.values,beta2.biases_lN,col="brown",lty=5)
lines(c0.values,beta2.biases_chi3,col="purple",lty=6)
lines(c0.values,beta2.biases_beta,col="pink",lty=1)
lines(c0.values,beta2.biases_mist,col="cyan",lty=2)
abline(h=0)
legend(0.5,0.1,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)","
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

plot(c0.values,lambda1.biases_N,type="l",col="blue",xlab="c0",ylab="bias",
main="lambda1 - n=100, m=20",ylim=c(-0.4,0.1))
lines(c0.values,lambda1.biases_t3,col="red",lty=2)
lines(c0.values,lambda1.biases_gh1,col="orange",lty=3)
lines(c0.values,lambda1.biases_gh2,col="green",lty=4)
lines(c0.values,lambda1.biases_lN,col="brown",lty=5)
lines(c0.values,lambda1.biases_chi3,col="purple",lty=6)
lines(c0.values,lambda1.biases_beta,col="pink",lty=1)
lines(c0.values,lambda1.biases_mist,col="cyan",lty=2)
abline(h=0)
legend(0.5,-0.2,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)","
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

plot(c0.values,lambda2.biases_N,type="l",col="blue",xlab="c0",ylab="bias",
main="lambda2 - n=100, m=20",ylim=c(-0.4,0.1))
lines(c0.values,lambda2.biases_t3,col="red",lty=2)
lines(c0.values,lambda2.biases_gh1,col="orange",lty=3)
lines(c0.values,lambda2.biases_gh2,col="green",lty=4)
lines(c0.values,lambda2.biases_lN,col="brown",lty=5)
lines(c0.values,lambda2.biases_chi3,col="purple",lty=6)
lines(c0.values,lambda2.biases_beta,col="pink",lty=1)
lines(c0.values,lambda2.biases_mist,col="cyan",lty=2)
abline(h=0)
legend(0.5,-0.2,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)","
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

#Coperture:

```

```

beta0.coperture_N<-c(coperture_N_0.5[1], coperture_N_1[1], coperture_N_sqrt2[1],
coperture_N_2[1], coperture_N_3[1])
beta0.coperture_t3<-c(coperture_t3_0.5[1], coperture_t3_1[1], coperture_t3_sqrt2[1],
coperture_t3_2[1], coperture_t3_3[1])
beta0.coperture_gh1<-c(coperture_gh1_0.5[1], coperture_gh1_1[1],
coperture_gh1_sqrt2[1], coperture_gh1_2[1], coperture_gh1_3[1])
beta0.coperture_gh2<-c(coperture_gh2_0.5[1], coperture_gh2_1[1], coperture_gh2_sqrt2[1],
coperture_gh2_2[1], coperture_gh2_3[1])
beta0.coperture_lN<-c(coperture_lN_0.5[1], coperture_lN_1[1], coperture_lN_sqrt2[1],
coperture_lN_2[1], coperture_lN_3[1])
beta0.coperture_chi3<-c(coperture_chi3_0.5[1], coperture_chi3_1[1],
coperture_chi3_sqrt2[1], coperture_chi3_2[1],
coperture_chi3_3[1])
beta0.coperture_beta<-c(coperture_beta_0.5[1], coperture_beta_1[1],
coperture_beta_sqrt2[1], coperture_beta_2[1],
coperture_beta_3[1])
beta0.coperture_mist<-c(coperture_mist_0.5[1], coperture_mist_1[1],
coperture_mist_sqrt2[1], coperture_mist_2[1],
coperture_mist_3[1])

```

```

beta1.coperture_N<-c(coperture_N_0.5[2], coperture_N_1[2], coperture_N_sqrt2[2],
coperture_N_2[2], coperture_N_3[2])
beta1.coperture_t3<-c(coperture_t3_0.5[2], coperture_t3_1[2], coperture_t3_sqrt2[2],
coperture_t3_2[2], coperture_t3_3[2])
beta1.coperture_gh1<-c(coperture_gh1_0.5[2], coperture_gh1_1[2], coperture_gh1_sqrt2[2],
coperture_gh1_2[2], coperture_gh1_3[2])
beta1.coperture_gh2<-c(coperture_gh2_0.5[2], coperture_gh2_1[2], coperture_gh2_sqrt2[2],
coperture_gh2_2[2], coperture_gh2_3[2])
beta1.coperture_lN<-c(coperture_lN_0.5[2], coperture_lN_1[2], coperture_lN_sqrt2[2],
coperture_lN_2[2], coperture_lN_3[2])
beta1.coperture_chi3<-c(coperture_chi3_0.5[2], coperture_chi3_1[2],
coperture_chi3_sqrt2[2], coperture_chi3_2[2],
coperture_chi3_3[2])
beta1.coperture_beta<-c(coperture_beta_0.5[2], coperture_beta_1[2],
coperture_beta_sqrt2[2], coperture_beta_2[2],
coperture_beta_3[2])
beta1.coperture_mist<-c(coperture_mist_0.5[2], coperture_mist_1[2],
coperture_mist_sqrt2[2], coperture_mist_2[2],
coperture_mist_3[2])

```

```

beta2.coperture_N<-c(coperture_N_0.5[3], coperture_N_1[3], coperture_N_sqrt2[3],
coperture_N_2[3], coperture_N_3[3])
beta2.coperture_t3<-c(coperture_t3_0.5[3], coperture_t3_1[3], coperture_t3_sqrt2[3],
coperture_t3_2[3], coperture_t3_3[3])
beta2.coperture_gh1<-c(coperture_gh1_0.5[3], coperture_gh1_1[3], coperture_gh1_sqrt2[3],
coperture_gh1_2[3], coperture_gh1_3[3])
beta2.coperture_gh2<-c(coperture_gh2_0.5[3], coperture_gh2_1[3], coperture_gh2_sqrt2[3],
coperture_gh2_2[3], coperture_gh2_3[3])
beta2.coperture_lN<-c(coperture_lN_0.5[3], coperture_lN_1[3], coperture_lN_sqrt2[3],
coperture_lN_2[3], coperture_lN_3[3])
beta2.coperture_chi3<-c(coperture_chi3_0.5[3], coperture_chi3_1[3],
coperture_chi3_sqrt2[3], coperture_chi3_2[3],
coperture_chi3_3[3])
beta2.coperture_beta<-c(coperture_beta_0.5[3], coperture_beta_1[3],
coperture_beta_sqrt2[3], coperture_beta_2[3],

```

```

coperture_beta_3[3])
beta2.coperture_mist<-c(coperture_mist_0.5[3],coperture_mist_1[3],
coperture_mist_sqrt2[3],coperture_mist_2[3],
coperture_mist_3[3])

lambda1.coperture_N<-c(coperture_N_0.5[4],coperture_N_1[4],coperture_N_sqrt2[4],
coperture_N_2[4],coperture_N_3[4])
lambda1.coperture_t3<-c(coperture_t3_0.5[4],coperture_t3_1[4],coperture_t3_sqrt2[4],
coperture_t3_2[4],coperture_t3_3[4])
lambda1.coperture_gh1<-c(coperture_gh1_0.5[4],coperture_gh1_1[4],
coperture_gh1_sqrt2[4],coperture_gh1_2[4],
coperture_gh1_3[4])
lambda1.coperture_gh2<-c(coperture_gh2_0.5[4],coperture_gh2_1[4],
coperture_gh2_sqrt2[4],coperture_gh2_2[4],
coperture_gh2_3[4])
lambda1.coperture_1N<-c(coperture_1N_0.5[4],coperture_1N_1[4],coperture_1N_sqrt2[4],
coperture_1N_2[4],coperture_1N_3[4])
lambda1.coperture_chi3<-c(coperture_chi3_0.5[4],coperture_chi3_1[4],
coperture_chi3_sqrt2[4],coperture_chi3_2[4],
coperture_chi3_3[4])
lambda1.coperture_beta<-c(coperture_beta_0.5[4],coperture_beta_1[4],
coperture_beta_sqrt2[4],coperture_beta_2[4],
coperture_beta_3[4])
lambda1.coperture_mist<-c(coperture_mist_0.5[4],coperture_mist_1[4],
coperture_mist_sqrt2[4],coperture_mist_2[4],
coperture_mist_3[4])

lambda2.coperture_N<-c(coperture_N_0.5[5],coperture_N_1[5],coperture_N_sqrt2[5],
coperture_N_2[5],coperture_N_3[5])
lambda2.coperture_t3<-c(coperture_t3_0.5[5],coperture_t3_1[5],coperture_t3_sqrt2[5],
coperture_t3_2[5],coperture_t3_3[5])
lambda2.coperture_gh1<-c(coperture_gh1_0.5[5],coperture_gh1_1[5],
coperture_gh1_sqrt2[5],coperture_gh1_2[5],
coperture_gh1_3[5])
lambda2.coperture_gh2<-c(coperture_gh2_0.5[5],coperture_gh2_1[5],
coperture_gh2_sqrt2[5],coperture_gh2_2[5],
coperture_gh2_3[5])
lambda2.coperture_1N<-c(coperture_1N_0.5[5],coperture_1N_1[5],coperture_1N_sqrt2[5],
coperture_1N_2[5],coperture_1N_3[5])
lambda2.coperture_chi3<-c(coperture_chi3_0.5[5],coperture_chi3_1[5],
coperture_chi3_sqrt2[5],coperture_chi3_2[5],
coperture_chi3_3[5])
lambda2.coperture_beta<-c(coperture_beta_0.5[5],coperture_beta_1[5],
coperture_beta_sqrt2[5],coperture_beta_2[5],
coperture_beta_3[5])
lambda2.coperture_mist<-c(coperture_mist_0.5[5],coperture_mist_1[5],
coperture_mist_sqrt2[5],coperture_mist_2[5],
coperture_mist_3[5])

plot(c0.values,beta0.coperture_N,type="l", col="blue",xlab="c0",
ylab="vera copertura",main="beta0 - n=100, m=20",ylim=c(0.8,1))

lines(c0.values,beta0.coperture_t3,col="red",lty=2)
lines(c0.values,beta0.coperture_gh1,col="orange",lty=3)
lines(c0.values,beta0.coperture_gh2,col="green",lty=4)

```

```

lines(c0.values,beta0.coperture_lN,col="brown",lty=5)
lines(c0.values,beta0.coperture_chi3,col="purple",lty=6)
lines(c0.values,beta0.coperture_beta,col="pink",lty=1)
lines(c0.values,beta0.coperture_mist,col="cyan",lty=2)
abline(h=0.95)
legend(0.5,0.89,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)",
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

```

```

plot(c0.values,beta1.coperture_N,type="l", col="blue",xlab="c0",
ylab="vera copertura",main="beta1 - n=100, m=20",ylim=c(0.8,1))
lines(c0.values,beta1.coperture_t3,col="red",lty=2)
lines(c0.values,beta1.coperture_gh1,col="orange",lty=3)
lines(c0.values,beta1.coperture_gh2,col="green",lty=4)
lines(c0.values,beta1.coperture_lN,col="brown",lty=5)
lines(c0.values,beta1.coperture_chi3,col="purple",lty=6)
lines(c0.values,beta1.coperture_beta,col="pink",lty=1)
lines(c0.values,beta1.coperture_mist,col="cyan",lty=2)
abline(h=0.95)
legend(0.5,0.89,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)",
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

```

```

plot(c0.values,beta2.coperture_N,type="l", col="blue",xlab="c0",
ylab="vera copertura",main="beta2 - n=100, m=20",ylim=c(0.8,1))
lines(c0.values,beta2.coperture_t3,col="red",lty=2)
lines(c0.values,beta2.coperture_gh1,col="orange",lty=3)
lines(c0.values,beta2.coperture_gh2,col="green",lty=4)
lines(c0.values,beta2.coperture_lN,col="brown",lty=5)
lines(c0.values,beta2.coperture_chi3,col="purple",lty=6)
lines(c0.values,beta2.coperture_beta,col="pink",lty=1)
lines(c0.values,beta2.coperture_mist,col="cyan",lty=2)
abline(h=0.95)
legend(0.5,0.89,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)",
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.6)

```

```

plot(c0.values,lambda1.coperture_N,type="l", col="blue",xlab="c0",
ylab="vera copertura",main="lambda1 - n=100, m=20",ylim=c(0,1))
lines(c0.values,lambda1.coperture_t3,col="red",lty=2)
lines(c0.values,lambda1.coperture_gh1,col="orange",lty=3)
lines(c0.values,lambda1.coperture_gh2,col="green",lty=4)
lines(c0.values,lambda1.coperture_lN,col="brown",lty=5)
lines(c0.values,lambda1.coperture_chi3,col="purple",lty=6)
lines(c0.values,lambda1.coperture_beta,col="pink",lty=1)
lines(c0.values,lambda1.coperture_mist,col="cyan",lty=2)
abline(h=0.95)
legend(0.5,0.33,legend=c("N(0,1)","t(3)","Tukey(0.25,0.05)","Tukey(0.5,0.05)",
"log-N(0,1)","chi(3)","Beta(2,4)","Mistura"),
col=c("blue","red","orange","green","brown","purple","pink","cyan"),
lty=c(1:6,1,2),cex=0.5)

```

```

plot(c0.values, lambda2.coperture_N, type="l", col="blue", xlab="c0",
ylab="vera copertura", main="lambda2 - n=100, m=20", ylim=c(0,1))
lines(c0.values, lambda2.coperture_t3, col="red", lty=2)
lines(c0.values, lambda2.coperture_gh1, col="orange", lty=3)
lines(c0.values, lambda2.coperture_gh2, col="green", lty=4)
lines(c0.values, lambda2.coperture_lN, col="brown", lty=5)
lines(c0.values, lambda2.coperture_chi3, col="purple", lty=6)
lines(c0.values, lambda2.coperture_beta, col="pink", lty=1)
lines(c0.values, lambda2.coperture_mist, col="cyan", lty=2)
abline(h=0.95)
legend(0.5, 0.33, legend=c("N(0,1)", "t(3)", "Tukey(0.25,0.05)", "Tukey(0.5,0.05)",
"log-N(0,1)", "chi(3)", "Beta(2,4)", "Mistura"),
col=c("blue", "red", "orange", "green", "brown", "purple", "pink", "cyan"),
lty=c(1:6,1), cex=0.5)

non.converge.stima_0.5<-c(N_0.5$non.converge.stima, t3_0.5$non.converge.stima,
gh1_0.5$non.converge.stima, gh2_0.5$non.converge.stima,
lN_0.5$non.converge.stima, chi3_0.5$non.converge.stima,
beta_0.5$non.converge.stima, mist_0.5$non.converge.stima)

non.converge.ic_0.5<-c(N_0.5$non.converge.ic, t3_0.5$non.converge.ic,
gh1_0.5$non.converge.ic, gh2_0.5$non.converge.ic,
lN_0.5$non.converge.ic, chi3_0.5$non.converge.ic,
beta_0.5$non.converge.ic, mist_0.5$non.converge.ic)

non.converge.stima_1<-c(N_1$non.converge.stima, t3_1$non.converge.stima,
gh1_1$non.converge.stima, gh2_1$non.converge.stima,
lN_1$non.converge.stima, chi3_1$non.converge.stima,
beta_1$non.converge.stima, mist_1$non.converge.stima)

non.converge.ic_1<-c(N_1$non.converge.ic, t3_1$non.converge.ic,
gh1_1$non.converge.ic, gh2_1$non.converge.ic,
lN_1$non.converge.ic, chi3_1$non.converge.ic,
beta_1$non.converge.ic, mist_1$non.converge.ic)

non.converge.stima_sqrt2<-c(N_sqrt2$non.converge.stima, t3_sqrt2$non.converge.stima,
gh1_sqrt2$non.converge.stima,
gh2_sqrt2$non.converge.stima,
lN_sqrt2$non.converge.stima,
chi3_sqrt2$non.converge.stima,
beta_sqrt2$non.converge.stima,
mist_sqrt2$non.converge.stima)

non.converge.ic_sqrt2<-c(N_sqrt2$non.converge.ic, t3_sqrt2$non.converge.ic,
gh1_sqrt2$non.converge.ic, gh2_sqrt2$non.converge.ic,
lN_sqrt2$non.converge.ic, chi3_sqrt2$non.converge.ic,
beta_sqrt2$non.converge.ic, mist_sqrt2$non.converge.ic)

non.converge.stima_2<-c(N_2$non.converge.stima, t3_2$non.converge.stima,
gh1_2$non.converge.stima, gh2_2$non.converge.stima,
lN_2$non.converge.stima, chi3_2$non.converge.stima,
beta_2$non.converge.stima, mist_2$non.converge.stima)

non.converge.ic_2<-c(N_2$non.converge.ic, t3_2$non.converge.ic,
gh1_2$non.converge.ic, gh2_2$non.converge.ic,

```

```
lN_2$non.converge.ic,chi3_2$non.converge.ic,
beta_2$non.converge.ic,mist_2$non.converge.ic)

non.converge.stima_3<-c(N_3$non.converge.stima,t3_3$non.converge.stima,
gh1_3$non.converge.stima,gh2_3$non.converge.stima,
lN_3$non.converge.stima,chi3_3$non.converge.stima,
beta_3$non.converge.stima,mist_3$non.converge.stima)

non.converge.ic_3<-c(N_3$non.converge.ic,t3_3$non.converge.ic,
gh1_3$non.converge.ic,gh2_3$non.converge.ic,
lN_3$non.converge.ic,chi3_3$non.converge.ic,
beta_3$non.converge.ic,mist_3$non.converge.ic)

non.converge.stima_0.5
non.converge.stima_1
non.converge.stima_sqrt2
non.converge.stima_2
non.converge.stima_3
non.converge.ic_0.5
non.converge.ic_1
non.converge.ic_sqrt2
non.converge.ic_2
non.converge.ic_3
```

E.2 Funzione gradiente

```

rm(list=ls())
library(mvtnorm)
library(nlme)

#Si simula ora un dataset con la struttura di quelli dello studio di simulazione, ma
#considerando solamente intercetta casuale.

n=50
m=10

beta0=2
beta1=1
beta2=1
beta=c(beta0,beta1,beta2)

a<-rnorm(n) #a contiene le intercette casuali per le n unita'
var.component=1 #vero valore della componente di varianza
a<-rep(a,each=m) #uniformo intercette casuali al formato lungo
id<-rep(c(1:n),each=m) #id e' l'etichetta dell'unita' per ogni osservazione
x1<-rep(seq(-1,1,length=m),n) #seconda colonna per ogni unita' valori equispaziati tra
#-1 e 1
x2<-runif(n*m,-1,1) #terza colonna valori generati indipendentemente da una U(-1,1)
epsilon<-rnorm(n*m) #errori casuali. Si assume qui sempre ipotesi di indipendenza
#tra errori casuali relativi a osservazioni distinte anche sulla stessa unita'
sigma2.epsilon=1 #vero valore della varianza within-cluster
theta=c(beta,var.component,sigma2.epsilon) #vero valore del parametro del modello
y<-beta0+a+beta1*x1+beta2*x2+epsilon #y e' il vettore di lunghezza n*m che
#contiene le risposte simulate.

X<-cbind(rep(1,n*m),x1,x2) #matrice del modello

matsplitter<-function(M, r, c) {
  rg <- (row(M)-1)%/%r+1
  cg <- (col(M)-1)%/%c+1
  rci <- (rg-1)*max(cg) + cg
  N <- prod(dim(M))/r/c
  cv <- unlist(lapply(1:N, function(x) M[rci==x]))
  dim(cv)<-c(r,c,N)
  cv
}

XX<-matsplitter(X,m,3)

#XX[, ,i] e' la matrice (m x p) delle esplicative per l'unita' i, X_i
#Serve per calcolare le densita' marginale e condizionata per l'unita' i

#Si consideri ora solo intercetta casuale: le Z_i sono semplicemente
#vettori di uni.
Z_i<-rep(1,m)

#Stimo il modello:
library(nlme)
dati.lme<-lme(y~x1+x2,random=~1|id,method="ML")

```

```

summary(dati.lme)

beta.hat0<-dati.lme$coefficients$fixed #beta stimati
sigma2.epsilon.hat0<-(dati.lme$sigma)^2 #stima della varianza within cluster
var.component.hat0<-as.numeric(VarCorr(dati.lme)[1,1]) #stima della componente
#di varianza
theta.hat0<-c(beta.hat0,sigma2.epsilon.hat0,var.component.hat0) #stima del parametro
theta.hat0

conditional.density<-function(x,i,u,theta.hat) { #f_i(y_i/u) densita' condizionata
  #per l'unita' i fissata l'intercetta casuale a u, data la stima del parametro
  #theta.hat
  #y_i/u~Nm(X_i%*%beta+Z_i%*%u , sigma^2_epsilon*Im)
  dmvnorm(x,mean=XX[, ,i]%*%theta.hat[1:3]+Z_i*u,sigma=theta.hat[5]*diag(m))
}
#theta.hat[1:3] sono le stime di beta0,beta1,beta2, theta.hat[5] e' la stima
#di sigma2.epsilon

marginal.density<-function(x,i,theta.hat) { #f_i(y_i) densita' marginale dell'unita' i
  #y_i~Nm(X_i%*%beta , Z_i%*%Sigma_u%*%t(Z_i)+sigma^2_epsilon*Im)
  dmvnorm(x,mean=XX[, ,i]%*%theta.hat[1:3] ,
  sigma=theta.hat[4]*rep(1,m)%*%t(rep(1,m))+theta.hat[5]*diag(m))
}

gradient<-function(u,theta.hat) { #funzione gradiente, calcolata nel punto u
  rapporti<-rep(NA,n)
  for (i in 1:n) {
    y_i<-y[(m*i-m+1):(m*i)] #vettore osservato y_i
    rapporti[i]=conditional.density(y_i,i,u,theta.hat)/marginal.density(y_i,i,theta.hat)
  }
  mean(rapporti)
}

grid.values<-seq(-2.5,2.5,length=100) #griglia di valori in cui si calcola la
#funzione gradiente
gradient.values0<-sapply(grid.values,FUN=function(x) gradient(x,theta.hat0))
plot(grid.values,gradient.values0,type="l",ylim=c(0,2),xlab="",ylab="gradiente")
abline(h=1,lty=2)

#I risultati sono coerenti con le aspettative; anche riprovando piu' volte, si
#ottiene sempre una funzione gradiente che "balla" attorno ad 1, senza grandi
#spostamenti, e dunque suggerisce corretta specificazione per la distribuzione
#degli effetti casuali (cosa che effettivamente e', le intercette casuali sono
# generate da una normale standard).
#Per valori di u molto piccoli o molto grandi (maggiori di 2 in valore assoluto)
#il gradiente va a volte a 0 e a volte assume valori grandi: tuttavia queste zone
#del supporto della distribuzione assunta G() sono associate a densita' molto basse,
#per cui comportamenti anomali sono "accettabili". Interessa piu' guardare a
#dove si concentra principalmente la massa di probabilita' di G (si puo' dire
#l'intervallo [-2,2]).
#Si riprova ora sotto misspecificazione per vedere cosa dice la funzione gradiente.
#Si generano le intercette casuali da una log-normale (0,1) standardizzata,
#distribuzione che nello studio di simulazione si e' avuto modo di vedere che porti
# ad avere considerevoli distorsioni nelle stime e sottocoperture negli intervalli.

```

```

r_lN<-function(n) {
  1/sqrt(exp(1)*(exp(1)-1))*rlnorm(n)-exp(0.5)/sqrt(exp(1)*(exp(1)-1))
} #algoritmo per generare da una logN(0,1) standardizzata

a<-r_lN(n)
a<-rep(a,each=m)
x2<-runif(n*m,-1,1)
epsilon<-rnorm(n*m)
y<-beta0 + a + beta1*x1 + beta2*x2+epsilon

dati.lme<-lme(y~x1+x2,random=~1|id,method="ML")
summary(dati.lme)
beta.hat1<-dati.lme$coefficients$fixed #beta stimati
sigma2.epsilon.hat1<-(dati.lme$sigma)^2 #stima della varianza within cluster
var.component.hat1<-as.numeric(VarCorr(dati.lme)[1,1]) #stima della varianza
theta.hat1<-c(beta.hat1,sigma2.epsilon.hat1,var.component.hat1)
gradient.values1<-sapply(grid.values,function(x) gradient(x,theta.hat1))
plot(grid.values,gradient.values1,type="l",ylim=c(0,2),xlab="",ylab="gradiente")
abline(h=1,lty=2)
#funzione gradiente mostra ampi allontanamenti da 1, a indicazione che c'e'
#misspecificazione. Segnala inoltre che e' necessario prendere una specificazione
#alternativa H() che rispetto a G() abbia maggiore massa di probabilita' attorno
#a -0.5 e minore massa di probabilita' in [-2,1] e [1,2].

curve(dnorm(x),-2.5,2.5,xlab="",ylab="",ylim=c(0,1.5))
curve(sqrt(exp(1)*(exp(1)-1))*dlnorm(sqrt(exp(1)*(exp(1)-1))*x+exp(0.5),0,1),-5,5,
xlab="",ylab="",col="red",lty=2,add=T)
legend(0.8,1.5,legend=c("distribuzione assunta",
"vera distribuzione"),col=c("black","red"),
lty=c(1,2),cex=0.6)
#andando a vedere come sono fatte le due distribuzioni (vera e assunta) e'
#effettivamente cio' che si dovrebbe fare. La funzione gradiente e' uno strumento
#di diagnosi valido.

```

E.3 Statistica test di Drikvandi et al. (2017)

```
K<-1000
nodes<-rep(NA,K)

for(i in 1:K) {
  nodes[i]=(2*i-1)/(2*K)
}

b<-qnorm(nodes)

T.stat<-function(theta.hat) {
  gradient.values=rep(NA,K) #vettore che conterra' il valore osservato della
#funzione gradiente per ogni nodo
  for(i in 1:K) {
    gradient.values[i]=gradient(b[i],theta.hat)
    if (i %% 10 == 0) {
      print(i) #tengo il passo del punto nel ciclo
    }
  }
  1/K * sum((gradient.values-1)^2)
}

T.oss0<-T.stat(theta.hat0) #valore osservato della stat. test nel primo caso
#in cui non c'e' misspecificazione
T.oss1<-T.stat(theta.hat1) #valore osservato quando invece c'e' misspecificazione

T.oss0
T.oss1 #la stat. test, che misura la distanza della funzione gradiente da 1,
#e' piu' grande nel secondo caso.
```

Bibliografia

- AGRESTI, A., CAFFO, B. & OHMAN-STRICKLAND, P. (2004). Examples in which misspecification of a random effects distribution reduces efficiency, and possible remedies. *Computational Statistics & Data Analysis* **47**, 639–653.
- ALONSO, A., LITIÈRE, S. & MOLENBERGHS, G. (2008). A family of tests to detect misspecifications in the random-effects structure of generalized linear mixed models. *Computational Statistics & Data Analysis* **52**, 4474–4486.
- BATES, D., MÄCHLER, M., BOLKER, B. & WALKER, S. (2015). *lme4: Linear Mixed-Effects Models using 'Eigen' and S4*. R package version 1.1-30.
- CHEN, J., ZHANG, D. & DAVIDIAN, M. (2002). A monte carlo em algorithm for generalized linear mixed models with flexible random effects distribution. *Biostatistics* **3**, 347–360.
- DRIKVANDI, R., VERBEKE, G. & MOLENBERGHS, G. (2017). Diagnosing misspecification of the random-effects distribution in mixed models. *Biometrics* **73**, 63–71.
- GENZ, A., BRETZ, F., MIWA, T., MI, X., LEISCH, F., SCHEIPL, F. & HOTHORN, T. (2021). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.1-3.
- HE, Y. & RAGHUNATHAN, T. E. (2006). Tukey's gh distribution for multiple imputation. *The American Statistician* **60**, 251–256.
- HUI, F. K., MÜLLER, S. & WELSH, A. H. (2021). Random effects misspecification can have severe consequences for random effects inference in linear mixed models. *International Statistical Review* **89**, 186–206.
- LIANG, K.-Y. & ZEGER, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika* **73**, 13–22.

- LITIÈRE, S., ALONSO, A. & MOLENBERGHS, G. (2008). The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in Medicine* **27**, 3125–3144.
- MCCULLOCH, C. E. & NEUHAUS, J. M. (2011a). Misspecifying the shape of a random effects distribution: Why getting it wrong may not matter. *Statistical Science* **26**, 388 – 402.
- MCCULLOCH, C. E. & NEUHAUS, J. M. (2011b). Prediction of random effects in linear and generalized linear models under model misspecification. *Biometrics* **67**, 270–279.
- NEUHAUS, J. M., MCCULLOCH, C. E. & BOYLAN, R. (2013). Estimation of covariate effects in generalized linear mixed models with a misspecified distribution of random intercepts and slopes. *Statistics in medicine* **32**, 2419–2429.
- PINHEIRO, J., BATES, D., DEBROY, S., SARKAR, D., HEISTERKAMP, S., VAN WILIGEN, B. & MAINTAINER, R. (2017). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-158.
- SALVAN, A., SARTORI, N. & PACE, L. (2020). *Modelli Lineari Generalizzati*. Milano: Springer-Verlag Italia.
- SARTORI, N. & SEVERINI, T. A. (2004). Conditional likelihood inference in generalized linear mixed models. *Statistica Sinica* **14**, 349–360.
- TCHETGEN, E. J. & COULL, B. A. (2006). A diagnostic test for the mixing distribution in a generalised linear mixed model. *Biometrika* **93**, 1003–1010.
- TUKEY, J. W. (1977). Modern techniques in data analysis. NSF-sponsored regional research conference at Southeastern Massachusetts University, North Dartmouth, MA.
- VERBEKE, G. & LESAFFRE, E. (1997). The effect of misspecifying the random-effects distribution in linear mixed models for longitudinal data. *Computational Statistics & Data Analysis* **23**, 541–556.
- VERBEKE, G. & MOLENBERGHS, G. (2013). The gradient function as an exploratory goodness-of-fit assessment of the random-effects distribution in mixed models. *Biostatistics* **14**, 477–490.
- WHITE, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica* **50**, 1–25.

