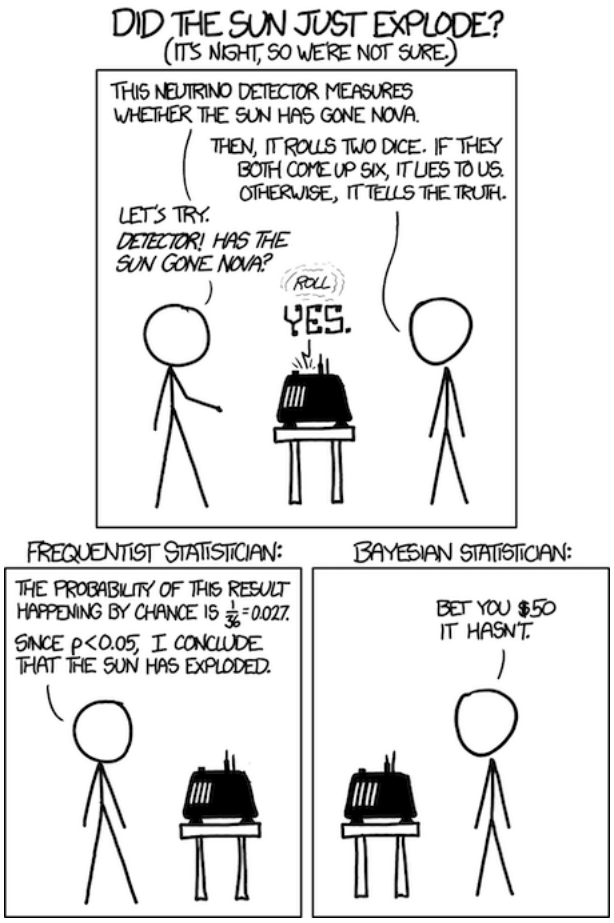
Corrado Caudek

Data Science per psicologi

Psicometria – AA 2021/2022



Indice

Elenco delle figure **vii**

Elenco delle tabelle **ix**

Prefazione **xi**

I Il modello lineare **1**

1 Introduzione **3**
1.1 La funzione lineare 3
1.2 Una media per ciascuna osservazione 5
1.2.1 Relazione lineare tra la media $y \mid x$ e il predittore 6
1.2.2 Il modello lineare 7

2 Modello lineare in Stan **9**
2.1 Linguaggio Stan 9
2.2 Interpretazione dei parametri 17
2.2.1 Centrare i predittori 18

3 Inferenza sul modello lineare **21**
3.1 Rappresentazione grafica dell'incertezza della stima . . . 21
3.2 Intervalli di credibilità 24
3.2.1 Quale soglia usare? 26
3.3 Test di ipotesi 27
3.4 Modello lineare robusto 28

4 Adattare il modello lineare ai dati **33**
4.1 Minimi quadrati 33
4.1.1 Stima della deviazione standard dei residui σ . . 34
4.2 Calcolare la somma dei quadrati 34

Elenco delle figure

1.1 La funzione lineare $y = a + bx$. 5

Elenco delle tabelle

Prefazione

Data Science per psicologi contiene il materiale delle lezioni dell'insegnamento di *Psicometria B000286* (A.A. 2021/2022) rivolto agli studenti del primo anno del Corso di Laurea in Scienze e Tecniche Psicologiche dell'Università degli Studi di Firenze. *Psicometria* si propone di fornire agli studenti un'introduzione all'analisi dei dati in psicologia. Le conoscenze/competenze che verranno sviluppate in questo insegnamento sono quelle della Data science, ovvero un insieme di conoscenze/competenze che si pongono all'intersezione tra statistica (ovvero, richiedono la capacità di comprendere teoremi statistici) e informatica (ovvero, richiedono la capacità di sapere utilizzare un software).

La psicologia e la Data science

Sembra sensato spendere due parole su un tema che è importante per gli studenti: quello indicato dal titolo di questo Capitolo. È ovvio che agli studenti di psicologia la statistica non piace. Se piacesse, forse studierebbero Data science e non psicologia; ma non lo fanno. Di conseguenza, gli studenti di psicologia si chiedono: "perché dobbiamo perdere tanto tempo a studiare queste cose quando in realtà quello che ci interessa è tutt'altro?" Questa è una bella domanda.

C'è una ragione molto semplice che dovrebbe farci capire perché la Data science è così importante per la psicologia. Infatti, a ben pensarci, la psicologia è una disciplina intrinsecamente statistica, se per statistica intendiamo quella disciplina che studia la variazione delle caratteristiche degli individui nella popolazione. La psicologia studia *gli individui* ed è proprio la variabilità inter- e intra-individuale ciò che vogliamo descrivere e, in certi casi, predire. In questo senso, la psicologia è molto diversa dall'ingegneria, per esempio. Le proprietà di un determinato ponte sotto certe condizioni, ad esempio, sono molto simili a quelle di un altro pon-

te, sotto le medesime condizioni. Quindi, per un ingegnere la statistica è poco importante: le proprietà dei materiali sono unicamente dipendenti dalla loro composizione e restano costanti. Ma lo stesso non può dirsi degli individui: ogni individuo è unico e cambia nel tempo. E le variazioni tra gli individui, e di un individuo nel tempo, sono l'oggetto di studio proprio della psicologia: è dunque chiaro che i problemi che la psicologia si pone sono molto diversi da quelli affrontati, per esempio, dagli ingegneri. Questa è la ragione per cui abbiamo tanto bisogno della Data science in psicologia: perché la Data science ci consente di descrivere la variazione e il cambiamento. E queste sono appunto le caratteristiche di base dei fenomeni psicologici.

Sono sicuro che, leggendo queste righe, a molti studenti sarà venuta in mente la seguente domanda: perché non chiediamo a qualche esperto di fare il "lavoro sporco" (ovvero le analisi statistiche) per noi, mentre noi (gli psicologi) ci occupiamo solo di ciò che ci interessa, ovvero dei problemi psicologici slegati dai dettagli "tecnici" della Data science? La risposta a questa domanda è che non è possibile progettare uno studio psicologico sensato senza avere almeno una comprensione rudimentale della Data science. Le tematiche della Data science non possono essere ignorate né dai ricercatori in psicologia né da coloro che svolgono la professione di psicologo al di fuori dell'Università. Infatti, anche i professionisti al di fuori dall'università non possono fare a meno di leggere la letteratura psicologica più recente: il continuo aggiornamento delle conoscenze è infatti richiesto dalla deontologia della professione. Ma per potere fare questo è necessario conoscere un bel po' di Data science! Basta aprire a caso una rivista specialistica di psicologia per rendersi conto di quanto ciò sia vero: gli articoli che riportano i risultati delle ricerche psicologiche sono zeppi di analisi statistiche e di modelli formali. E la comprensione della letteratura psicologica rappresenta un requisito minimo nel bagaglio professionale dello psicologo.

Le considerazioni precedenti cercano di chiarire il seguente punto: la Data science non è qualcosa da studiare a malincuore, in un singolo insegnamento universitario, per poi poterla tranquillamente dimenticare. Nel bene e nel male, gli psicologi usano gli strumenti della Data science in tantissimi ambiti della loro attività professionale: in particolare quando costruiscono, somministrano e interpretano i test psicometrici. È dunque chiaro che possedere delle solide basi di Data science è un tassello imprescindibile del bagaglio professionale dello psicologo. In questo insegnamento verranno trattati i temi base della Data science e verrà

adottato un punto di vista bayesiano, che corrisponde all'approccio più recente e sempre più diffuso in psicologia.

Come studiare

Il giusto metodo di studio per prepararsi all'esame di Psicometria è quello di seguire attivamente le lezioni, assimilare i concetti via via che essi vengono presentati e verificare in autonomia le procedure presentate a lezione. Incoraggio gli studenti a farmi domande per chiarire ciò che non è stato capito appieno. Incoraggio gli studenti a utilizzare i forum attivi su Moodle e, soprattutto, a svolgere gli esercizi proposti su Moodle. I problemi forniti su Moodle rappresentano il livello di difficoltà richiesto per superare l'esame e consentono allo studente di comprendere se le competenze sviluppate fino a quel punto sono sufficienti rispetto alle richieste dell'esame.

La prima fase dello studio, che è sicuramente individuale, è quella in cui è necessario acquisire le conoscenze teoriche relative ai problemi che saranno presentati all'esame. La seconda fase di studio, che può essere facilitata da scambi con altri e da incontri di gruppo, porta ad acquisire la capacità di applicare le conoscenze: è necessario capire come usare un software (R) per applicare i concetti statistici alla specifica situazione del problema che si vuole risolvere. Le due fasi non sono però separate: il saper fare molto spesso ci aiuta a capire meglio.

Sviluppare un metodo di studio efficace

Avendo insegnato molte volte in passato un corso introduttivo di analisi dei dati ho notato nel corso degli anni che gli studenti con l'atteggiamento mentale che descriverò qui sotto generalmente ottengono ottimi risultati. Alcuni studenti sviluppano naturalmente questo approccio allo studio, ma altri hanno bisogno di fare uno sforzo per maturarlo. Fornisco qui sotto una breve descrizione del "metodo di studio" che, nella mia esperienza, è il più efficace per affrontare le richieste di questo insegnamento.

- Dedicate un tempo sufficiente al materiale di base, apparentemente facile; assicuratevi di averlo capito bene. Cercate le lacune nella vostra comprensione. Leggere presentazioni diverse dello stesso materiale (in libri o articoli diversi) può fornire nuove intuizioni.
- Gli errori che facciamo sono i nostri migliori maestri. Istintivamente cerchiamo di dimenticare subito i nostri errori. Ma il miglior modo di imparare è apprendere dagli errori che commettiamo. In questo senso, una soluzione corretta è meno utile di una soluzione sbagliata. Quando commettiamo un errore questo ci fornisce un'informazione importante: ci fa capire qual è il materiale di studio sul quale dobbiamo ritornare e che dobbiamo capire meglio.
- C'è ovviamente un aspetto "psicologico" nello studio. Quando un esercizio o problema ci sembra incomprensibile, la cosa migliore da fare è dire: "mi arrendo", "non ho idea di cosa fare!". Questo ci rilassa: ci siamo già arresi, quindi non abbiamo niente da perdere, non dobbiamo più preoccuparci. Ma non dobbiamo fermarci qui. Le cose "migliori" che faccio (se ci sono) le faccio quando non ho voglia di lavorare. Alle volte, quando c'è qualcosa che non so fare e non ho idea di come affontare, mi dico: "oggi non ho proprio voglia di fare fatica", non ho voglia di mettermi nello stato mentale per cui "in 10 minuti devo risolvere il problema perché dopo devo fare altre cose". Però ho voglia di *divertirmi* con quel problema e allora mi dedico a qualche aspetto "marginale" del problema, che so come affrontare, oppure considero l'aspetto più difficile del problema, quello che non so come risolvere, ma invece di cercare di risolverlo, guardo come altre persone hanno affrontato problemi simili, opppure lo stesso problema in un altro contesto. Non mi pongo l'obiettivo "risolvi il problema in 10 minuti", ma invece quello di farmi un'idea "generale" del problema, o quello di capire un caso più specifico e più semplice del problema. Senza nessuna pressione. Infatti, in quel momento ho deciso di non lavorare (ovvero, di non fare fatica). Va benissimo se "parto per la tangente", ovvero se mi metto a leggere del materiale che sembra avere poco a che fare con il problema centrale (le nostre intuizioni e la nostra curiosità solitamente ci indirizzano sulla strada giusta). Quando faccio così, molto spesso trovo la soluzione del problema che mi ero posto e, paradossalmente, la trovo in un tempo minore di quello che, in precedenza, avevo dedicato a "lavorare" al problema. Allora perché non faccio sempre così? C'è ovviamente l'aspetto dei "10 minuti" che non è sempre facile da dimenticare. Sotto pressione, possiamo solo agire in maniera automatica, ovvero possia-

mo solo applicare qualcosa che già sappiamo fare. Ma se dobbiamo imparare qualcosa di nuovo, la pressione è un impedimento.

- È utile farsi da soli delle domande sugli argomenti trattati, senza limitarsi a cercare di risolvere gli esercizi che vengono assegnati. Quando studio qualcosa mi viene in mente: "se questo è vero, allora deve succedere quest'altra cosa". Allora verifico se questo è vero, di solito con una simulazione. Se i risultati della simulazione sono quelli che mi aspetto, allora vuol dire che ho capito. Se i risultati sono diversi da quelli che mi aspettavo, allora mi rendo conto di non avere capito e ritorno indietro a studiare con più attenzione la teoria che pensavo di avere capito – e ovviamente mi rendo conto che c'era un aspetto che avevo frainteso. Questo tipo di verifica è qualcosa che dobbiamo fare da soli, in prima persona: nessun altro può fare questo al posto nostro.
- Non aspettatevi di capire tutto la prima volta che incontrate un argomento nuovo.[1] È utile farsi una nota mentalmente delle lacune nella vostra comprensione e tornare su di esse in seguito per carcare di colmarle. L'atteggiamento naturale, quando non capiamo i dettagli di qualcosa, è quello di pensare: "non importa, ho capito in maniera approssimativa questo punto, non devo preoccuparmi del resto". Ma in realtà non è vero: se la nostra comprensione è superficiale, quando il problema verrà presentato in una nuova forma, non riusciremo a risolverlo. Per cui i dubbi che ci vengono quando studiamo qualcosa sono il nostro alleato più prezioso: ci dicono esattamente quali sono gli aspetti che dobbiamo approfondire per potere migliorare la nostra preparazione.
- È utile sviluppare una visione d'insieme degli argomenti trattati, capire l'obiettivo generale che si vuole raggiungere e avere chiaro il contributo che i vari pezzi di informazione forniscono al raggiungimento di tale obiettivo. Questa organizzazione mentale del materiale di studio facilita la comprensione. È estremamente utile creare degli schemi di ciò che si sta studiando. Non aspettate che sia io a fornirvi un riepilogo di ciò che dovete imparare: sviluppate da soli tali schemi e tali riassunti.
- Tutti noi dobbiamo imparare l'arte di trovare le informazioni, non solo nel caso di questo insegnamento. Quando vi trovate di fronte a qualcosa che non capite, o ottenete un oscuro messaggio di errore da

[1] Ricordatevi inoltre che gli individui tendono a sottostimare la propria capacità di apprendere (Horn and Loewenstein, 2021).

un software, ricordatevi: “Google is your friend”!

Corrado Caudek
Marzo 2022

Parte I

Il modello lineare

1

Introduzione

Lo scopo della ricerca è trovare le associazioni tra le variabili e fare confronti fra le condizioni sperimentali. Nel caso della psicologia, il ricercatore vuole scoprire le leggi generali che descrivono le relazioni tra i costrutti psicologici e le relazioni che intercorrono tra i fenomeni psicologici e quelli non psicologici (sociali, economici, storici, ...). Abbiamo già visto come la correlazione di Pearson sia uno strumento adatto a questo scopo. Infatti, essa ci informa sulla direzione e sull'intensità della relazione lineare tra due variabili. Tuttavia, la correlazione non è sufficiente, in quanto il ricercatore ha a disposizione solo i dati di un campione, mentre vorrebbe descrivere la relazione tra le variabili nella popolazione. A causa della variabilità campionaria, le proprietà dei campioni sono necessariamente diverse da quelle della popolazione: ciò che si può osservare nella popolazione potrebbe non emergere nel campione e, al contrario, il campione manifesta caratteristiche che non sono necessariamente presenti nella popolazione. È dunque necessario chiarire, dal punto di vista statistico, il legame che intercorre tra le proprietà del campione e le proprietà della popolazione da cui esso è stato estratto. Il modello lineare utilizza la funzione matematica più semplice per descrivere la relazione fra due variabili, ovvero la funzione lineare. In questo Capitolo vedremo come si possa fare inferenza sulla relazione tra due variabili mediante il modello lineare bayesiano. Inizieremo a descrivere le proprietà geometriche della funzione lineare per poi utilizzare questa semplice funzione per costruire un modello statistico secondo un approccio bayesiano.

1.1 La funzione lineare

Iniziamo con un ripasso sulla funzione di lineare. Si chiama *funzione lineare* una funzione del tipo

$$f(x) = a + bx, \tag{1.1}$$

dove a e b sono delle costanti. Il grafico di tale funzione è una retta di cui il parametro b è detto *coefficiente angolare* e il parametro a è detto *intercetta* con l'asse delle y [infatti, la retta interseca l'asse y nel punto $(0, a)$, se $b \neq 0$].

Per assegnare un'interpretazione geometrica alle costanti a e b si consideri la funzione

$$y = bx. \tag{1.2}$$

Tale funzione rappresenta un caso particolare, ovvero quello della *proporzionalità diretta* tra x e y. Il caso generale della linearità

$$y = a + bx \tag{1.3}$$

non fa altro che sommare una costante a a ciascuno dei valori $y = bx$. Nella funzione lineare $y = a + bx$, se b è positivo allora y aumenta al crescere di x; se b è negativo allora y diminuisce al crescere di x; se $b = 0$ la retta è orizzontale, ovvero y non muta al variare di x.

Consideriamo ora il coefficiente b. Si consideri un punto x_0 e un incremento arbitrario ε come indicato nella figura 1.1. Le differenze $\Delta x = (x_0 + \varepsilon) - x_0$ e $\Delta y = f(x_0 + \varepsilon) - f(x_0)$ sono detti *incrementi* di x e y. Il coefficiente angolare b è uguale al rapporto

$$b = \frac{\Delta y}{\Delta x} = \frac{f(x_0 + \varepsilon) - f(x_0)}{(x_0 + \varepsilon) - x_0}, \tag{1.4}$$

indipendentemente dalla grandezza degli incrementi Δx e Δy. Il modo più semplice per assegnare un'interpretazione geometrica al coefficiente angolare (o pendenza) della retta è dunque quello di porre $\Delta x = 1$. In tali circostanze infatti $b = \Delta y$.

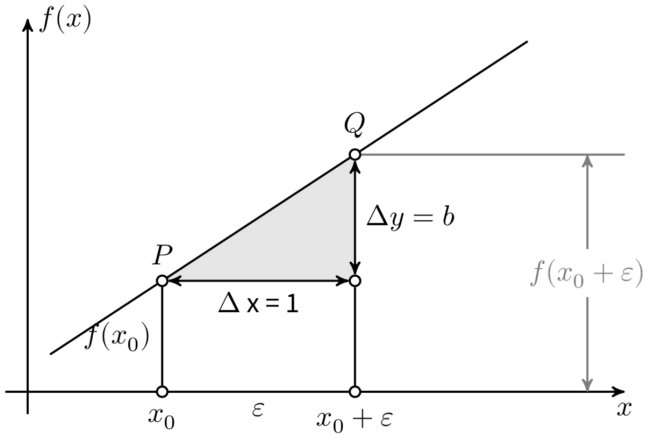


Figura 1.1: *La funzione lineare* $y = a + bx$.

1.2 Una media per ciascuna osservazione

In precedenza abbiamo visto come sia possibile stimare i parametri di un modello bayesiano nel quale le osservazioni sono indipendenti e identicamente distribuite secondo una densità gaussiana,

$$Y_i \overset{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma), \quad i = 1, \dots, n. \tag{1.5}$$

Il modello (1.5) assume che ogni Y_i sia la realizzazione di una v.c. descritta da una $\mathcal{N}(\mu, \sigma^2)$. Da un punto di vista bayesiano, si assegnano distribuzioni a priori ai parametri μ e σ, si genera la verosimiglianza in base ai dati osservati e, con queste informazioni, si generano le distribuzione a posteriori dei parametri (Gelman et al., 2020):

$$\begin{aligned} Y_i \mid \mu, \sigma &\overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2) \\ \mu &\sim \mathcal{N}(\mu_0, \tau^2) \\ \sigma &\sim \text{Cauchy}(x_0, \gamma) \end{aligned}$$

È comune però che vengano registrate altre variabili x_i che possono essere associate alla risposta di interesse y_i. La variabile x_i viene chiamata *predittore* (o variabile indipendente) in quanto il ricercatore è tipicamente

interessato a predire il valore y_i a partire da x_i. Come si può estende il modello (1.5) per lo studio della possibile relazione tra y_i e x_i?

Il modello (1.5) assume una media μ comune per ciascuna osservazione Y_i. Dal momento che desideriamo introdurre una nuova variabile x_i che assume un diverso valore per ciascuna osservazione y_i, il modello (1.5) può essere modificato in modo che la media comune μ venga sostituita da una media μ_i specifica a ciascuna osservazione i-esima:

$$Y_i \mid \mu_i, \sigma \overset{ind}{\sim} \mathcal{N}(\mu_i, \sigma), \quad i = 1, \dots, n. \tag{1.6}$$

Si noti che le osservazioni $Y_1, \dots, Y_n$ non sono più identicamente distribuite poiché hanno medie diverse, ma sono ancora indipendenti come indicato dalla notazione `ind` posta sopra il simbolo $\sim$ nella (1.6).

1.2.1 Relazione lineare tra la media $y \mid x$ e il predittore

L'approccio che consente di mettere in relazione un predittore x_i con la risposta Y_i è quello di assumere che la media di ciascuna Y_i, ovvero μ_i, sia una funzione lineare del predittore x_i. Una tale relazione lineare è scritta come

$$\mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n. \tag{1.7}$$

Nella (1.7), ciascuna x_i è una costante nota (ecco perché viene usata una lettera minuscola per la x) e β_0 e β_1 sono parametri incogniti. Questi parametri rappresentano l'intercetta e la pendenza della retta di regressione e sono delle variabili casuali.[1] L'inferenza bayesiana procede assegnando una distribuzione a priori a β_0 e a β_1 e si esegue l'inferenza riassumendo la distribuzione a posteriori di questi parametri.

Nel modello (1.7), la funzione lineare $\beta_0 + \beta_1 x_i$ è interpretata come il valore atteso della Y_i per ciascun valore x_i, mentre l'intercetta β_0 rappresenta il valore atteso della Y_i quando $x_i = 0$. Il parametro β_1 (pendenza) rappresenta invece l'aumento medio della Y_i quando x_i aumenta di un'unità. È importante notare che la relazione lineare (1.6) di parametri β_0 e β_1 descrive l'associazione tra *la media* μ_i e il predittore x_i. In altri termini, tale relazione lineare ci fornisce una predizione sul valore medio μ_i, non sul valore *effettivo* Y_i.

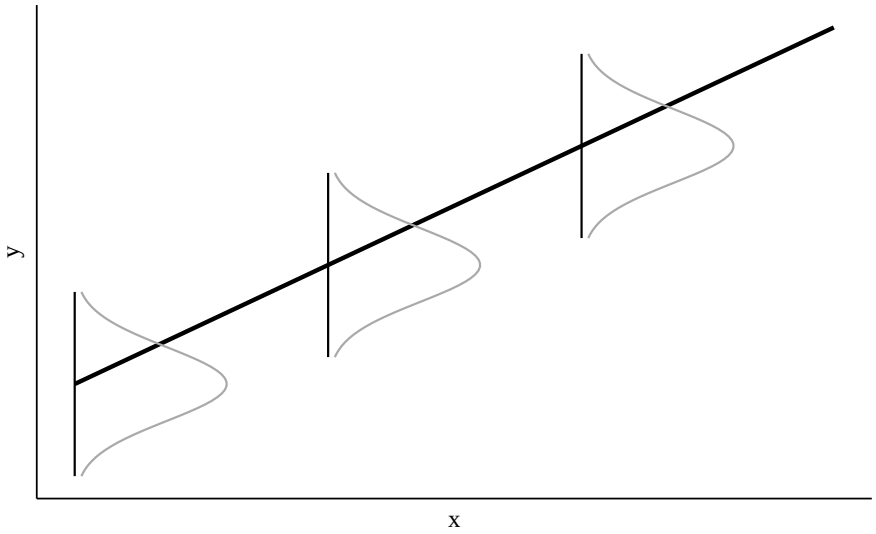
[1] Una notazione alternativa per tali parametri è α, β, anziché β_0, β_1.

1.2.2 Il modello lineare

Sostituendo la (1.7) nella (1.6) otteniamo il modello lineare:

$$Y_i \mid \beta_0, \beta_1, \sigma \overset{ind}{\sim} \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma), \quad i = 1, \dots, n. \tag{1.8}$$

Questo è un caso speciale del modello di campionamento Normale, dove le Y_i seguono indipendentemente una densità Normale con una media $(\beta_0 + \beta_1 x_i)$ specifica per ciascuna osservazione e con una deviazione standard (σ) comune a tutte le osservazioni. Poiché include un solo predittore (x), questo modello è comunemente chiamato *modello di regressione lineare semplice*.



Commenti e considerazioni finali

Il modello lineare semplice viene usato per descrivere la relazione tra due variabili e per determinare il segno e l'intensità di tale relazione. Inoltre, il modello lineare ci consente di prevedere il valore della variabile dipendente in base ai valori della variabile indipendente.

2

Modello lineare in Stan

Benché sia possibile una trattazione formale del modello lineare (per un'introduzione, si vedano le Appendici 4 e **??**), qui ci limiteremo ad esaminare l'uso del linguaggio probabilistico Stan per la stima dei parametri del modello. Vedremo anche come interpretare i risultati dell'analisi bayesiana.

2.1 Linguaggio Stan

Leggiamo in R il dataset `kidiq`:

```
library("rio")
df <- rio::import(here::here("data", "kidiq.dta"))
head(df)
#>   kid_score mom_hs mom_iq mom_work mom_age
#> 1        65      1 121.12        4      27
#> 2        98      1  89.36        4      25
#> 3        85      1 115.44        4      27
#> 4        83      1  99.45        3      25
#> 5       115      1  92.75        4      27
#> 6        98      0 107.90        1      18
```

Vogliamo descrivere l'associazione tra il QI dei figli e il QI delle madri mediante un modello lineare. Per farci un'idea del valore dei parametri, adattiamo il modello lineare ai dati mediante la procedura di massima verosimiglianza:

```
summary(lm(kid_score ~ mom_iq, data = df))
#>
```

```
#> Call:
#> lm(formula = kid_score ~ mom_iq, data = df)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -56.75 -12.07   2.22  11.71  47.69
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  25.7998     5.9174    4.36  1.6e-05 ***
#> mom_iq        0.6100     0.0585   10.42  < 2e-16 ***
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 18.3 on 432 degrees of freedom
#> Multiple R-squared:  0.201,  Adjusted R-squared:  0.199
#> F-statistic:  109 on 1 and 432 DF,  p-value: <2e-16
```

Sulla base delle informazioni precedenti, giungiamo alla seguente formulazione bayesiana del modello lineare:

$$
\begin{aligned}
y_i &\sim \mathcal{N}(\mu_i, \sigma) \\
\mu_i &= \beta_0 + \beta_1 x_i \\
\beta_0 &\sim \mathcal{N}(25, 10) \\
\beta_1 &\sim \mathcal{N}(0, 1) \\
\sigma &\sim \text{Cauchy}(18, 5)
\end{aligned}
$$

La prima riga definisce la funzione di verosimiglianza e le righe successive definiscono le distribuzioni a priori dei parametri. Il segno $\sim$ (tilde) si può leggere "si distribuisce come". La prima riga ci dice che ciascuna osservazione y_i è una variabile casuale che segue la distribuzione gaussiana di parametri μ_i e σ. La seconda riga specifica, in maniera deterministica, che ciascun μ_i è una funzione lineare di x_i, con parametri β_0 e β_1. Le due righe successive specificano le distribuzioni a priori per β_0 e β_1. La distribuzione a priori di β_0 è una distribuzione gaussiana di parametri $\mu_\alpha = 25$ e deviazione standard $\sigma_\alpha = 10$; la distribuzione a priori di β_1

è una distribuzione gaussiana standardizzata. L'ultima riga definisce la distribuzione a priori di σ, ovvero una Cauchy di parametri 18 e 5.

Dobbiamo ora specificare il modello bayesiano descritto sopra in linguaggio Stan[1]. Il codice Stan viene eseguito più velocemente se l'input è standardizzato così da avere una media pari a zero e una varianza unitaria.[3] Ponendo $y = (y_1, \dots, y_n)$ e $x = (x_1, \dots, x_n)$, il modello lineare può essere scritto come

$$y_i = \alpha + \beta x_i + \varepsilon_i,$$

dove

$$\varepsilon_i \sim \mathcal{N}(0, \sigma).$$

Seguendo la notazione del manuale Stan, i parametri del modello lineare sono qui denotati da α e β. Per eseguire la standardizzazione dei dati, è necessario centrare i dati, sottraendo da essi la media campionaria, per poi scalarli dividendo per la deviazione standard campionaria. Una singola osservazione u viene standardizzata dalla funzione z definita da

$$z_y(u) = \frac{u - \bar{y}}{\mathtt{sd}(y)}$$

dove la media $\bar{y}$ è

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i,$$

e la deviazione standard è

[1] Nella discussione che segue ripeto pari pari ciò che è riportato nel manuale del linguaggio Stan[2].

[3] Si noti un punto importante. Il fatto di standardizzare i dati fa in modo che le distribuzioni a priori sui parametri vadano espresse sulla scala delle v.c. normali standardizzate. Se centriamo sullo 0 tali distribuzioni a priori, con una deviazione standard dell'ordine di grandezza dell'unità, i discorsi sull'arbitrarietà delle distribuzioni a priori perdono di significato: nel caso di dati standardizzati le distribuzioni a priori formulate come indicato sopra sono distribuzioni debolmente informative il cui unico scopo è la regolarizzazione dei dati, ovvero di mantenere le inferenze in una gamma ragionevole di valori. L'uso di distribuzioni a priori debolmente informative contribuisce nel contempo a limitare l'influenza eccessiva delle osservazioni estreme (valori anomali). Il punto importante qui è che tali distribuzioni a priori non introducono alcuna distorsione sistematica nella stima a posteriori.

$$\texttt{sd} = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 \right)^{-\frac{1}{2}}.$$

La trasformata inversa è definita invertendo i due passaggi precedenti: la deviazione standard è usata per scalare i valori u e la media campionaria è usata per traslare la distribuzione dei valori u scalati:

$$z_y^{-1}(u) = \texttt{sd}(y)u + \bar{y}.$$

Consideriamo il seguente modello iniziale in linguaggio Stan:

```
modelString <- "
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  // priors
  alpha ~ normal(25, 10);
  beta ~ normal(0, 1);
  sigma ~ cauchy(18, 5);
  // likelihood
  for (n in 1:N)
    y[n] ~ normal(alpha + beta * x[n], sigma);
}
"
writeLines(modelString, con = "code/simpleregkidiq.stan")
```

La funzione `modelString()` registra una stringa di testo mentre `writeLines()` crea un file nell'indirizzo specificato. Tale file deve avere l'estensione `.stan`.

Modificando il codice precedente otteniamo il modello Stan per dati standardizzati. Il blocco `data` è identico a quello del caso precedente. I predittori e la risposta standardizzati sono definiti nel blocco `transformed data`. Per semplificare la notazione (e velocizzare l'esecuzione), nel blocco `model` l'istruzione di campionamento è espressa in forma vettorializzata: `y_std ~ normal(alpha_std + beta_std * x_std, sigma_std);`.

```
modelString <- "
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
transformed data {
  vector[N] x_std;
  vector[N] y_std;
  x_std = (x - mean(x)) / sd(x);
  y_std = (y - mean(y)) / sd(y);
}
parameters {
  real alpha_std;
  real beta_std;
  real<lower=0> sigma_std;
}
transformed parameters {
  vector[N] mu_std = alpha_std + beta_std * x_std;
}
model {
  alpha_std ~ normal(0, 1);
  beta_std ~ normal(0, 1);
  sigma_std ~ normal(0, 1);
  y_std ~ normal(mu_std, sigma_std);
}
generated quantities {
  // transform to the original data scale
  real alpha;
  real beta;
  real<lower=0> sigma;
  alpha = sd(y) * (alpha_std - beta_std * mean(x) / sd(x)) + mean(y);
```

```
  beta = beta_std * sd(y) / sd(x);
  sigma = sd(y) * sigma_std;
}
"
writeLines(modelString, con = "code/simpleregstd.stan")
```

Si noti che i parametri vengono rinominati per indicare che non sono i parametri "naturali", ma per il resto il modello è identico. Sono qui utilizzate distribuzioni a priori debolmente informative per i parametri `alpha` e `beta`.

I valori dei parametri sulla scala originale dei dati vengono calcolati nel blocco `generated quantities` e possono essere recuperati con un po' di algebra.

$$\begin{aligned} y_n &= \mathrm{z}_y^{-1}(\mathrm{z}_y(y_n)) \\ &= \mathrm{z}_y^{-1}\left(\alpha' + \beta' \mathrm{z}_x(x_n) + \epsilon'_n\right) \\ &= \mathrm{z}_y^{-1}\left(\alpha' + \beta'\left(\frac{x_n - \bar{x}}{\mathtt{sd}(x)}\right) + \epsilon'_n\right) \\ &= \mathtt{sd}(y)\left(\alpha' + \beta'\left(\frac{x_n - \bar{x}}{\mathtt{sd}(x)}\right) + \epsilon'_n\right) + \bar{y} \\ &= \left(\mathtt{sd}(y)\left(\alpha' - \beta'\frac{\bar{x}}{\mathtt{sd}(x)}\right) + \bar{y}\right) + \left(\beta'\frac{\mathtt{sd}(y)}{\mathtt{sd}(x)}\right) x_n + \mathtt{sd}(y)\epsilon'_n, \end{aligned} \tag{2.1}$$

da cui

$$\alpha = \mathtt{sd}(y)\left(\alpha' - \beta'\frac{\bar{x}}{\mathtt{sd}(x)}\right) + \bar{y}; \qquad \beta = \beta'\frac{\mathtt{sd}(y)}{\mathtt{sd}(x)}; \qquad \sigma = \mathtt{sd}(y)\sigma'.$$

Per svolgere l'analisi bayesiana sistemiamo i dati nel formato appropriato per Stan:

```
data_list <- list(
  N = length(df$kid_score),
  y = df$kid_score,
  x = df$mom_iq
)
```

La funzione `file.path()` ritorna l'indirizzo del file con il codice Stan:

```
file <- file.path("code", "simpleregstd.stan")
```

Il codice Stan può essere stampato usando il metodo `$print()`:

```
mod$print()
```

Prendendo come input un file contenente un programma Stan, la funzione `cmdstan_model()` ritorna un oggetto di classe `CmdStanModel`. In pratica, `CmdStan` traduce un programma Stan in C++ e crea un eseguibile compilato.

```
mod <- cmdstan_model(file)
```

L'indirizzo dell'eseguibile compilato viene ritornato da `$exe_file()`:

```
mod$exe_file()
```

Applicando il metodo `$sample()` ad un oggetto `CmdStanModel` eseguiamo il campionamento MCMC:

```
fit <- mod$sample(
  data = data_list,
  iter_sampling = 4000L,
  iter_warmup = 2000L,
  seed = SEED,
  chains = 4L,
  parallel_chains = 2L,
  refresh = 0,
  thin = 1
)
```

Al metodo `$sample()` possono essere passati molti argomenti. La pagina di documentazione è disponibile al seguente link[4].

[4] https://mc-stan.org/cmdstanr/reference/model-method-sample.html

Un sommario della distribuzione a posteriori per i parametri stimati si ottiene con il metodo `$summary()`, il quale chiama la funzione `summarise_draws()` del pacchetto `posterior`:

```
fit$summary(c("alpha", "beta", "sigma"))
#> # A tibble: 3 x 10
#>   variable   mean median     sd    mad     q5    q95
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
#> 1 alpha    25.9   25.8   6.02   6.02   16.0   35.8
#> 2 beta      0.609  0.609 0.0596 0.0603  0.511  0.707
#> 3 sigma    18.3   18.3   0.634  0.644  17.3   19.4
#> # ... with 3 more variables: rhat <dbl>,
#> #   ess_bulk <dbl>, ess_tail <dbl>
```

Da questo output possiamo valutare rapidamente la convergenza del modello osservando i valori di Rhat per ciascun parametro. Quando questi sono pari o vicini a 1, le catene hanno realizzato la convergenza. Ci sono molti altri test diagnostici, ma questo test è importante per Stan. Oppure è possibile usare:

```
fit$cmdstan_summary()
```

Le statistiche diagnostiche sono fornite dal metodo `$cmdstan_diagnose()`:

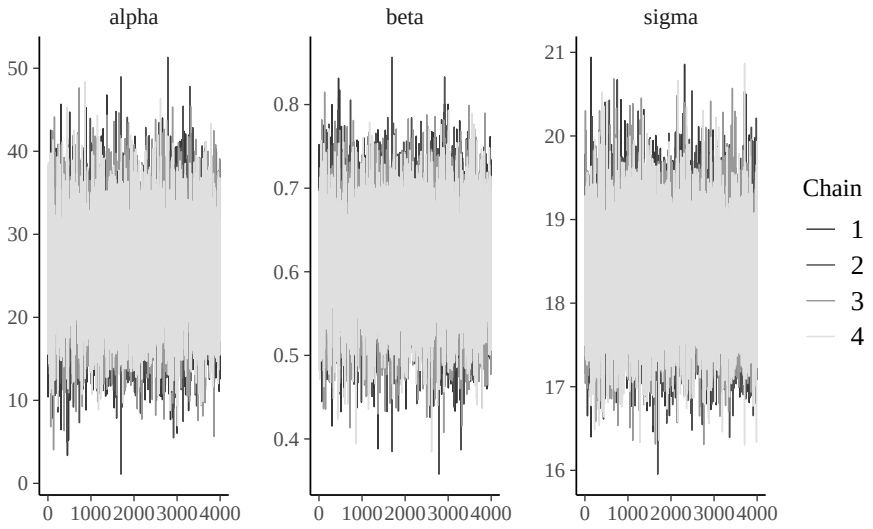
```
fit$cmdstan_diagnose()
```

È possibile creare un oggetto di classe `stanfit`

```
stanfit <- rstan::read_stan_csv(fit$output_files())
```

per poi utilizzare le funzioni del pacchetto `bayesplot`. Ad esempio:

```
stanfit %>%
  mcmc_trace(pars = c("alpha", "beta", "sigma"))
```



Infine, eseguendo la funzione `launch_shinystan(fit)`, è possibile analizzare oggetti di classe `stanfit` mediante le funzionalità del pacchetto `ShinyStan`.

2.2 Interpretazione dei parametri

Assegnamo ai parametri la seguente interpretazione.

- L'intercetta pari a 25.9 indica il QI medio dei bamini la cui madre ha un QI = 0. Ovviamente questo non ha alcun significato. Vedremo nel modello successivo come trasformare il modello in modo da potere assegnare all'intercetta un'interpretazione sensata.
- La pendenza di 0.61 indica che, all'aumentare di un punto del QI delle madri, il QI medio dei loro bambini aumenta di 0.61 unità. Se consideriamo la gamma di variazione del QI delle madri nel campione, il QI medio dei bambini cambia di 41 punti. Questo indica un sostanziale effetto del QI delle madri sul QI dei loro bambini: $(138.89 - 71.04) * 0.61 = 41.39$.
- Il parametro $\sigma = 18.3$ fornisce una stima della dispersione delle osservazioni attorno al valore predetto dal modello lineare, ovvero forni-

sce una stima della deviazione standard dei residui attorno al valore atteso del modello lineare.

2.2.1 Centrare i predittori

Per migliorare l'interpretazione dell'intercetta possiamo "centrare" la x, ovvero esprimere la x nei termini degli scarti dalla media: $x - \bar{x}$. In tali circostanze, la pendenza della retta specificata dal modello lineare resta immutata, ma l'intercetta corrisponde a $\mathbb{E}(y \mid x = \bar{x})$. Per ottenere questo risultato, modifichiamo i dati da passare a Stan:

```
data2_list <- list(
  N = length(df$kid_score),
  y = df$kid_score,
  x = df$mom_iq - mean(df$mom_iq)
)
```

Adattiamo il modello:

```
fit2 <- mod$sample(
  data = data2_list,
  iter_sampling = 4000L,
  iter_warmup = 2000L,
  seed = SEED,
  chains = 4L,
  parallel_chains = 2L,
  refresh = 0,
  thin = 1
)
```

Trasformiamo l'oggetto `fit` in un oggetto di classe `stanfit`:

```
stanfit <- rstan::read_stan_csv(fit2$output_files())
```

Le stime a posteriori dei parametri si ottengono con

```
fit2$summary(c("alpha", "beta", "sigma"))
#> # A tibble: 3 x 10
```

```
#>   variable   mean median     sd    mad     q5    q95
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
#> 1 alpha    86.8   86.8   0.876  0.871  85.4   88.2
#> 2 beta      0.609  0.609 0.0591 0.0589  0.513  0.707
#> 3 sigma    18.3   18.3   0.630  0.624  17.3   19.4
#> # ... with 3 more variables: rhat <dbl>,
#> #   ess_bulk <dbl>, ess_tail <dbl>
```

Si noti la nuova intercetta, ovvero 86.8. Questo valore indica il QI medio dei bambini le cui madri hanno un QI pari alla media del campione. Centrare i dati consente dunque di assegnare all'intercetta un'interpretazione utile.

Commenti e considerazioni finali

La presente discussione suggerisce che è conveniente standardizzare i dati prima di procedere con l'analisi. Ciò può essere fatto all'interno del codice Stan (come negli esempi di questo Capitolo), oppure prima di passare i dati a Stan. Se vengono usati dati standardizzati diventa poi facile utilizzare distribuzioni a priori debolmente informative per i parametri. Tali distribuzioni a priori hanno, come unico scopo, quello di regolarizzare i dati e di facilitare la stima dei parametri mediante MCMC.

3

Inferenza sul modello lineare

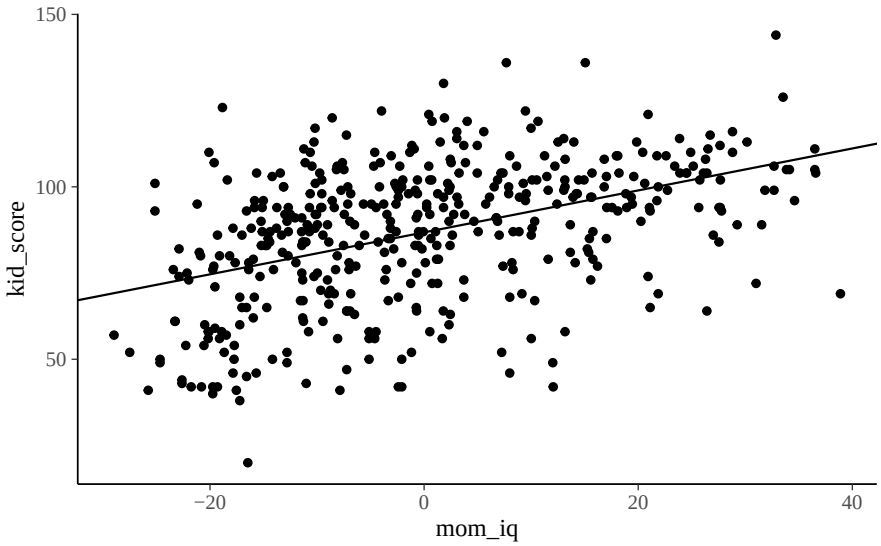
3.1 Rappresentazione grafica dell'incertezza della stima

Un primo modo per rappresentare l'incertezza dell'inferenza in un ottica bayesiana è quella di rappresentare graficamente la retta specificata dal modello lineare. Continuando con l'esempio descritto nel Capitolo precedente (ovvero, i dati `kid_score` e i valori `mom_iq` centrati), usando la funzione `rstan::read_stan_csv` leggiamo i file CSV generati da `cmdstan` e trasformiamo le stime a posteriori dei parametri in formato `stanfit`:

```
stanfit <- rstan::read_stan_csv(fit2$output_files())
posterior <- extract(stanfit)
```

Creiamo ora un diagramma a dispersione dei dati con sovrapposto il valore atteso della y:

```
tibble(
  kid_score = df$kid_score,
  mom_iq = df$mom_iq - mean(df$mom_iq)
) %>%
  ggplot(aes(mom_iq, kid_score)) +
  geom_point() +
  geom_abline(
    intercept = mean(posterior$alpha),
    slope = mean(posterior$beta)
  )
```



L'incertezza della stima della retta specifiata dal modello lineare può essere visualizzata tracciando molteplici rette, ciascuna delle quali definita da un diverso valore estratto a caso dalla distribuzione a posteriori dei parametri α e β. Per ottenere questo risultato dobbiamo estrarre le informazioni richieste dall'oggetto `stanfit` che abbiamo creato; usiamo, per esempio, le funzionalità di `tidybayes`:

```
tidybayes::get_variables(stanfit)
#>  [1] "alpha_std"     "beta_std"      "sigma_std"
#>  [4] "alpha"         "beta"          "sigma"
#>  [7] "lp__"          "accept_stat__" "treedepth__"
#> [10] "stepsize__"    "divergent__"   "n_leapfrog__"
#> [13] "energy__"
```

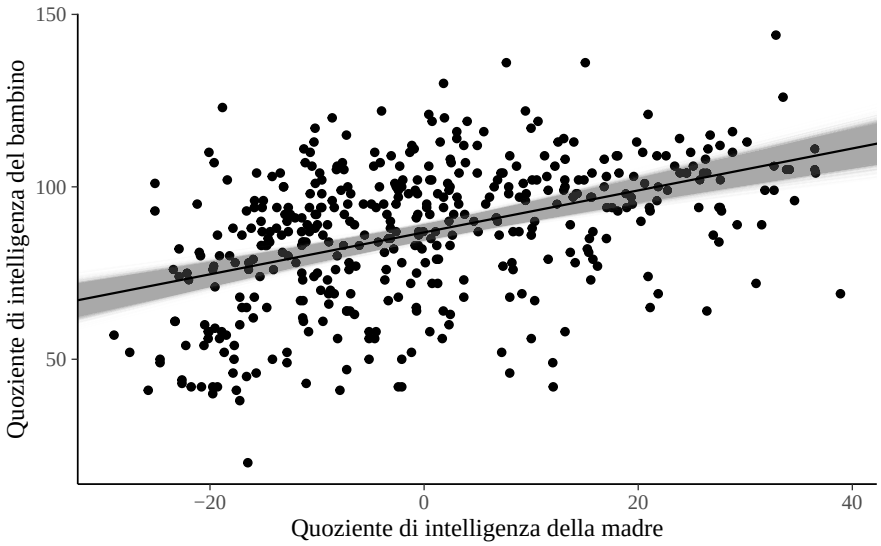
Creiamo un Dataframe in formato tidy (cioè, tale per cui le osservazioni stanno sulle righe e le variabili stanno sulle colonne) che contiene le stime a posteriori di α e β:

```
draws <- stanfit %>%
  spread_draws(beta, alpha)
```

```
draws %>%
  head(10)
#> # A tibble: 10 x 5
#>   .chain .iteration .draw  beta alpha
#>    <int>      <int> <int> <dbl> <dbl>
#> 1      1          1     1 0.632  88.4
#> 2      1          2     2 0.491  87.5
#> 3      1          3     3 0.717  85.9
#> 4      1          4     4 0.478  87.5
#> 5      1          5     5 0.610  86.4
#> 6      1          6     6 0.570  86.7
#> 7      1          7     7 0.623  87.0
#> 8      1          8     8 0.616  87.2
#> # ... with 2 more rows
```

Possiamo ora generare un diagramma a dispersione con `ggplot()`:

```
tibble(
  kid_score = df$kid_score,
  mom_iq = df$mom_iq - mean(df$mom_iq)
) %>%
  ggplot(aes(mom_iq, kid_score)) +
  geom_point() +
  geom_abline(
    data = draws, aes(intercept = alpha, slope = beta),
    size = 0.2, alpha = 0.01, color = "darkgray"
  ) +
  geom_abline(
    intercept = mean(posterior$alpha),
    slope = mean(posterior$beta)
  ) +
  labs(
    x = "Quoziente di intelligenza della madre",
    y = "Quoziente di intelligenza del bambino"
  )
```



Il grafico indica che le rette di regressione costruite estraendo a caso valori dalla distribuzione a posteriori dei parametri β_0 e β_1 tendono ad essere molto simili tra loro. Ciò significa che, relativamente alla dipendenza (lineare) del quoziente di intelligenza del bambino da quello della madra, la nostra incertezza è molto piccola.

3.2 Intervalli di credibilità

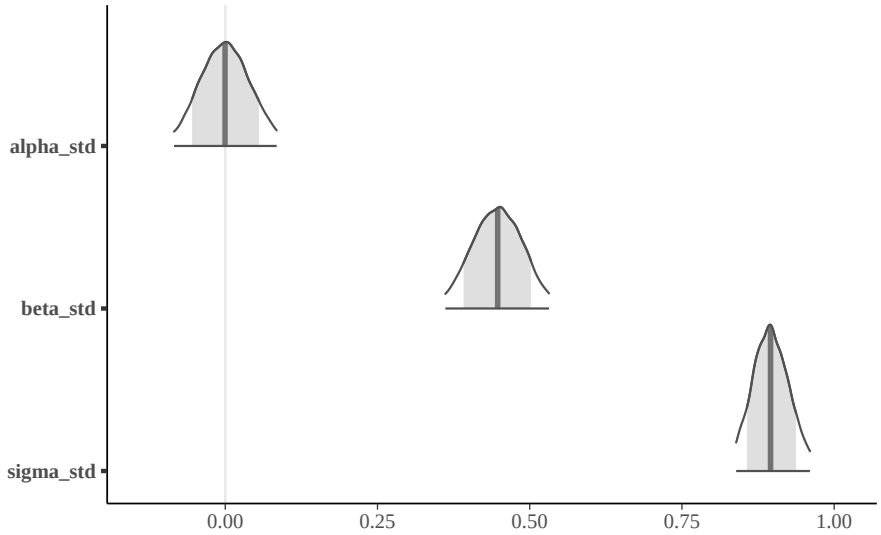
L'incertezza inferenziale sui parametri può anche essere descritta mediante gli *intervalli di credibilità*, ovvero gli intervalli che contengono la quota desiderata (es., il 95%) della distribuzione a posteriori. Per l'esempio che stiamo discutendo, gli intervalli di credibilità al 95% si ottengono nel modo seguente:

```
rstantools::posterior_interval(
  as.matrix(stanfit),
  prob = 0.95
)
#>                2.5%      97.5%
#> alpha_std   -0.08427    0.08442
#> beta_std     0.36137    0.53165
```

```
#> sigma_std    0.83903    0.96033
#> alpha       85.07713   88.52021
#> beta         0.49172    0.72343
#> sigma       17.12519   19.60111
#> lp__      -173.15908 -168.54400
```

Un grafico che, nel caso dei dati standardizzati, riporta l'intervallo di credibilità ai livelli di probabilità desiderati per i parametri α, β e σ si ottiene con l'istruzione

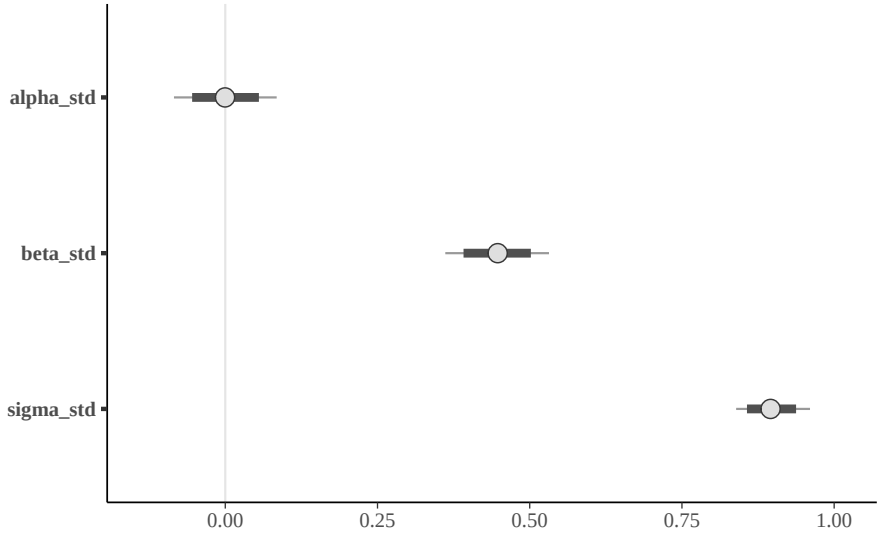
```
mcmc_areas(
  fit2$draws(c("alpha_std", "beta_std", "sigma_std")),
  prob = 0.8,
  prob_outer = 0.95
)
```



oppure nel modo nel modo seguente

```
stanfit %>%
  mcmc_intervals(
    pars = c("alpha_std", "beta_std", "sigma_std"),
```

```
    prob = 0.8,
    prob_outer = 0.95
  )
```

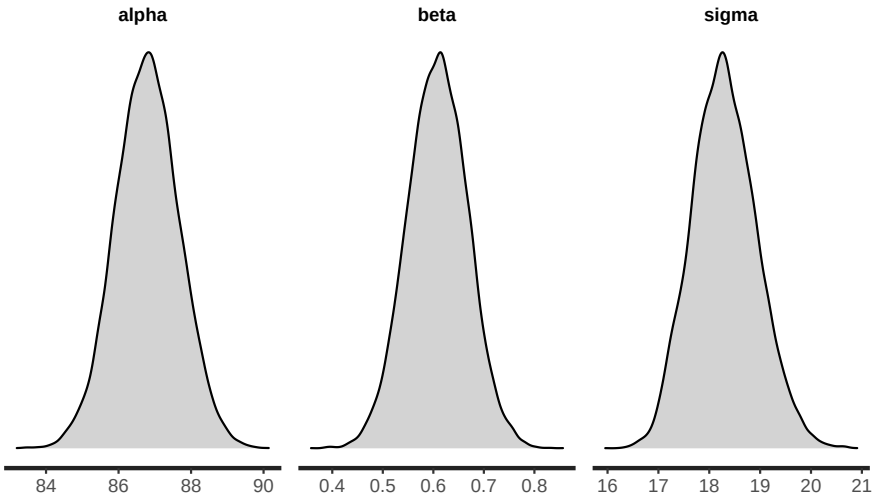


3.2.1 Quale soglia usare?

Non c'è niente di "magico" o necessario relativamente al livello di 0.95: il valore 0.95 è arbitrario. Sono possibili tantissime altre soglie per quantificare la nostra incertezza: alcuni ricercatori usano il livello di 0.89, altri quello di 0.5. Se l'obiettivo è quello di descrivere il livello della nostra incertezza relativamente alla stima del parametro, allora dobbiamo riconoscere che la nostra incertezza è descritta dall'*intera* distribuzione a posteriori. Per cui il metodo più semplice, più diretto e più completo per descrivere la nostra incertezza rispetto alla stima dei parametri è semplicemente quello di riportare graficamente *tutta* la distribuzione a posteriori. Una rappresentazione della distribuzione a posteriori dei parametri del modello dell'esempio si ottiene nel modo seguente:

```
rstan::stan_dens(
  stanfit,
  pars = c("alpha", "beta", "sigma"),
```

```
  fill = "lightgray"
)
```



3.3 Test di ipotesi

È facile valutare ipotesi direzionali usando Stan. Per esempio, la probabilità $Pr(\hat{\beta}_1 > 0)$ è

```
sum(posterior$beta > 0) / length(posterior$beta)
#> [1] 1
```

ovvero, la probabilità $Pr(\hat{\beta}_1 < 0)$ è

```
sum(posterior$beta < 0) / length(posterior$beta)
#> [1] 0
```

3.4 Modello lineare robusto

Spesso i ricercatori devono affrontare il problema degli outlier: in presenza di outlier, un modello statistico basato sulla distribuzione gaussiana produrrà delle stime distorte dei parametri, ovvero stime che non si generalizzano ad altri campioni di dati. Il metodo tradizionale per affrontare questo problema è quello di eliminare gli outlier prima di eseguire l'analisi statistica. Il problema di questo approccio, però, è che il criterio utilizzato per eliminare gli outlier, quale esso sia, è arbitrario; dunque, usando criteri diversi per la rimozione di outlier, i ricercatori finiscono per trovare risultati diversi.

Questo problema trova una semplice soluzione nell'approccio bayesiano. Il modello lineare che abbiamo dicusso finora ipotizza una specifica distribuzione degli errori, ovvero $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$. In un modello formulato in questi termini, la presenza di solo un valore anomalo e influente ha un effetto drammatico sulle stime dei parametri.

Per fare un esempio, introduciamo un singlo valore anomalo e influente nel set dei dati dell'esempio che stiamo discutendo:

```
df2 <- df
df2$kid_score[434] <- -500
df2$mom_iq[434] <- 140
```

Per comodità, calcoliamo le stime di α e β con il metodo dei minimi quadrati (tali stime sono simili a quelle che si otterrebbero con un modello bayesiano gaussiano che impiega distribuzioni a priori debolmente informative). Sappiamo che, nel campione originale di dati, $\hat{\beta} \approx 0.6$. In presenza di un solo outlier troviamo la stima di β viene drammaticamente ridotta:

```
lm(kid_score ~ mom_iq, data = df2) %>%
  coef()
#> (Intercept)      mom_iq
#>     49.1880      0.3626
```

In generale, però, non è necessario assumere $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$. È altrettanto valido un modello che ipotizza una diversa distribuzione di densità per

gli errori come, ad esempio, la distribuzione t di Student con un piccolo numero di gradi di libertà. Una caratteristica della t di Student è che le code della distribuzione contengono una massa di probabilità maggiore della distribuzione gaussiana. Ciò fornisce alla t di Student la possibilità di "rendere conto" della presenza di osservazioni lontane dalla media della distribuzione. In altri termini, se in modello lineare usiamo la t di Student quale distribuzione degli errori, la presenza di outlier avrà una minore influenza sulle stime dei parametri di quanto avvenga nel tradizionale modello lineare gaussiano.

Per verificare questa affermazione, modifichiamo il codice Stan usato in precedenza in modo tale da ipotizzare che y segua una distribuzione t di Student con un numero ν gradi di libertà stimato dal modello: `student_t(nu, mu, sigma)`.[1]

```
modelString <- "
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
transformed data {
  vector[N] x_std;
  vector[N] y_std;
  x_std = (x - mean(x)) / sd(x);
  y_std = (y - mean(y)) / sd(y);
}
parameters {
  real alpha_std;
  real beta_std;
  real<lower=0> sigma_std;
  real<lower=1> nu;    // degrees of freedom is constrained >1
}
model {
  alpha_std ~ normal(0, 1);
  beta_std ~ normal(0, 1);
  sigma_std ~ normal(0, 1);
```

[1] È equivalente scrivere $y_i = \mu_i + \varepsilon_i$, dove $\mu_i = \alpha + \beta x_i, \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon)$, oppure $y_i \sim \mathcal{N}(\mu_i, \sigma_\varepsilon)$.

```
  nu ~ gamma(2, 0.1);   // Juárez and Steel(2010)
  y_std ~ student_t(nu, alpha_std + beta_std * x_std, sigma_std);
}
generated quantities {
  real alpha;
  real beta;
  real<lower=0> sigma;
  alpha = sd(y) * (alpha_std - beta_std * mean(x) / sd(x))
          + mean(y);
  beta = beta_std * sd(y) / sd(x);
  sigma = sd(y) * sigma_std;
}
"
writeLines(modelString, con = "code/simpleregstdrobust.stan")
```

Costruiamo la lista dei dati usando il data.frame `df2` che include l'outlier:

```
data3_list <- list(
  N = length(df2$kid_score),
  y = df2$kid_score,
  x = df2$mom_iq - mean(df2$mom_iq)
)
```

Adattiamo il modello lineare robusto ai dati:

```
file <- file.path("code", "simpleregstdrobust.stan")
mod <- cmdstan_model(file)

fit4 <- mod$sample(
  data = data3_list,
  iter_sampling = 4000L,
  iter_warmup = 2000L,
  seed = SEED,
  chains = 4L,
  parallel_chains = 2L,
  refresh = 0,
  thin = 1
)
```

Se esaminiamo le stime dei parametri notiamo che la stima di β non è stata influenzata dalla presenza di un'osservazione anomala e influente:

```
fit4$summary(c("alpha", "beta", "sigma", "nu"))
#> # A tibble: 4 x 10
#>   variable   mean median     sd    mad     q5    q95
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
#> 1 alpha    87.8   87.8   0.901  0.898  86.3   89.3
#> 2 beta      0.602  0.602 0.0589 0.0587  0.505  0.699
#> 3 sigma    15.9   15.9   0.800  0.803  14.6   17.2
#> 4 nu        5.58   5.46  1.15   1.09    3.93   7.64
#> # ... with 3 more variables: rhat <dbl>,
#> #   ess_bulk <dbl>, ess_tail <dbl>
```

Il modello lineare robusto non risente dunque della presenza di outlier.

Commenti e considerazioni finali

Nell'approccio bayesiano possiamo rappresentare l'incertezza delle nostre credenze a posteriori in due modi: mediante la rappresentazione grafica dell'intera distribuzione a posteriori dei parametri o mediante l'uso degli intervalli di credibilità. Un bonus della discussione del presente Capitolo è quello di mostrare come il modello lineare tradizionale (che assume $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$) possa essere facilmente esteso nei termini di un modello robusto che offre una semplice soluzione al problema di ridurre l'effetto della presenza di osservazioni outlier.

4

Adattare il modello lineare ai dati

In questo Capitolo verranno esposte alcune nozioni matematiche che stanno alla base dell'inferenza sul modello lineare.

4.1 Minimi quadrati

Nel modello lineare classico, $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, i coefficienti β_0 e β_1 sono stimati in modo tale da minimizzare gli errori ε_i. Se il numero dei dati n è maggiore di 2, non è generalmente possibile trovare una retta che passi per tutte le osservazioni (x, y) (sarebbe $y_i = \beta_0 + \beta_1 x_i$, senza errori, per tutti i punti $i = 1, \dots, n$). L'obiettivo della stima dei minimi quadrati è quello di scegliere i valori $(\hat{\beta}_0, \hat{\beta}_1)$ che minimizzano la somma dei quadrati dei residui,

$$e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\,. \tag{4.1}$$

Distinguiamo tra i residui $e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$ e gli *errori* $\varepsilon_i = y_i - (\beta_0 + \beta_1 x_i)$. Il modello di regressione è scritto in termini degli errori, ma possiamo solo lavorare con i residui: non possiamo calcolare gli errori perché per farlo sarebbe necessario conoscere i parametri ignoti β_0 e β_1.

La somma dei residui quadratici (*residual sum of squares*) è

$$\text{RSS} = \sum_{i=1}^{n} (y_i = (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2\,. \tag{4.2}$$

I coefficienti $(\hat{\beta}_0, \hat{\beta}_1)$ che minimizzano RSS sono chiamati stime dei minimi quadrati, o minimi quadrati ordinari (*ordinari least squares*), o stime OLS.

4.1.1 Stima della deviazione standard dei residui σ

Nel modello lineare, gli errori ε_i provengono da una distribuzione con media 0 e deviazione standard σ: la media è zero per definizione (qualsiasi media diversa da zero viene assorbita nell'intercetta, β_0), e la deviazione standard degli errori può essere stimata dai dati. Un modo apparentemente naturale per stimare σ potrebbe essere quello di calcolare la deviazione standard dei residui, $\sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}$, ma questo approccio finisce per sottostimare σ. La correzione standard di questa sottostima consiste nel sostituire n con $n-2$ al denominatore (la sottrazione di 2 deriva dal fatto che il valore atteso del modello lineare è stato calcolato utilizzando i due coefficienti nel modello, l'intercetta e la pendenza, i quali sono stati stimati dai dati campionari – si dice che, in questo modo, abbiamo perso due gradi di libertà). Così facendo otteniamo

$$\hat{\sigma} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}\,. \tag{4.3}$$

Quando $n = 1$ o 2 l'equazione precedente è priva di significato, il che ha senso: con solo due osservazioni è possibile adattare esattamente una retta al diagramma di dispersione e quindi non c'è modo di stimare l'errore dai dati.

4.2 Calcolare la somma dei quadrati

Seguendo Solomon Kurz[1], creiamo una funzione per calcolare la somma dei quadrati per diversi valori di β_0 e β_1 che, per semplicità, qui verranno chiamati a e b:

```
rss <- function(x, y, a, b) {
  # x and y are vectors,
  # a and b are scalars
  resid <- y - (a + b * x)
```

[1] https://github.com/ASKurz/Working-through-Regression-and-other-stories/blob/main/08.Rmd

```
  return(sum(resid^2))
}
```

Per fare un esempio concreto useremo un famoso dataset chiamato `kidiq` (Gelman et al., 2020) che riporta i dati di un'indagine del 2007 su un campione di donne americane adulte e sui loro bambini di età compresa tra i 3 e i 4 anni. I dati sono costituiti da 434 osservazioni e 4 variabili:

- `kid_score`: QI del bambino; è il punteggio totale del *Peabody Individual Achievement Test* (PIAT) costituito dalla somma dei punteggi di tre sottoscale (Mathematics, Reading comprehension, Reading recognition);
- `mom_hs`: variabile dicotomica (0 or 1) che indica se la madre del bambino ha completato le scuole superiori (1) oppure no (0);
- `mom_iq`: QI della madre;
- `mom_age`: età della madre.

```
library("rio")
df <- rio::import(here::here("data", "kidiq.dta"))
head(df)
#>   kid_score mom_hs mom_iq mom_work mom_age
#> 1        65      1 121.12        4      27
#> 2        98      1  89.36        4      25
#> 3        85      1 115.44        4      27
#> 4        83      1  99.45        3      25
#> 5       115      1  92.75        4      27
#> 6        98      0 107.90        1      18
```

Calcoliamo alcune statistiche descrittive:

```
summary(df)
#>    kid_score         mom_hs          mom_iq
#>  Min.   : 20.0   Min.   :0.000   Min.   : 71.0
#>  1st Qu.: 74.0   1st Qu.:1.000   1st Qu.: 88.7
#>  Median : 90.0   Median :1.000   Median : 97.9
#>  Mean   : 86.8   Mean   :0.786   Mean   :100.0
#>  3rd Qu.:102.0   3rd Qu.:1.000   3rd Qu.:110.3
#>  Max.   :144.0   Max.   :1.000   Max.   :138.9
```

```
#>     mom_work      mom_age
#>  Min.   :1.0   Min.   :17.0
#>  1st Qu.:2.0   1st Qu.:21.0
#>  Median :3.0   Median :23.0
#>  Mean   :2.9   Mean   :22.8
#>  3rd Qu.:4.0   3rd Qu.:25.0
#>  Max.   :4.0   Max.   :29.0
```

Il QI medio dei bambini è di circa 87 mentre quello della madre è di 100. La gamma di età delle madri va da 17 a 29 anni con una media di circa 23 anni. Si noti infine che il 79% delle mamme ha un diploma di scuola superiore.

Ci poniamo ora il problema di descrivere l'associazione tra il QI dei figli, `kid_score`, e il QI delle madri, `mom_iq`, mediante un modello lineare. Le stime dei minimi quadrati sono fornite dalla funzione `lm()`:

```
fm <- lm(kid_score ~ mom_iq, data = df)
fm %>%
  tidy() %>%
  filter(term == c("(Intercept)", "mom_iq")) %>%
  pull(estimate)
#> [1] 25.80  0.61
```

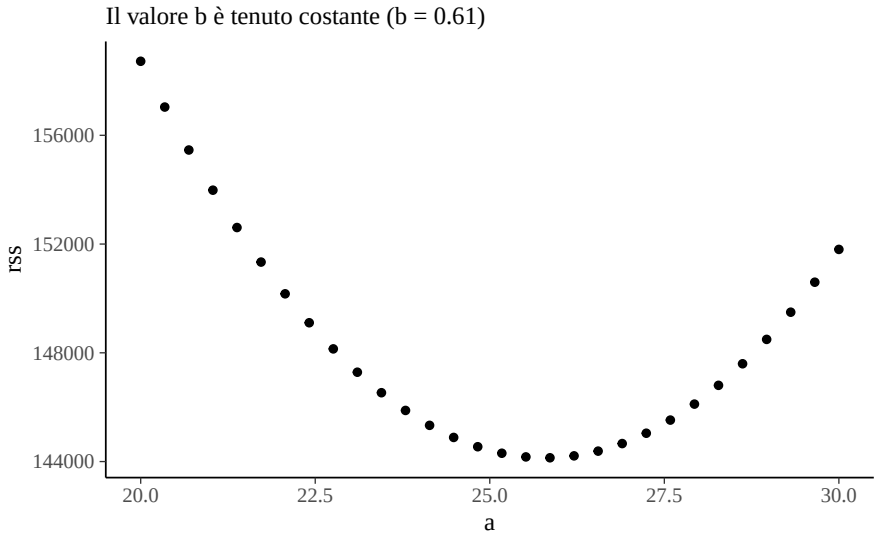
Calcoliamo la somma dei residui quadratici in base al modello di regressione $\hat{y}_i = 25.8 + 0.61 x_i$:

```
rss(df$mom_iq, df$kid_score, 25.8, 0.61)
#> [1] 144137
```

Per sviluppare una comprensione intuitiva del metodo dei minimi quadrati, esploriamo i valori assunti da `rss` per diversi valori di a e b. Per semplicità, manteniamo costante `b = 0.61` e variamo i valori `a`.

```
tibble(a = seq(20, 30, length.out = 30)) %>%
  mutate(
    rss = purrr::map_dbl(
      a,
```

```
      rss,
      x = df$mom_iq,
      y = df$kid_score,
      b = 0.61
    )
  ) %>%
  ggplot(aes(x = a, y = rss)) +
  geom_point() +
  labs(subtitle = "Il valore b è tenuto costante (b = 0.61)")
```

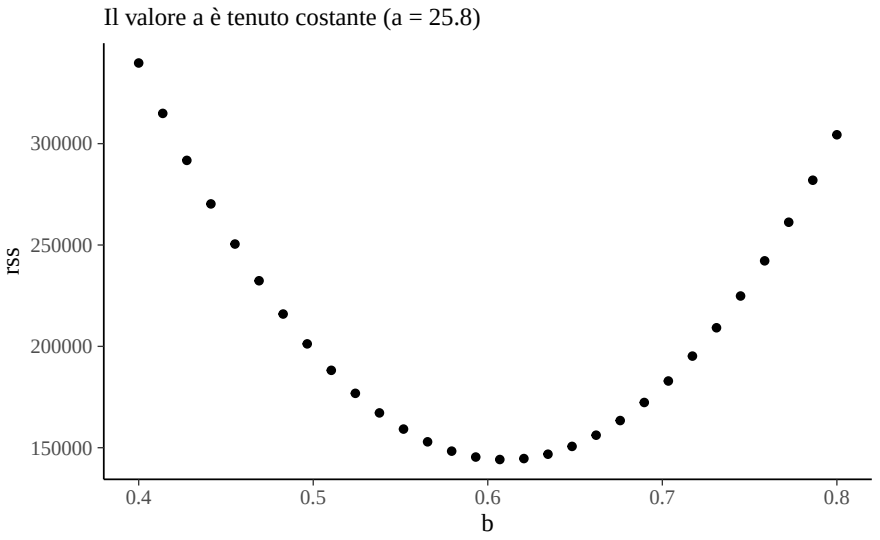


Il minimo della funzione che qui abbiamo discretizzato costituisce la stima dei minimi quadrati del parametro β_0.

Lo stesso può essere fatto per b:

```
tibble(b = seq(0.4, 0.8, length.out = 30)) %>%
  mutate(
    rss = purrr::map_dbl(
      b,
      rss,
      x = df$mom_iq,
      y = df$kid_score,
```

```
    a = 25.8
  )
) %>%
ggplot(aes(x = b, y = rss)) +
geom_point() +
labs(subtitle = "Il valore a è tenuto costante (a = 25.8)")
```



Il minimo della funzione rappresentata qui sopra costituisce la stima dei minimi quadrati del parametro β_1.

In generale, possiamo dire che il metodo dei minimi quadrati consente di minimizzare la funzione quadratica $RSS = \sum_{i=1}^{n} \left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\right)^2$ rispetto alle due incognite $\hat{\beta}_0$ e $\hat{\beta}_1$. Numericamente, ciò corrisponde a variare sia `a` che `b` simultaneamente in un listato simile a quello riportato sopra. Anche se il codice R necessario per ottenere questo risultato è più complesso di quello qui esaminato, l'idea di base non cambia.

Osservazione. Nelle precedenti istruzioni R abbiamo utilizzato la funzione `purrr::map_dbl()`. Questo oggetto R consente di applicare una funzione (in questo caso, `rss()`) a ciascun elemento di un vettore in input (nel caso presente, il vettore `a` oppure il vettore `b`). La funzione `purrr::map_dbl()` ritorna un numero reale.

Commenti e considerazioni finali

Se gli errori del modello lineare sono indipendenti e distribuiti normalmente, in modo che $y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$ per ogni i, allora la stima ai minimi quadrati di $(\hat{\beta}_0, \hat{\beta}_1)$ coincide con la stima di massima verosimiglianza di questi parametri. In un modello lineare, la funzione di verosimiglianza è definita come la densità di probabilità delle osservazioni, dati i parametri e i predittori, ovvero,

$$p(y \mid \beta_0, \beta_1, \sigma, x) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid \beta_0 + \beta_1 x_i, \sigma^2), \tag{4.4}$$

dove $\mathcal{N}(\cdot|\cdot, \cdot)$ è la funzione gaussiana.

Un studio della (4.4) mostra che la massimizzazione della verosimiglianza richiede la minimizzazione della somma dei quadrati dei residui. Se gli errori sono indipendenti e distribuiti normalmente, quindi, la stima dei minimi quadrati $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)$ coincide con la stima di massima verosimiglianza per i parametri del modello lineare.

Bibliografia

Gelman, A., Hill, J., and Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press.

Horn, S. and Loewenstein, G. (2021). Underestimating learning by doing. *Available at SSRN 3941441*.