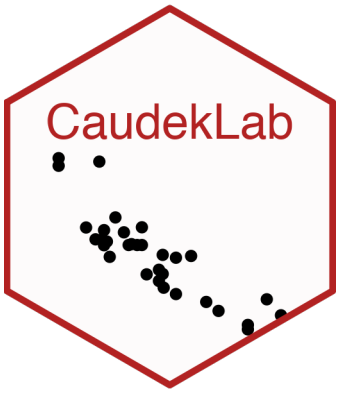
Psicometria

Corrado Caudek

2021-12-04

Questo documento è stato realizzato con:

- LaTeX e la classe memoir (http://www.ctan.org/pkg/memoir);
- R (http://www.r-project.org/) e RStudio (http://www.rstudio.com/);
- bookdown (http://bookdown.org/) e memoiR (https://ericmarcon.github.io/memoiR/).



Nel blog della mia pagina personale sono forniti alcuni approfondimenti degli argomenti qui trattati.

https://ccaudek.github.io/caudeklab/

Indice

Data della versione presente: Dicembre 04, 2021.

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.

L'insegnamento di 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 le 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

> It's worth noting, before getting started, that this material is hard. If you find yourself confused at any point, you are normal. Any sense of confusion you feel is just your brain correctly calibrating to the subject matter. Over time, confusion is replaced by comprehension [...] — Richard McElreath

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 ponte, 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 psico-

logico 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 verrano 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

> I know quite certainly that I myself have no special talent. Curiosity, obsession and dogged endurance, combined with self-criticism, have brought me to my ideas. — Albert Einstein

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

> Memorization is not learning. — Richard Phillips Feynman

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 (Burger & Starbird, 2012).

- 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 possiamo 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

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

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 un software, ricordatevi: "Google is your friend".

Corrado Caudek

Febbraio 2022

CAPITOLO 1

Introduzione alla regressione lineare

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 di regressione 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 di regressione 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 $b = \Delta y$.

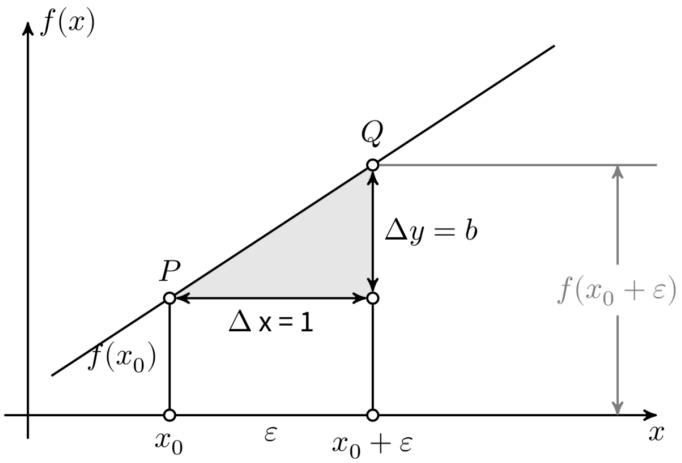


Figura 1.1: *La funzione lineare $y = a + bx$.*

1.2 L'errore di misurazione

Per descrivere l'associazione tra due variabili, tuttavia, la funzione lineare non è sufficiente. Nel mondo empirico, infatti, la relazione tra variabili non è mai perfettamente lineare. È dunque necessario includere nel modello di regressione anche una componente d'errore, ovvero una componente della y che non può essere spiegata dal modello lineare. Nel caso di due sole variabili, questo ci conduce alla seguente formulazione del modello di regressione:

$$y = \alpha + \beta x + \varepsilon, \tag{1.5}$$

laddove i parametri α e β descrivono l'associazione tra le variabili casuali y e x, e il termine d'errore ε specifica quant'è grande la porzione della variabile y che non può essere predetta nei termini di una relazione lineare con la x.

Si noti che la (1.5) consente di formulare una predizione, nei termini di un modello lineare, del valore atteso della y conoscendo x, ovvero

$$\hat{y} = \mathbb{E}(y \mid x) = \alpha + \beta x. \tag{1.6}$$

In altri termini, se i parametri del modello (α e β) sono noti, allora è possibile predire la y sulla base della nostra conoscenza della x. Per esempio, se conosciamo la relazione lineare tra quoziente di intelligenza ed aspettativa di vita, allora possiamo prevedere quanto a lungo vivrà una persona sulla base del suo QI. Sì, c'è una relazione lineare tra intelligenza e aspettativa di vita (Hambrick, 2015)! Ma quando è accurata la previsione? Ciò dipende dal termine d'errore della (1.5). L'analisi di regressione fornisce un metodo per rispondere a domande di questo tipo[1].

1.3 Una media per ciascuna osservazione

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

$$Y_i \overset{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma), \quad i = 1, \dots, n. \tag{1.7}$$

Il modello (1.7) assume che ogni Y_i sia una realizzazione della stessa $\mathcal{N}(\mu, \sigma^2)$. Da un punto di vista bayesiano[2], 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 però 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 Normale della (1.7) per lo studio della possibile relazione tra y_i e x_i?

Il modello (1.7) assume una media μ comune per ciascuna osservazione Y_i. Dal momento che desideriamo introdurre una nuova variabile x_i che assume un valore specifico per ciascuna osservazione y_i, il modello (1.7) può essere modificato in modo che la media comune μ venga sostituita da una media μ_i specifica a ciascuna i-esima osservazione:

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

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 (1.8)

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

$$\mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n. \tag{1.9}$$

[1] Per una discussione sugli aspetti di base della regressione lineare, si veda il capitolo 7 di *Introduction to Modern Statistics*.

[2] Per un'introduzione alla trattazione frequentista dell'analisi di regressione, si veda l'Appendice **??**.

Nella (1.9), 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 che rappresentano l'intercetta e la pendenza della retta di regressione sono variabili casuali. Si assegna una distribuzione a priori a β_0 e a β_1 e si esegue l'inferenza riassumendo la distribuzione a posteriori di questi parametri.

In questo modello, 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 (1.8) 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.

Il modello di regressione lineare

Sostituendo la (1.9) nel modello (1.8) otteniamo il modello di regressione lineare:

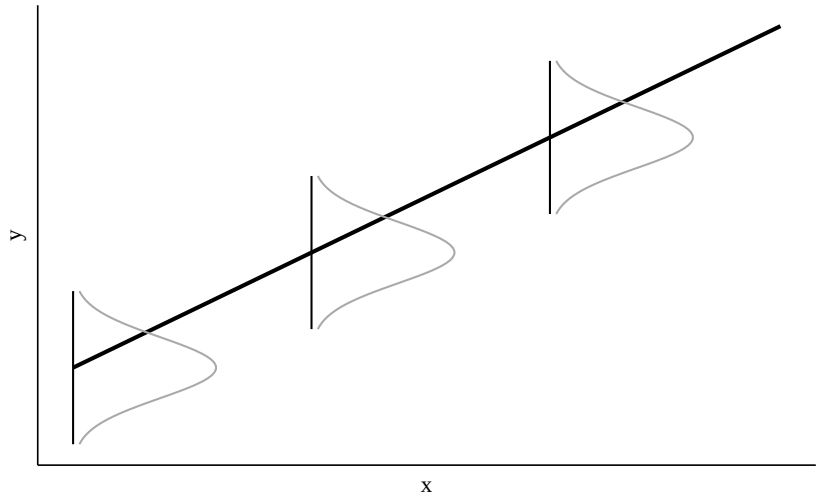
$$Y_i \mid \beta_0, \beta_1, \sigma \overset{ind}{\sim} \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma), \quad i = 1, \dots, n. \tag{1.10}$$

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*.

In maniera equivalente, il modello (1.10) può essere formulato come

$$Y_i = \mu_i + \varepsilon_i, \quad i = 1, \dots, n, \tag{1.11}$$

dove la risposta media è $\mu_i = \beta_0+\beta_1 x_i$ e i residui $\varepsilon_1, \dots, \varepsilon_n$ sono i.i.d. da una Normale con media 0 e deviazione standard σ.



Nel modello di regressione lineare, l'osservazione Y_i è una variabile casuale, il predittore x_i è una costante fissa, e β_0, β_1 e σ sono parametri incogniti. Utilizzando il paradigma bayesiano, viene assegnata una distribuzione a priori congiunta a $(\beta_0, \beta_1, \sigma)$. Dopo avere osservato le risposte $Y_i, i = 1, \dots, n$, l'inferenza procede stimando la distribuzione a posteriori dei parametri.

Nella costruzione di un modello di regressione bayesiano, è importante iniziare dalle basi e procedere un passo alla volta. Sia Y una variabile di risposta e sia x un predittore

o un insieme di predittori. È possibile costruire un modello di regressione di Y su x applicando i seguenti principi generali:

- Stabilire se Y è discreto o continuo. Di conseguenza, identificare l'appropriata struttura dei dati (per esempio, Normale, di Poisson, o Binomiale).
- Esprimere la media di Y come funzione dei predittori x (per esempio, $\mu = \beta_0 + \beta_1 x$).
- Identificare tutti i parametri incogniti del modello (per esempio, μ, β_1, β_2).
- Valutare quali valori che ciascuno di questi parametri potrebbe assumere. Di conseguenza, identificare le distribuzioni a priori appropriate per questi parametri.

Nel caso di una variabile Y continua che segue la legge Normale e un solo predittore, ad esempio, il modello diventa:

$$\begin{aligned} Y_i \mid \beta_0, \beta_1, \sigma &\overset{ind}{\sim} \mathcal{N}(\mu_i, \sigma^2) \quad \text{con} \quad \mu_i = \beta_0 + \beta_1 x_i \\ \beta_0 &\sim \mathcal{N}(\mu_0, \sigma_0^2) \\ \beta_1 &\sim \mathcal{N}(\mu_1, \sigma_1^2) \\ \sigma &\sim \text{Cauchy}(x_0, \gamma) \, . \end{aligned}$$

Un algoritmo MCMC viene usato per simulare i campioni dalle distribuzioni a posteriori e, mediante tali campioni, si fanno inferenze sulla risposta attesa $\beta_0 + \beta_1 x$ per ciascuno specifico valore del predittore x. Inoltre, è possibile valutare le dimensioni degli errori di previsione mediante un indice sintetico della densità a posteriori della deviazione standard σ.

Considerazioni conclusive

Il modello di regressione 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 di regressione ci consente di prevedere il valore della variabile dipendente in base ad alcuni nuovi valori della variabile indipendente. Il modello di regressione lineare semplice è in realtà molto limitato, in quanto descrive soltanto la relazione tra la variabile dipendente y e una sola variabile esplicativa x. Esso diventa molto più utile quando incorpora più variabili indipendenti. In questo secondo caso, però, i calcoli per la stima dei coefficienti del modello diventano più complicati. Abbiamo deciso di iniziare considerando il modello di regressione lineare semplice perché, in questo caso, sia la logica dell'inferenza sia le procedure di calcolo sono facilmente maneggiabili. Nel caso più generale, quello del modello di regressione multipla, la logica dell'inferenza rimarrà identica a quella discussa qui, ma le procedure di calcolo richiedono l'uso dell'algebra matriciale. Il modello di regressione multipla può includere sia regressori quantitativi, sia regressori qualitativi, utilizzando un opportuna schema di codifica. È interessante notare come un modello di regressione multipla che include una sola variabile esplicativa quantitativa corrisponde all'analisi della varianza ad una via; un modello di regressione multipla che include più di una variabile esplicativa quantitativa corrisponde all'analisi della varianza più vie. Possiamo qui concludere dicendo che il modello di regressione, nelle sue varie forme e varianti, costituisce la tecnica di analisi dei dati maggiormente usata in psicologia.

CAPITOLO 2

Adattare il modello di regressione ai dati

In breve

In questo Capitolo verranno esposte alcune nozioni matematiche che stanno alla base dell'inferenza per i modelli di regressione e un po' di algebra che ci aiuterà a comprendere la stima della regressione lineare. Spiegheremo anche la logica per l'uso della funzione bayesiana `brm()` e la sua connessione con la regressione lineare classica.

2.1 Minimi quadrati

Nel modello di regressione lineare classico, $y_i = a + bx_i + \varepsilon_i$, i coefficienti a e b 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 $y_i = a + bx_i$, senza errori, per tutti i punti $i = 1, \dots, n$). L'obiettivo della stima dei minimi quadrati è quello di scegliere i valori $(\hat{a}, \hat{b})$ che minimizzano la somma dei quadrati dei residui,

$$e_i = y_i - (\hat{a} + \hat{b}x_i)\,. \tag{2.1}$$

Distinguiamo tra i residui $e_i = y_i - (\hat{a} + \hat{b}x_i)$ e gli *errori* $\varepsilon_i = y_i - (a + bx_i)$. 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 a e b.

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

$$\text{RSS} = \sum_{i=1}^{n} (y_i = (\hat{a} + \hat{b}x_i))^2\,. \tag{2.2}$$

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

Stima della deviazione standard dei residui σ

Nel modello di regressione, 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, a), 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, $\sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i - (\hat{a} + \hat{b}x_i))^2}$, 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 della regressione 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

$$\hat{\sigma} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - (\hat{a} + \hat{b}x_i))^2}\,. \tag{2.3}$$

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.

2.2 Calcolare la somma dei quadrati

Seguendo Solomon Kurz, creiamo una funzione per calcolare la somma dei quadrati per diversi valori di a e b:

```
rss <- function(x, y, a, b) {
  # x and y are vectors,
  # a and b are scalars
  resid <- y - (a + b * x)
  return(sum(resid^2))
  }
```

Useremo i dati ….

```
df <- read.dta(here("data", "kidiq.dta"))
head(df)
#>   kid_score mom_hs mom_iq mom_work mom_age
#> 1        65      1  121.1        4      27
#> 2        98      1   89.4        4      25
#> 3        85      1  115.4        4      27
#> 4        83      1   99.4        3      25
#> 5       115      1   92.7        4      27
#> 6        98      0  107.9        1      18
```

Nell'esempio, `kid_score` è la variabile y e `mom_iq` è il predittore. Le stime dei minimi quadrati sono fornite dalla funzione `lm()`:

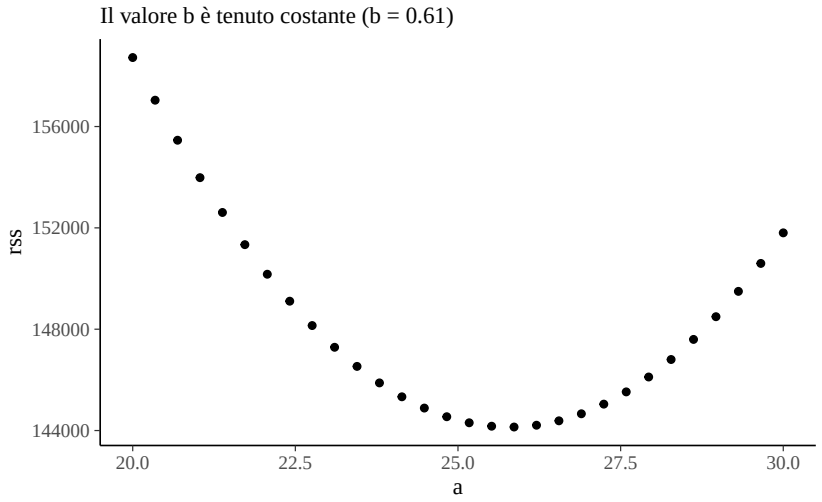
```
fm <- lm(kid_score ~ mom_iq, data = df)
fm %>%
  tidy() %>%
  filter(term == c("(Intercept)", "mom_iq")) %>%
  pull(estimate)
#> [1] 25.80  0.61
```

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

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

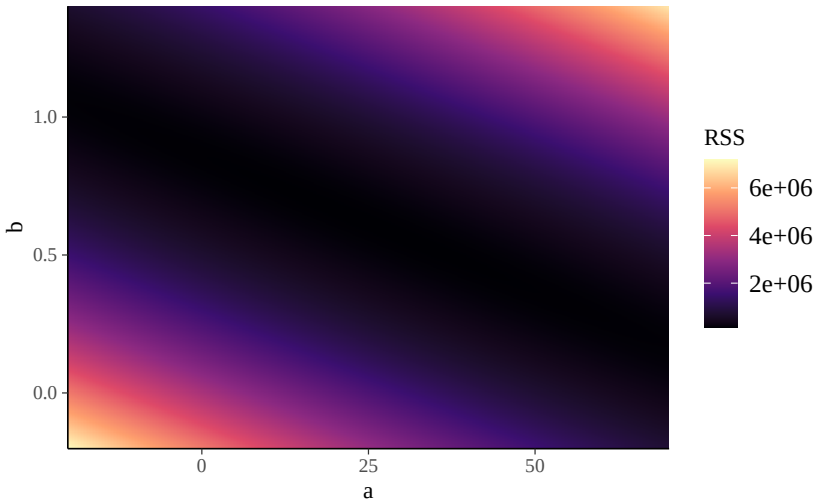
Esploriamo ora i valori assunti da *rss* per diversi valori di a e b. Per iniziare, utilizziamo un vettore di valori `a`, mantenendo costante `b = 0.61`.

```
# set the global plotting theme
# theme_set(theme_linedraw() +
#             theme(panel.grid = element_blank()))
# simulate
tibble(a = seq(20, 30, length.out = 30)) %>%
  mutate(
    rss = map_dbl(
      a,
      rss,
      x = df$mom_iq,
      y = df$kid_score,
      b = 0.61)
    ) %>%

  # plot
  ggplot(aes(x = a, y = rss)) +
  geom_point() +
  labs(subtitle = "Il valore b è tenuto costante (b = 0.61)")
```



Ora variamo sia `a` che `b`, facendo assumere a ciascun parametro un insieme di valori in un intevallo, e rappresentiamo i risultati in una heat map che rappresenta l'intensitè di `rss` in funzione dei valori `a` e `b`.

```
d <-
  crossing(a = seq(-20, 70, length.out = 400),
           b = seq(-0.2, 1.4, length.out = 400)) %>%
  mutate(rss = map2_dbl(a, b, rss, x = df$mom_iq, y = df$kid_score))
d %>%
  ggplot(aes(x = a, y = b, fill = rss)) +
  geom_tile() +
  scale_fill_viridis_c("RSS", option = "A") +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0))
```



Poiché la stima dei minimi quadrati enfatizza il valore RSS minimo, la soluzione che cerchiamo corrisponde alle combinazione di `a` e `b` nell'intervallo più scuro rappresentato nella figura. Tra gli `a` e `b` che abbiamo preso in considerazione, la coppia di valori a cui è associato il minimo valore `rss` si trova nel modo seguente:

```
d %>%
  arrange(rss) %>%
  slice(1)
#> # A tibble: 1 x 3
#>       a     b     rss
#>   <dbl> <dbl>   <dbl>
#> 1  25.8 0.610 144137.
```

Si noti che i valori trovati in questo modo corrispondono alla soluzione fornita nell'output della funzione `lm()`.

2.3 Massima verosimiglianza

Se gli errori del modello lineare sono indipendenti e distribuiti normalmente, in modo che $y_i \sim \mathcal{N}(a + bx_i, \sigma^2)$ per ogni i, allora la stima ai minimi quadrati di (`a`, `b`) è anche la stima di massima verosimiglianza. La *funzione di verosimiglianza* in un modello di regressione è definita come la densità di probabilità delle osservazioni dati i parametri e i predittori; quindi, in questo esempio,

$$p(y \mid a, b, \sigma, x) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid a + bx_i, \sigma^2),$$

dove $\mathcal{N}(\cdot|\cdot, \cdot)$ è la funzione di densità di probabilità normale,

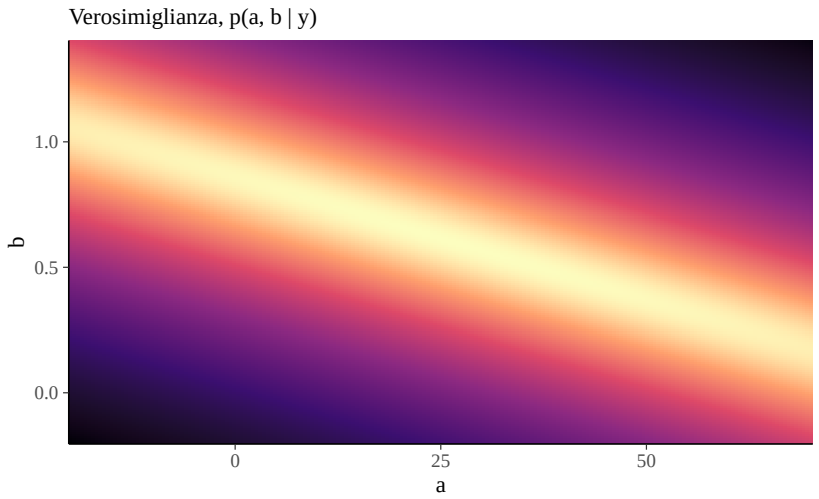
$$\mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right).$$

Un studio della funzione precedente rivela che la massimizzazione della verosimiglianza richiede la minimizzazione della somma dei quadrati dei residui; quindi la stima dei minimi quadrati $\hat{\beta} = (\hat{a}, \hat{b})$ può essere vista come una stima di massima verosimiglianza nel modello normale:

```
ll <- function(x, y, a, b) {
  resid <- y - (a + b * x)
  sigma <- sqrt(sum(resid^2) / length(x))
  d <- dnorm(y, mean = a + b * x, sd = sigma, log = TRUE)
  tibble(sigma = sigma, ll = sum(d))
}
```

Calcoliamo dunque le stime di verosimiglianza logaritmica per varie combinazioni di (`a`, `b`), date due colonne di dati, `x` e `y`. La funzione resituisce anche il valore $\hat{\sigma}$.

```
d <-
  crossing(a = seq(-20, 70, length.out = 200),
           b = seq(-0.2, 1.4, length.out = 200)) %>%
  mutate(ll = map2(a, b, ll, x = df$mom_iq, y = df$kid_score)) %>%
  unnest(ll)
p1 <-
  d %>%
  ggplot(aes(x = a, y = b, fill = ll)) +
  geom_tile() +
  scale_fill_viridis_c(option = "A", breaks = NULL) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  labs(subtitle = "Verosimiglianza, p(a, b | y)")
p1
```



Le stime di $\hat{a}, \hat{b}$ ottenute mediante il metodo di massima verosimiglianza sono:

```
d %>%
  arrange(desc(ll)) %>%
  slice(1)
#> # A tibble: 1 x 4
#>       a     b sigma     ll
#>   <dbl> <dbl> <dbl>  <dbl>
#> 1  25.7 0.612  18.2 -1876.
```

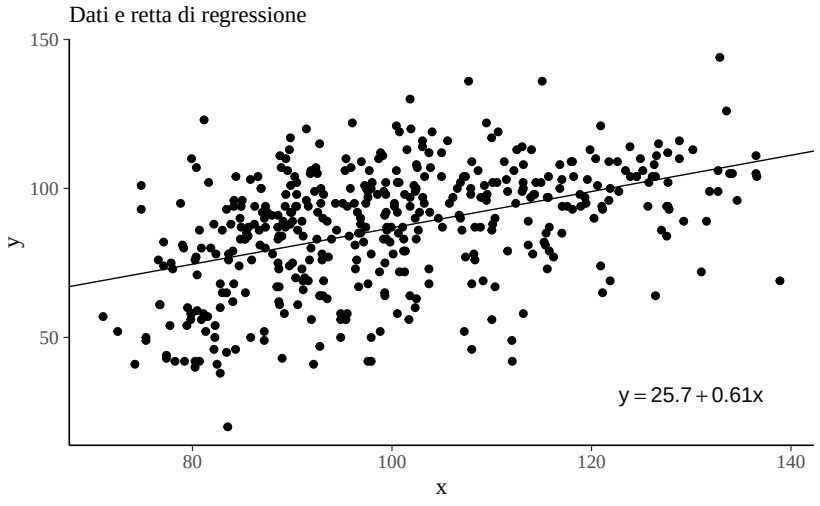
Inferenza bayesiana

Usiamo ora la funzione `brms::brm()` per eseguire l'analisi mediante un approccio bayesiano:

```
m <-
  brm(
    kid_score ~ mom_iq,
    data = df,
    backend = "cmdstan",
    refresh = 0
  )
#> Running MCMC with 4 chains, at most 8 in parallel...
#>
#> Chain 1 finished in 0.0 seconds.
#> Chain 2 finished in 0.0 seconds.
#> Chain 3 finished in 0.0 seconds.
#> Chain 4 finished in 0.0 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 0.0 seconds.
#> Total execution time: 0.3 seconds.
```

Utilizzando i coefficienti calcolati da `brms::brm()`, aggiungiamo la stima della retta di regressione al diagramma di dispersione dei dati:

```
df %>%
  ggplot(aes(x = mom_iq, y = kid_score)) +
  geom_point() +
  geom_abline(
    intercept = fixef(m, robust = TRUE)[1, 1],
    slope = fixef(m, robust = TRUE)[2, 1],
    size = 1/3
  ) +
  annotate(
    geom = "text",
    x = 130, y = 30,
    label = expression(y == 25.7 + 0.61 * x)
  ) +
  labs(
    subtitle = "Dati e retta di regressione",
    x = "x",
    y = "y"
  )
```

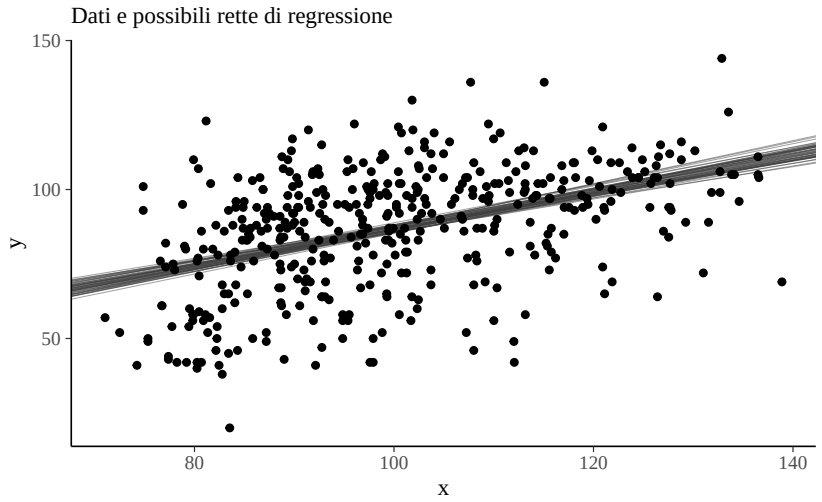


La funzione `brms::posterior_samples()` consente di estrarre molti campioni dalla distribuzione a posteriori del modello `m`. In questo modo otteniamo un vettore di valori per ciascuno dei tre parametri, i quali, in questo output sono chiamati `b_Intercept`, `b_mom_iq` e `sigma`. Abbiamo quindi usato `slice_sample()` per ottenere un sottoinsieme casuale di 50 righe. Per semplicità, qui ne stampiamo solo 5.

```
set.seed(8)

posterior_samples(m) %>%
  slice_sample(n = 5)
#>   b_Intercept b_mom_iq sigma Intercept  lp__
#> 1        18.7    0.671  19.2      85.8 -1883
#> 2        20.4    0.663  17.9      86.7 -1881
#> 3        33.3    0.546  18.5      87.8 -1882
#> 4        25.5    0.620  18.3      87.5 -1881
#> 5        29.0    0.577  18.2      86.7 -1881
```

Possiamo interpretare i valori `b_Intercept`, `b_mom_iq` come un insieme di valori credibili per i parametri a e b. Dati questi valori credibili per i parametri del modello di regressione, possiamo aggiungere al diagramma a dispersione 50 stime possibili della retta di regressione, alla luce dei dati osservati.

```
set.seed(8)

posterior_samples(m) %>%
  slice_sample(n = 50) %>%

  ggplot() +
  geom_abline(
    aes(intercept = b_Intercept, slope = b_mom_iq),
        size = 1/4, alpha = 1/2, color = "grey25") +
  geom_point(
    data = df,
    aes(x = mom_iq, y = kid_score)
  ) +
  labs(
    subtitle = "Dati e possibili rette di regressione",
    x = "x",
    y = "y"
  )
```



I minimi quadrati o la massima verosimiglianza trovano i parametri che meglio si adattano ai dati (secondo un criterio prestabilito), ma senza altrimenti vincolare o guidare l'adattamento. Ma di solito abbiamo informazioni preliminari sui parametri del modello. L'inferenza bayesiana produce un compromesso tra informazioni a priori e i dati, moltiplicando la verosimiglianza con una distribuzione a priori che codifica probabilisticamente le informazioni esterne sui parametri. Il prodotto della verosimiglianza $p(y \mid a, b, \sigma)$ e della distribuzione a priori è chiamato *distribuzione a posteriori* e, dopo aver visto i dati, riassume la nostra credenza sui valori dei parametri.

La soluzione dei minimi quadrati fornisce una stima puntuale dei coefficienti che producono il miglior adattamento complessivo ai dati. Per un modello bayesiano, la corrispondente stima puntuale è la moda a posteriori, la quale fornisce il miglior adattamento complessivo ai dati e alla distribuzione a priori. La stima dei minimi quadrati o di massima verosimiglianza è la moda a posteriori corrispondente al modello bayesiano che utilizza una distribuzione a priori uniforme.

Ma non vogliamo solo una stima puntuale; vogliamo anche una misura dell'incertezza della stima. La figura precedente fornisce, in forma grafica, una descrizione di tale incertezza.

Gli intervalli di credibilità al 95% si ottengono nel modo seguente:

```
print(m, robust = TRUE)
#>  Family: gaussian
#>   Links: mu = identity; sigma = identity
#> Formula: kid_score ~ mom_iq
#>    Data: df (Number of observations: 434)
#>   Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
#>          total post-warmup draws = 4000
#>
#> Population-Level Effects:
#>           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
#> Intercept    25.83      5.80    14.44    37.47 1.00     3329     3005
#> mom_iq        0.61      0.06     0.49     0.72 1.00     3310     3009
#>
#> Family Specific Parameters:
#>       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
#> sigma    18.26      0.61    17.13    19.53 1.00     3824     3105
#>
#> Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
```

```
#> and Tail_ESS are effective sample size measures, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

In alternativa, è possibile usare la funzione `posterior_interval()`:

```
posterior_interval(m)
#>                   2.5%     97.5%
#> b_Intercept    14.445    37.471
#> b_mom_iq        0.495     0.722
#> sigma          17.132    19.529
#> Intercept      85.119    88.563
#> lp__        -1885.190 -1880.580
```

CAPITOLO 3

Regressione lineare in Stan

3.1 La specificazione del modello in linguaggio Stan

Obiettivo di questo Capitolo è illustrare come svolgere l'analisi di regressione bayesiana usando il linguaggio Stan.[1] 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 compres 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.

Leggiamo i dati in R:

```
library("foreign")
df <- read.dta(here("data", "kidiq.dta"))
head(df)
#>   kid_score mom_hs mom_iq mom_work mom_age
#> 1        65      1  121.1        4      27
#> 2        98      1   89.4        4      25
#> 3        85      1  115.4        4      27
#> 4        83      1   99.4        3      25
#> 5       115      1   92.7        4      27
#> 6        98      0  107.9        1      18
```

Calcoliamo alcune statistiche descrittive:

```
summary(df)
#>    kid_score          mom_hs          mom_iq         mom_work      mom_age
```

[1] Una descrizione dell'approccio frequentista all'analisi di regressione è fornita nell'Appendice **??**.

```
#>  Min.   : 20.0   Min.   :0.000   Min.   : 71.0   Min.   :1.0   Min.   :17.0
#>  1st Qu.: 74.0   1st Qu.:1.000   1st Qu.: 88.7   1st Qu.:2.0   1st Qu.:21.0
#>  Median : 90.0   Median :1.000   Median : 97.9   Median :3.0   Median :23.0
#>  Mean   : 86.8   Mean   :0.786   Mean   :100.0   Mean   :2.9   Mean   :22.8
#>  3rd Qu.:102.0   3rd Qu.:1.000   3rd Qu.:110.3   3rd Qu.:4.0   3rd Qu.:25.0
#>  Max.   :144.0   Max.   :1.000   Max.   :138.9   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 il problema di descrivere l'associazione tra il QI dei figli e il QI delle madri mediante un modello di regressione lineare. Per farci un'idea del valore dei parametri, adattiamo il modello di regressione ai dati mediante la procedura di massima verosimiglianza:

```
coef(lm(kid_score ~ mom_iq, data = df))
#> (Intercept)      mom_iq
#>       25.80        0.61
```

Il modello statistico bayesiano di regressione lineare è:

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

La prima riga definisce la funzione di verosimiglianza e righe successive definiscono le distribuzioni a priori dei parametri. Il segno $\sim$ (tilde) si può leggere "si distribuisce come". La prima riga, dunque, ci dice che ciascuna osservazione y_i è una variabile casuale che segue la distribuzione Normale 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 α e β. Le due righe successive specificano le distribuzioni a priori per α e β; per α, la distribuzione a priori è una distribuzione Normale di parametri $\mu_\alpha = 25$ e deviazione standard $\sigma_\alpha = 10$; per β, la distribuzione a priori è una distribuzione Normale standardizzata. L'ultima riga definisce la la distribuzione a priori di σ, ovvero una Cauchy di parametri 18 e 5.

Poniamoci ora il problema di specificare il modello bayesiano descritto sopra in linguaggio Stan[2]. 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 di regressione può essere scritto come

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

dove

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

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

[3]Si noti un punto importante. Il fatto di standardizzare i dati fa in modo che le distribuzioni a priori sui parametri andranno 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 sicuramente distribuzioni vagamente informative il cui unico scopo è quello della regolarizzazione dei dati, ovvero l'obiettivo di mantenere le inferenze in una gamma ragionevole di valori; ciò contribuisce nel contempo a limitare l'influenza eccessiva delle osservazioni estreme (valori anomali) — certamente tali distribuzioni a priori non introducono alcuna distorsione sistematica nella stima a posteriori.

I dati devono essere prima centrati sottraendo la media campionaria, quindi scalati dividendo per la deviazione standard campionaria. Quindi un'osservazione u viene standardizzata dalla funzione z definita da

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

dove la media $\bar{y}$ è

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

e la deviazione standard è

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

La trasformata inversa è definita invertendo i due passaggi precedenti, ovvero usando la deviazione standard per scalare di nuovo i valori u per poi traslarli con la media campionaria:

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

Consideriamo il seguente modello iniziale nel 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`.

Qui sotto è invece riportato il modello per i 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 per 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;
}
model {
  alpha_std ~ normal(0, 2);
  beta_std ~ normal(0, 2);
  sigma_std ~ cauchy(0, 2);
  y_std ~ normal(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/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. Le distribuzioni a priori per i parametri sono vagamente informative.

I parametri originali possono essere recuperati con un po' di algebra.

$$
\begin{aligned}
y_n &= \mathrm{z}_y^{-1}(\mathrm{z}_y(y_n)) \\
&= \mathrm{z}_y^{-1}\left(\alpha' + \beta' \mathrm{z}_x(x_n) + \epsilon'_n\right) \\
&= \mathrm{z}_y^{-1}\left(\alpha' + \beta'\left(\frac{x_n - \bar{x}}{\mathtt{sd}(x)}\right) + \epsilon'_n\right) \\
&= \mathtt{sd}(y)\left(\alpha' + \beta'\left(\frac{x_n - \bar{x}}{\mathtt{sd}(x)}\right) + \epsilon'_n\right) + \bar{y} \\
&= \left(\mathtt{sd}(y)\left(\alpha' - \beta'\frac{\bar{x}}{\mathtt{sd}(x)}\right) + \bar{y}\right) + \left(\beta'\frac{\mathtt{sd}(y)}{\mathtt{sd}(x)}\right) x_n + \mathtt{sd}(y)\epsilon'_n,
\end{aligned} \tag{3.1}
$$

da cui

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

I valori dei parametri sulle scale originali vengono calcolati nel blocco `generated quantities`.

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")
```

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)
```

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

```
mod$print()
#>
#> 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;
#> }
#> model {
#>   alpha_std ~ normal(0, 2);
#>   beta_std ~ normal(0, 2);
#>   sigma_std ~ cauchy(0, 2);
#>   y_std ~ normal(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;
#> }
```

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

```
mod$exe_file()
#> [1] "/Users/connado/_repositories/dspp/code/simpleregstd"
```

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.

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  rhat ess_bulk ess_tail
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl>
#> 1 alpha    25.8   25.9   5.98   5.97   15.8   35.7    1.00   16460.   11859.
#> 2 beta      0.610  0.609 0.0594 0.0590  0.513  0.709  1.00   16455.   11999.
#> 3 sigma    18.3   18.3   0.616  0.611  17.3   19.3    1.00   16786.   12022.
```

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()
#> Processing csv files: /var/folders/hl/dt523djx7_q7xjnthzjpdvc40000gn/T/RtmpyS0NAw/simpleregstd-202112
#>
#> Checking sampler transitions treedepth.
#> Treedepth satisfactory for all transitions.
#>
#> Checking sampler transitions for divergences.
#> No divergent transitions found.
#>
#> Checking E-BFMI - sampler transitions HMC potential energy.
#> E-BFMI satisfactory.
```

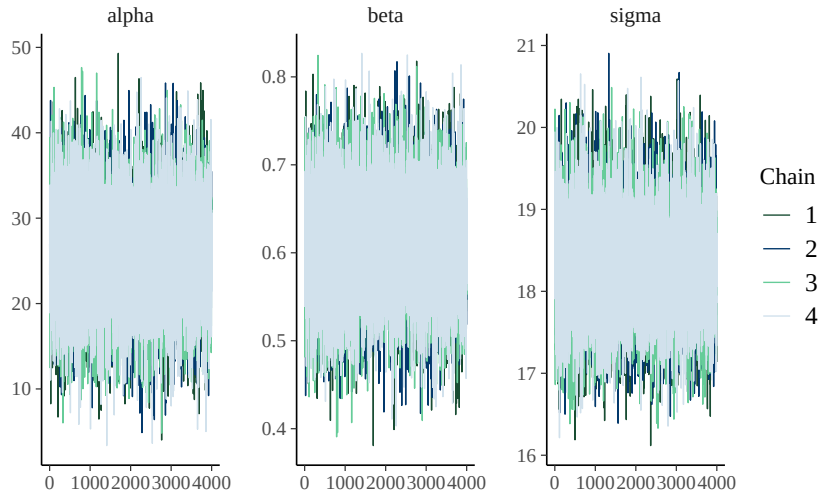
```
#>
#> Effective sample size satisfactory.
#>
#> Split R-hat values satisfactory all parameters.
#>
#> Processing complete, no problems detected.
```

È anche possibile creare un oggetto di classe `stanfit`

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

per poi utilizzare le funzioni del pacchetto `bayesplot`:

```
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`.

3.2 Interpretazione dei parametri

Assegnamo ai parametri la seguente interpretazione.

- L'intercetta pari a 25.8 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
#> [1] 41.4
```

- Il parametro σ fornisce una stima della dispersione delle osservazioni attorno al valore predetto dal modello di regressione, ovvero fornisce una stima della deviazione standard dei residui attorno alla retta di regressione.

Centrare i predittori

Per migliorare l'interpretazione dell'intercetta possiamo "centrare" la x, ovvero esprimere la x nei termini di scarti dalla media: $x - \bar{x}$. In tali circostanze, la pendenza della retta di regressione resterà immutata, ma l'intercetta corrisponderà 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  rhat ess_bulk ess_tail
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl>
#> 1 alpha    86.8   86.8   0.872  0.863  85.4   88.2    1.00   16613.   12276.
#> 2 beta      0.610  0.609 0.0591 0.0592  0.512  0.708  1.00   17947.   12419.
#> 3 sigma    18.3   18.3   0.616  0.616  17.3   19.3    1.00   16622.   11073.
```

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

CAPITOLO 4

Inferenza sul modello di regressione

I minimi quadrati o la massima verosimiglianza trovano i parametri che meglio si adattano ai dati (secondo un criterio prestabilito), ma senza altrimenti vincolare la stima. Ma di solito il ricercatore dispone di informazioni preliminari sui parametri del modello. L'inferenza bayesiana produce invece un compromesso tra tali informazioni pregresse e i dati.

La soluzione dei minimi quadrati è una stima puntuale che rappresenta il vettore dei coefficienti che fornisce il miglior adattamento complessivo ai dati. Per un modello bayesiano, la stima puntuale corrispondente è la *moda a posteriori*, che fornisce il miglior adattamento complessivo ai dati e alla distribuzione a priori. Si noti inoltre che la stima dei minimi quadrati (o di massima verosimiglianza) corrisponde alla moda a posteriori di un modello bayesiano con una distribuzione a priori uniforme.

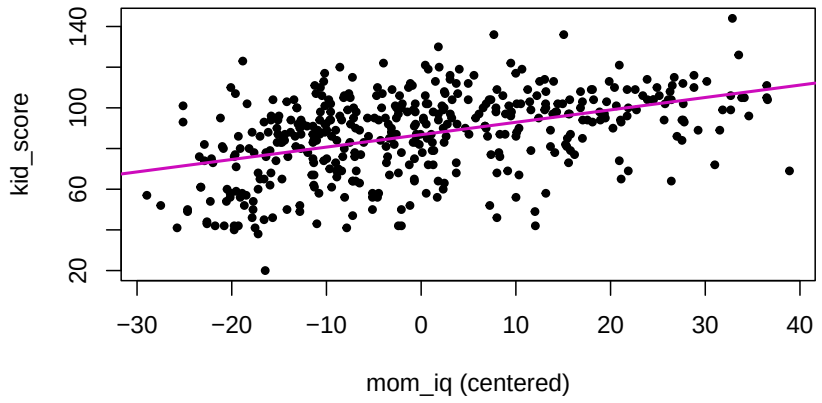
4.1 Rappresentazione grafica dell'incertezza della stima

Un primo modo per rappresentare l'incertezza dell'inferenza in un ottica bayesiana è quella di rappresentare graficamente la retta di regressione. Continuando con l'esempio descritto nel Capitolo precedente (ovvero, i dati `kid_score` e `mom_iq` centrati), usando la funzione `extract()`, salvo le stime a posteriori dei parametri in formato `list`:

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

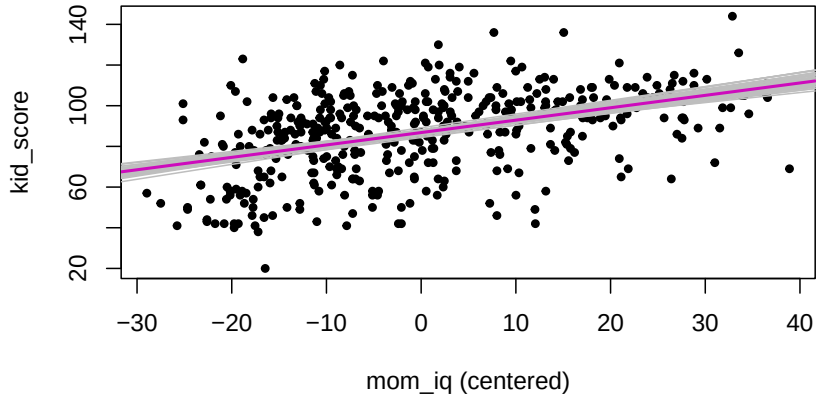
Un diagramma a dispersione dei dati con sovrapposto il valore atteso della y in base al modello bayesiano si ottiene nel modo seguente:

```
plot(
  df$kid_score ~ I(df$mom_iq - mean(df$mom_iq)),
  pch = 20,
  xlab = "mom_iq (centered)",
  ylab = "kid_score"
)
abline(mean(posterior$alpha), mean(posterior$beta), col = 6, lw = 2)
```



Un modo per visualizzare l'incertezza della stima della retta di regressione è quello di tracciare molteplici rette di regressione, ciascuna delle quali definita da una diversa stima dei parametri α e β che vengono estratti a caso dalle rispettive distribuzioni a posteriori.

```
plot(
  df$kid_score ~ I(df$mom_iq - mean(df$mom_iq)),
  pch = 20,
  xlab = "mom_iq (centered)",
  ylab = "kid_score"
)
for (i in 1:50) {
 abline(posterior$alpha[i], posterior$beta[i], col = "gray", lty = 1)
}
abline(mean(posterior$alpha), mean(posterior$beta), col = 6, lw = 2)
```



4.2 Intervalli di credibilità

L'incertezza inferenziale sui parametri può anche essere rappresentata mediante gli *intervalli di credibilità*, ovvero gli intervalli che contengono la quota desiderata (es., il 95%) della distribuzione a posteriori.

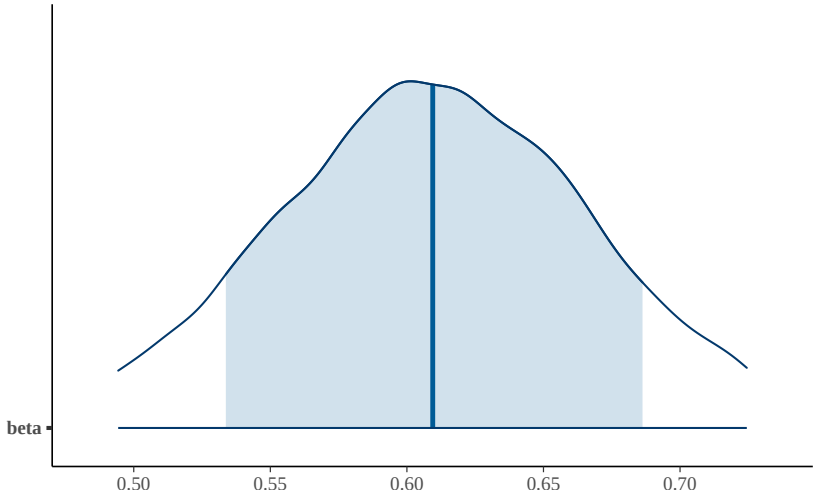
Per l'esempio che stiamo discutendo, gli intervalli di credibilità al 95% si ottengono nel modo seguente:

```
posterior <- extract(stanfit)
rstantools::posterior_interval(as.matrix(stanfit), prob = 0.95)
#>                2.5%     97.5%
#> alpha_std   -0.0838    0.0845
```

```
#> beta_std     0.3633    0.5324
#> sigma_std    0.8404    0.9580
#> alpha       85.0865   88.5229
#> beta         0.4943    0.7244
#> sigma       17.1538   19.5538
#> lp__      -172.8280 -168.2490
```

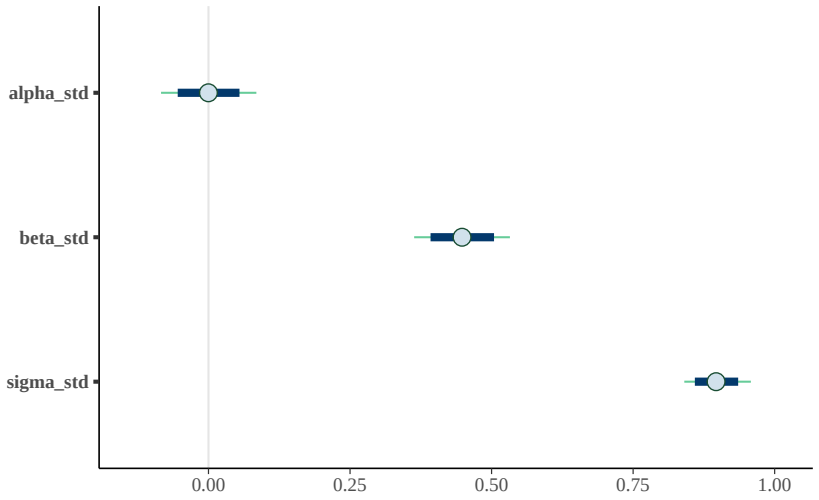
Un grafico che riporta l'intervallo di credibilità ai livelli di probabilità desiderati per β si ottiene con le seguenti istruzioni:

```
mcmc_areas(
  fit2$draws(c("beta")),
  prob = 0.8,
  prob_outer = 0.95
  )
```



Per i parametri ottenuti analizzando i dati standardizzati, abbiamo

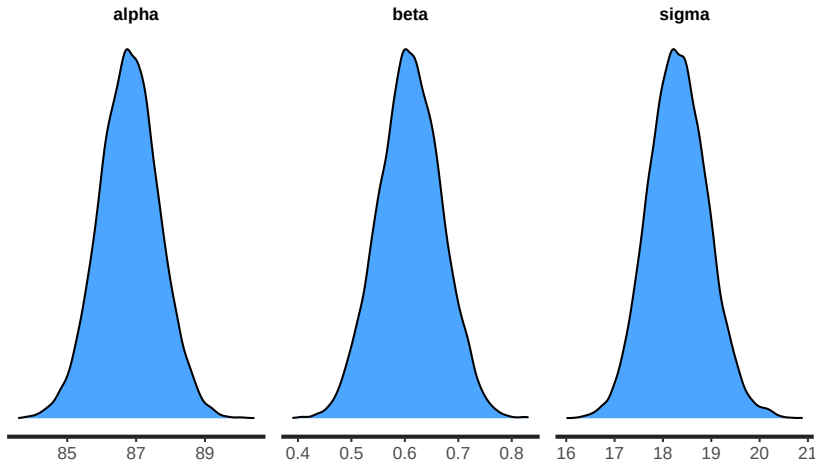
```
stanfit %>%
  mcmc_intervals(
    pars = c("alpha_std", "beta_std", "sigma_std"),
    prob = 0.8,
    prob_outer = 0.95
  )
```



4.3 Rappresentazione grafica della distribuzione a posteriori

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 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:

```
stan_dens(
  stanfit,
  pars = c("alpha", "beta", "sigma"),
  fill = "#4ca5ff"
)
```



4.4 Test di ipotesi

In Stan è facile valutare ipotesi direzionali. Per esempio, la probabilità di $\hat{\beta} > 0$ è

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

4.5 Regressione robusta

Spesso i ricercatori devono affrontare il problema degli outlier: in presenza di outlier, un modello statistico basato sulla distribuzione Normale produrrà delle stime dei parametri che non si generalizzano ad altri campioni di dati. Il metodo tradizoinale 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 eliminare gli outlier, i ricercatori finiscono per trovare risultati diversi.

Questo problema trova una semplice soluzione nell'approccio bayesiano. Nel modello di regressione che abbiamo dicusso finora è stato ipotizzato che $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$. Per un modello formulato in questi termini, la presenza di solo un valore anomalo e influente può avere un effetto drammatico sulle stime dei parametri.

Per fare un esempio, introduciamo un singlo valore anomalo 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 (le stime dei parametri sono simili a quelle di un modello bayesiano Normale con distribuzioni a priori vagamente informative). Sappiamo che, nel campione originari di dati, $\hat{\beta} \approx 0.6$. In presenza di un solo outlier troviamo che

```
coef(lm(kid_score ~ mom_iq, data = df2))
#> (Intercept)      mom_iq
#>      49.188       0.363
```

la stima di β viene drammaticamente ridotta (di quasi la metà!).

Non è però necessario assumere $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$. È altrettanto valido un modello che ipotizza una diversa distribuzione di densità per i residui 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 Normale. 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 di regressione usiamo la t di Student quale distribuzione dei residui, la presenza di outlier avrà una minore influenza sulle stime dei parametri di quanto avvenga nel modello Normale.

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

[1] È equivalente scrivere

$$y_i = \mu_i + \varepsilon_i, \quad \text{dove } \mu_i = \alpha + \beta x_i, \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon),$$

oppure

$$y_i \sim \mathcal{N}(\mu_i, \sigma_\varepsilon).$$

```
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, 2);
  beta_std ~ normal(0, 2);
  sigma_std ~ cauchy(0, 2);
  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 di regressione robusta 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
)
```

Esaminando le stime dei parametri

```
fit4$summary(c("alpha", "beta", "sigma", "nu"))
#> # A tibble: 4 x 10
#>   variable   mean median     sd    mad     q5    q95  rhat ess_bulk ess_tail
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl>
#> 1 alpha    87.8   87.8   0.887  0.899  86.3   89.3    1.00   13776.   11987.
#> 2 beta      0.603  0.603 0.0585 0.0577  0.506  0.698  1.00   14185.   11067.
#> 3 sigma    15.9   15.9   0.806  0.812  14.6   17.3    1.00   12699.   11256.
#> 4 nu        5.58   5.44  1.15   1.11    3.94   7.65   1.00   12258.   12033.
```

notiamo che la stima di β è rimasta praticamente immutata. La regressione "robusta" non risente dunque della presenza degli outlier.

CAPITOLO 5

Confronto tra due gruppi indipendenti

Il problema del confronto tra due gruppi indipendenti può essere formulato nei termini di un modello di regressione nel quale la variabile x è dicotomica, ovvero assume solo due valori.

5.1 Regressione lineare con una variabile dicotomica

Se x è una variabile dicotomica con valori 0 e 1, allora per il modello di regressione $\mu_i = \alpha + \beta x_i$ abbiamo quanto segue. Quando $x = 0$, il modello diventa

$$\mu_i = \alpha$$

mentre, quando $X = 1$, il modello diventa

$$\mu_i = \alpha + \beta.$$

Ciò significa che il parametro α è uguale al valore atteso del gruppo codificato con $X = 0$ e il parametro β è uguale alla differenza tra le medie dei due gruppi (essendo la media del secondo gruppo uguale a $\alpha + \beta$). Il parametro β, dunque, codifica l'effetto di una manipolazione sperimentale o di un trattamento, e l'inferenza su β corrisponde direttamente all'inferenza sull'efficacia di un trattamento o di un effetto sperimentale.[1] L'inferenza su β, dunque, viene utilizzata per capire quanto "credibile" può essere considerato l'effetto di un trattamento o di una manipolazione sperimentale.

Un esempio concreto

Esaminiamo nuovamente un sottoinsieme di dati tratto dal *National Longitudinal Survey of Youth* i quali sono stati discussi da Gelman et al. (2020). I soggetti sono bambini di 3 e 4 anni. La variabile dipendente, `kid_score`, è il punteggio totale del *Peabody Individual Achievement Test* (PIAT) costituito dalla somma dei punteggi di tre sottoscale (Mathematics, Reading comprehension, Reading recognition). La variabile indipendente, `mom_hs`, è il livello di istruzione della madre, codificato con due livelli: scuola media superiore completata oppure no. La domanda della ricerca è se il QI del figlio (misurato sulla scala PIAT) risulta o meno associato al livello di istruzione della madre.

[1] Per "effetto di un trattamento" si intende la differenza tra le medie di due gruppi (per esempio, il gruppo "sperimentale" e il gruppo "di controllo").

Codifichiamo il livello di istruzione della madre (x) con una *variabile indicatrice* (ovvero, una variabile che assume solo i valori 0 e 1) tale per cui:

- $x = 0$: la madre non ha completato la scuola secondaria di secondo grado (scuola media superiore);
- $x = 1$: la madre ha completato la scuola media superiore.

Supponiamo che i dati siano contenuti nel data.frame `df`.

```
library("foreign")
df <- read.dta(here("data", "kidiq.dta"))
```

Calcoliamo le statistiche descrittive per i due gruppi:

```
df %>%
  group_by(mom_hs) %>%
  summarise(
    mean_kid_score = mean(kid_score),
    std = sqrt(var(kid_score))
  )
#> # A tibble: 2 x 3
#>   mom_hs mean_kid_score   std
#>    <dbl>          <dbl> <dbl>
#> 1      0           77.5  22.6
#> 2      1           89.3  19.0
```

Il punteggio medio PIAT è pari a 77.5 per i bambini la cui madre non ha il diploma di scuola media superiore e pari a 89.3 per i bambini la cui madre ha completato la scuola media superiore. Questa differenza suggerisce un'associazione tra le variabili, ma tale differenza potrebbe essere soltanto la conseguenza della variabilità campionaria, senza riflettere una caratteristica generale della popolazione. Come possiamo usare il modello statistico lineare per fare inferenza sulla differenza osservata tra i due gruppi? Non dobbiamo fare nient'altro che usare lo stesso modello di regressione che abbiamo definito in precedenza.

```
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;
}
model {
  alpha_std ~ normal(0, 2);
```

```
  beta_std ~ normal(0, 2);
  sigma_std ~ cauchy(0, 2);
  y_std ~ normal(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/simpleregstd.stan")
```

Come in precedenza, salviamo i dati in un oggetto di classe `list`:

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

Compiliamo il modello:

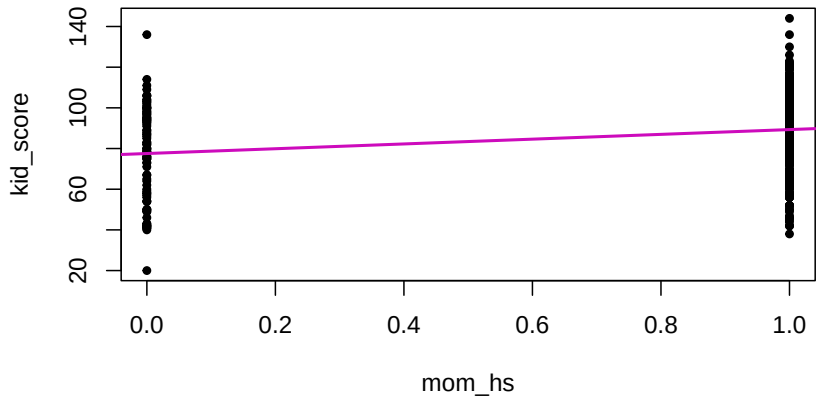
```
file <- file.path("code", "simpleregstd.stan")
mod <- cmdstan_model(file)
```

Adattiamo il modello ai dati:

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

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

```
plot(
  df$kid_score ~ df$mom_hs,
  pch = 20,
  xlab = "mom_hs",
  ylab = "kid_score"
)
abline(mean(posterior$alpha), mean(posterior$beta), col = 6, lw = 2)
```



Le stime a posteriori dei parametri si ottengono con:

```
fit$summary(c("alpha", "beta", "sigma"))
#> # A tibble: 3 x 10
#>   variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
#>   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
#> 1 alpha     77.5   77.6 2.07  2.06  74.2   81.0  1.00   17948.   11861.
#> 2 beta      11.8   11.8 2.34  2.33   7.91  15.6  1.00   18036.   12038.
#> 3 sigma     19.9   19.9 0.679 0.673 18.8   21.0  1.00   18897.   11950.
```

I risultati confermano ciò che ci aspettavamo:

- il coefficiente `alpha` = 77.56 corrisponde alla media del gruppo codificato con $x = 0$, ovvero la media dei punteggi PIAT per i bambini la cui madre non ha completato la scuola media superiore;
- il coefficiente `beta` = 11.76 corrisponde alla differenza tra le medie dei due gruppi, ovvero 89.32 - 77.55 = 11.77 (con piccoli errori di approssimazione).

La seguente chiamata ritorna l'intervallo di credibilità al 95% per tutti i parametri del modello:

```
rstantools::posterior_interval(as.matrix(stanfit), prob = 0.95)
#>                2.5%     97.5%
#> alpha_std   -0.0904    0.0916
#> beta_std     0.1447    0.3289
#> sigma_std    0.9120    1.0437
#> alpha       73.4854   81.6092
#> beta         7.1877   16.3437
#> sigma       18.6155   21.3025
#> lp__      -209.0430 -204.3220
```

Possiamo dunque concludere che i bambini la cui madre ha completato la scuola superiore ottengono in media circa 12 punti in più rispetto ai bambini la cui madre non ha completato la scuola superiore. L'intervallo di credibilità al 95% ci dice che possiamo essere sicuri al 95% che tale differenza sia di almeno 7 punti e possa arrivare fino a ben 16 punti. Per riassumere, possiamo concludere, con un grado di certezza soggettiva del 95%, che c'è un'associazione positiva tra il livello di scolarità della madre e l'intelligenza del bambino: le madri che hanno livello di istruzione più alto della media tendo ad avere bambini il cui QI è anch'esso più alto della media.

5.2 La dimensione dell'effetto

Avendo a disposizione le informazioni sulle distribuzioni a posteriori dei parametri è facile calcolare la dimensione dell'effetto nei termini del d di Cohen:

```
11.75398 / 19.90159
#> [1] 0.591
```

Il d di Cohen di entità "media" [$d > 0.5$; Sawilowsky (2009)] conferma l'importanza dell'influenza della scolarità delle madri sul QI dei bambini.

Bibliografia

Burger, E. B. & Starbird, M. (2012). *The 5 elements of effective thinking*. Princeton University Press. (Cit. a p. viii).

Gelman, A., Hill, J. & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press. (Cit. alle pp. 3, 17, 33).

Hambrick, D. (2015). Research confirms a link between intelligence and life expectancy. *Scientific American. Retrieved from http://www. scientificamerican. com/article/research-confirms-a-link-between-intelligence-and-life-expectancy* (cit. a p. 3).

Horn, S. & Loewenstein, G. (2021). Underestimating Learning by Doing. *Available at SSRN 3941441* (cit. a p. ix).

Sawilowsky, S. S. (2009). New effect size rules of thumb. *Journal of modern applied statistical methods*, *8*(2), 26 (cit. a p. 37).

Elenco delle figure

Abstract This document contains the material of the lessons of Psicometria B000286 (2021/2022) aimed at students of the first year of the Degree Course in Psychological Sciences and Techniques of the University of Florence, Italy.

Keywords Data science, Bayesian statistics.