Data Science per psicologi

Corrado Caudek

2021-09-13

Indice

Capitolo 1

Introduzione alla regressione lineare bayesiana

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 $b = \Delta y$.

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.

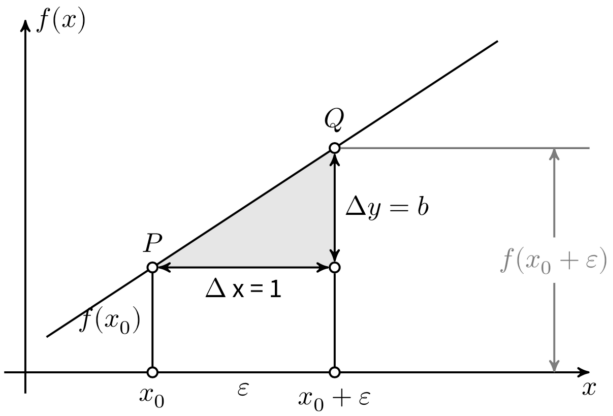


Figura 1.1: *La funzione lineare* $y = a + bx$.

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

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

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.

1.3 Il modello di regressione da una prospettiva bayesiana

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[1], si assegnano distribuzioni a priori ai parametri μ e σ, si genera la verosimiglianza in base ai dati osservati e, con

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

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?

1.3.1 Una media specifica per ciascuna osservazione

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)

1.3.2 Relazione lineare tra la media 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

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

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.

1.3.3 Il modello di regressione lineare

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

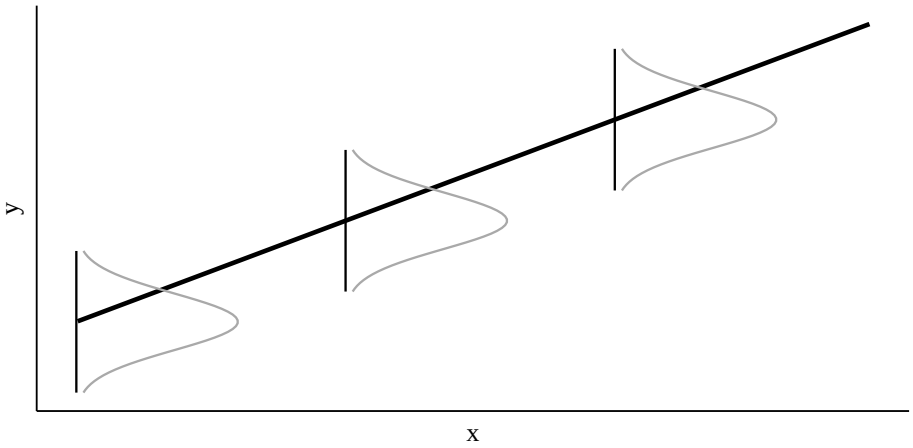
$$Y_i \mid \beta_0, \beta_1, \sigma \overset{ind}{\sim} \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma), \quad i = 1, \dots, n. \tag{1.10}$$

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

Regressione lineare con Stan

Obiettivo di questo Capitolo è illustrare come può essere svolta in pratica l'analisi di regressione lineare bayesiana usando il linguaggio Stan. Per fare un esempio concreto useremo un famoso dataset chiamaro `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 con le seguenti istruzioni R:

```
library("foreign")
df <- read.dta(here("data", "kidiq.dta"))
head(df)
#>   kid_score mom_hs    mom_iq mom_work mom_age
#> 1        65      1 121.11753        4      27
#> 2        98      1  89.36188        4      25
#> 3        85      1 115.44316        4      27
#> 4        83      1  99.44964        3      25
#> 5       115      1  92.74571        4      27
#> 6        98      0 107.90184        1      18
```

Calcoliamo alcune statistiche descrittive usando la funzione `skimr::skim()`:

```
df %>%
  skimr::skim() %>%
  skimr::yank("numeric")
```

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75
kid_score	0	1	86.80	20.41	20.00	74.00	90.00	102.00
mom_hs	0	1	0.79	0.41	0.00	1.00	1.00	1.00
mom_iq	0	1	100.00	15.00	71.04	88.66	97.92	110.27
mom_work	0	1	2.90	1.18	1.00	2.00	3.00	4.00
mom_age	0	1	22.79	2.70	17.00	21.00	23.00	25.00

Dall'output di `skim()` vediamo che 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, iniziamo ad adattare il modello di regressione usando 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
#> -56.753 -12.074   2.217  11.710  47.691
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 25.79978    5.91741    4.36 1.63e-05 ***
#> mom_iq       0.60997    0.05852   10.42  < 2e-16 ***
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 18.27 on 432 degrees of freedom
#> Multiple R-squared:  0.201,  Adjusted R-squared:  0.1991
#> F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

Il modello statistico diventa:

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

dove 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.

Il modello bayesiano descritto sopra può essere specificato usando il 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. Poniamoci dunque il problema di eseguire il campionamento MCMC sulle variabili standardizzate per poi riconvertire i parametri trovati sulla stessa scala di misura dei punteggi grezzi.

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

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

Se uno dei due vettori x o y ha valori molto grandi o molto piccoli o se la media campionaria dei valori è lontana da 0, allora può essere più efficiente standardizzare la variabile risposta y e i predittori x. I dati vengono 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

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

dove la media $\bar{y}$ è

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

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

e la deviazione standard è

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

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:

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

Per eseguire la standardizzare all'interno di un'analisi di regressione, i predittori e la variabile risposta vengono standardizzati. Questa trasformazione cambia la scala delle variabili, e quindi cambia anche la scala delle distribuzioni a priori dei parametri. Consideriamo il seguente modello iniziale specificato con la sintassi richiesta dal 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`.

Il blocco *data* per il modello standardizzato è identico a quello del caso precedente. I predittori e la risposta standardizzati sono definiti nel blocco *transformed data*. Inoltre, 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")
```

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.

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

da cui

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

I valori dei parametri sulle scale originali possono essere calcolati all'interno di Stan utilizzando il blocco *generated quantities* che segue il blocco *model*. 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/corrado/Documents/teaching/2021-22/psicometria/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
```

```
)
#> Running MCMC with 4 chains, at most 2 in parallel...
#>
#> Chain 1 finished in 0.3 seconds.
#> Chain 2 finished in 0.3 seconds.
#> Chain 3 finished in 0.3 seconds.
#> Chain 4 finished in 0.3 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 0.3 seconds.
#> Total execution time: 0.9 seconds.
```

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
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>
#> 1 alpha    25.8   25.9   5.98   5.97   15.8   35.7    1.00
#> 2 beta      0.610  0.609 0.0594 0.0590  0.513  0.709  1.00
#> 3 sigma    18.3   18.3   0.616  0.611  17.3   19.3    1.00
#> # ... with 2 more variables: ess_bulk <dbl>, ess_tail <dbl>
```

Oppure è possibile usare:

```
fit$cmdstan_summary()
#> Inference for Stan model: simpleregstd_model
#> 4 chains: each with iter=(4000,4000,4000,4000); warmup=(0,0,0,0); thin=(1,1,1,1); 16000 itera
#>
#> Warmup took (0.082, 0.083, 0.086, 0.081) seconds, 0.33 seconds total
#> Sampling took (0.24, 0.26, 0.23, 0.22) seconds, 0.94 seconds total
#>
#>                    Mean     MCSE  StdDev        5%       50%    95%    N_Eff  N_Eff/s    R_h
#>
#> lp__            -1.7e+02  1.4e-02     1.2  -1.7e+02  -1.7e+02   -168     8104     8585      1
#> accept_stat__       0.91  7.7e-04    0.11      0.68      0.95    1.0  2.0e+04  2.1e+04  1.0e+
#> stepsize__          0.81  4.4e-02   0.062      0.73      0.83   0.90  2.0e+00  2.1e+00  5.4e+
#> treedepth__          2.2  1.1e-01    0.54       1.0       2.0    3.0  2.3e+01  2.4e+01  1.0e+
#> n_leapfrog__         4.4  2.6e-01     2.0       3.0       3.0    7.0  5.8e+01  6.1e+01  1.0e+
#> divergent__         0.00      nan    0.00      0.00      0.00   0.00      nan      nan      n
#> energy__             171  2.1e-02     1.7       169       171    174  6.9e+03  7.3e+03  1.0e+
```

```
#>
#> alpha_std       -1.2e-04  3.3e-04   0.042  -6.9e-02  -1.2e-04  0.070    16028    16978     1.00
#> beta_std         4.5e-01  3.4e-04   0.044   3.8e-01   4.5e-01   0.52    16421    17396     1.00
#> sigma_std        9.0e-01  2.3e-04   0.030   8.5e-01   9.0e-01   0.95    16890    17892     1.00
#> alpha            2.6e+01  4.7e-02     6.0   1.6e+01   2.6e+01     36    16400    17372     1.00
#> beta             6.1e-01  4.6e-04   0.059   5.1e-01   6.1e-01   0.71    16421    17396     1.00
#> sigma            1.8e+01  4.7e-03    0.62   1.7e+01   1.8e+01     19    16890    17892     1.00
#>
#> Samples were drawn using hmc with nuts.
#> For each parameter, N_Eff is a crude measure of effective sample size,
#> and R_hat is the potential scale reduction factor on split chains (at
#> convergence, R_hat=1).
```

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

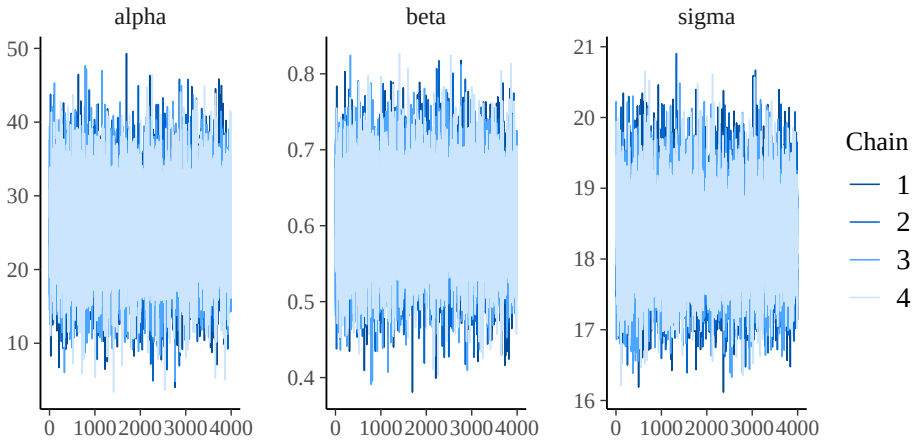
```
fit$cmdstan_diagnose()
#> Processing csv files: /var/folders/cy/4xdvhqx966nggmk95hsnyzc40000gn/T/Rtmpa1NwU5/simpleregstd-202109130
#>
#> Checking sampler transitions treedepth.
#> Treedepth satisfactory for all transitions.
#>
#> Checking sampler transitions for divergences.
#> No divergent transitions 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"))
```



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

Si noti un aspetto 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.

2.1 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 l'intercetta in modo tale che possa essere interpretabile.
- 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, il che indica un sostanziale effetto del QI delle madri sul QI dei loro bambini:

```
(138.89 - 71.04) * 0.61
#> [1] 41.3885
```

- 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.

2.1.1 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
)
#> Running MCMC with 4 chains, at most 2 in parallel...
#>
#> Chain 1 finished in 0.3 seconds.
#> Chain 2 finished in 0.3 seconds.
#> Chain 3 finished in 0.3 seconds.
#> Chain 4 finished in 0.3 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 0.3 seconds.
#> Total execution time: 0.8 seconds.
```

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
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>
#> 1 alpha    86.8   86.8   0.872  0.863  85.4   88.2    1.00
#> 2 beta      0.610  0.609 0.0591 0.0592  0.512  0.708  1.00
#> 3 sigma    18.3   18.3   0.616  0.616  17.3   19.3    1.00
#> # ... with 2 more variables: 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.
Si noti che la nuova intercetta, 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.

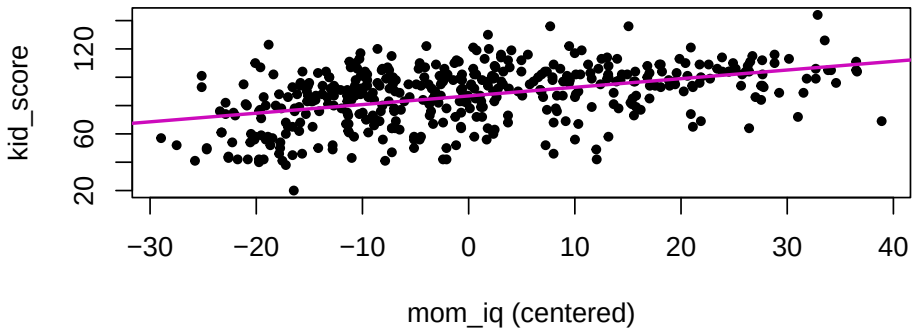
2.1.2 Rappresentazione grafica dell'incertezza della stima

Mediante la funzione `extract()` salvo le stime a posteriori dei parametri in formato `list`:

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

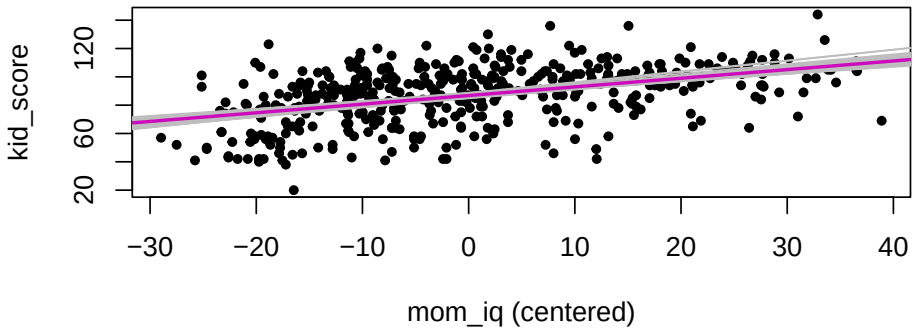
Un diagramma a dispersione dei dati con sovrapposto il valore atteso della y in base al modello bayesiano si ottiene nel modo seguente:

```
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)
```



2.2 Minimi quadrati

Nella trattazione classica del modello di regressione, $y_i = \alpha + \beta x_i + e_i$, i coefficienti $a = \hat{\alpha}$ e $b = \hat{\beta}$ vengono stimati in modo tale da minimizzare i residui

$$e_i = y_i - \hat{\alpha} - \hat{\beta} x_i. \tag{2.2}$$

In altri termini, il residuo i-esimo è la differenza fra l'ordinata del punto (x_i, y_i) e quella del punto di ascissa x_i sulla retta di regressione campionaria.
Per determinare i coefficienti a e b della retta $y_i = a + bx_i + e_i$ non è sufficiente minimizzare la somma dei residui $\sum_{i=1}^{n} e_i$, in quanto i residui possono essere sia positivi che negativi e la loro somma può essere molto prossima allo zero anche per differenze molto grandi tra i valori osservati e la retta di regressione. Infatti, ciascuna retta passante per il punto $(\bar{x}, \bar{y})$ ha $\sum_{i=1}^{n} e_i = 0$.

Una retta passante per il punto $(\bar{x}, \bar{y})$ soddisfa l'equazione $\bar{y} = a + b\bar{x}$. Sottraendo tale equazione dall'equazione $y_i = a + bx_i + e_i$ otteniamo

$$y_i - \bar{y} = b(x_i - \bar{x}) + e_i.$$

Sommando su tutte le osservazioni, si ha che

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - \bar{y}) - b \sum_{i=1}^{n} (x_i - \bar{x}) = 0 - b(0) = 0. \tag{2.3}$$

Questo problema viene risolto scegliendo i coefficienti a e b che minimizzano, non tanto la somma dei residui, ma bensì l'*errore quadratico*, cioè la somma dei quadrati degli errori:

$$S(a, b) = \sum_{i=1}^{n} e_i^2 = \sum (y_i - a - bx_i)^2. \tag{2.4}$$

Il metodo più diretto per determinare quelli che vengono chiamati i *coefficienti dei minimi quadrati* è quello di trovare le derivate parziali della funzione $S(a, b)$ rispetto ai coefficienti a e b:

$$\begin{aligned} \frac{\partial S(a,b)}{\partial a} &= \sum (-1)(2)(y_i - a - bx_i), \\ \frac{\partial S(a,b)}{\partial b} &= \sum (-x_i)(2)(y_i - a - bx_i). \end{aligned} \tag{2.5}$$

Ponendo le derivate uguali a zero e dividendo entrambi i membri per -2 si ottengono le *equazioni normali*

$$\begin{aligned} an + b \sum x_i &= \sum y_i, \\ a \sum x_i + b \sum x_i^2 &= \sum x_i y_i. \end{aligned} \tag{2.6}$$

I coefficienti dei minimi quadrati a e b si trovano risolvendo le (2.6) e sono uguali a:

$$a = \bar{y} - b\bar{x}, \tag{2.7}$$

$$b = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}. \tag{2.8}$$

2.2.1 Massima verosimiglianza

Se gli errori del modello lineare sono indipendenti e distribuiti secondo una Normale, così che $y_i \sim \mathcal{N}(\alpha + \beta x, \sigma^2)$ per ciascun i, allora le stime dei minimi quadrati di α e β corrispondono alla stima di massima verosimiglianza. La funzione di verosimiglianza del modello di regressione è definita come la funzione di densità di probabilità dei dati, dati i parametri e i predittori:

$$p(y \mid \alpha, \beta, \sigma, x) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid \alpha, \beta x_i, \sigma^2). \tag{2.9}$$

Massimizzare la (2.9) conduce alle stime dei minimi quadrati (2.8).

Capitolo 3

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 un compromesso tra tali informazioni precedenti e i dati, moltiplicando la funzione di verosimiglianza con una distribuzione a priori che codifica probabilisticamente le informazioni sui parametri possedute dal ricercatore prima di avere osservato i dati. Il prodotto della verosimiglianza (2.9) e della distribuzione a priori è chiamata distribuzione a posteriori e riassume la nostra conoscenza del parametro *dopo* aver visto 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.

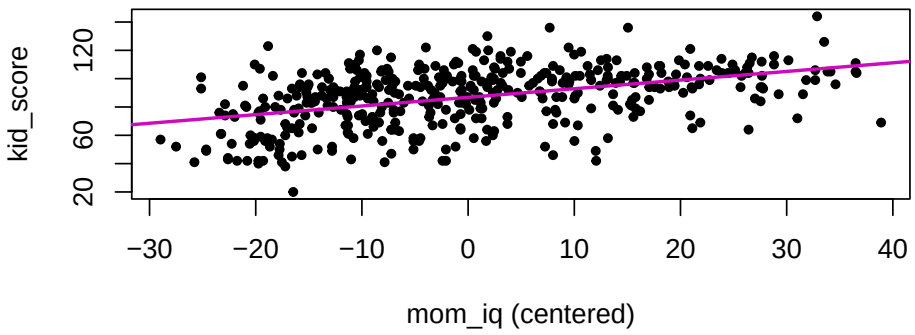
3.1 Rappresentazione grafica dell'incertezza della stima

Un primo modo per rappresentare l'incertezza dell'inferenza in un ottica bayesiana è quella di rappresentare graficamente la retta 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)
```

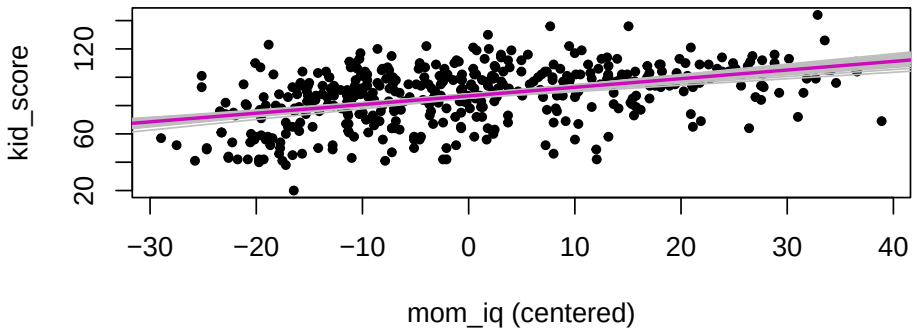
Un diagramma a dispersione dei dati con sovrapposto il valore atteso della y in base al modello bayesiano si ottiene nel modo seguente:

```
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)
```



3.2 Intervalli di credibilità

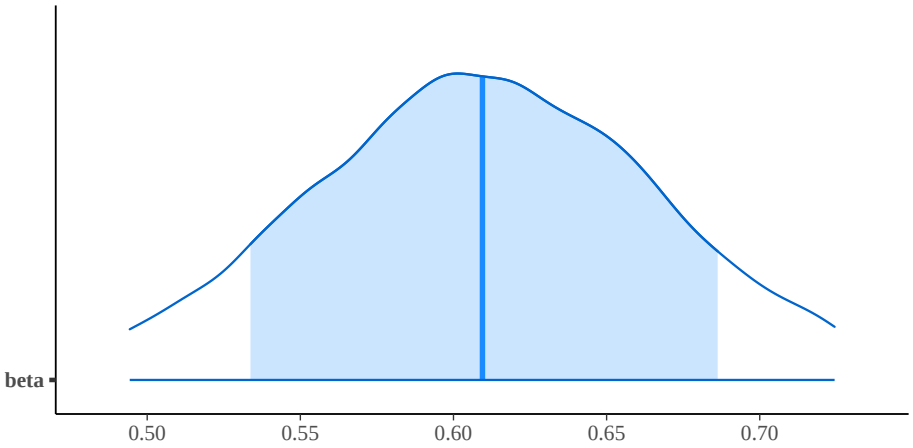
L'incertezza inferenziale sui parametri può essere rappresentata mediante gli *intervalli di credibilità*, ovvero gli intervalli che contengono la quota desiderata (es., il 95%) della distribuzione a posteriori.

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   -0.08381793    0.08454522
#> beta_std     0.36326418    0.53240170
#> sigma_std    0.84043118    0.95801893
#> alpha       85.08649000   88.52285250
#> beta         0.49429853    0.72444565
#> sigma       17.15376250   19.55382000
#> lp__      -172.82802500 -168.24900000
```

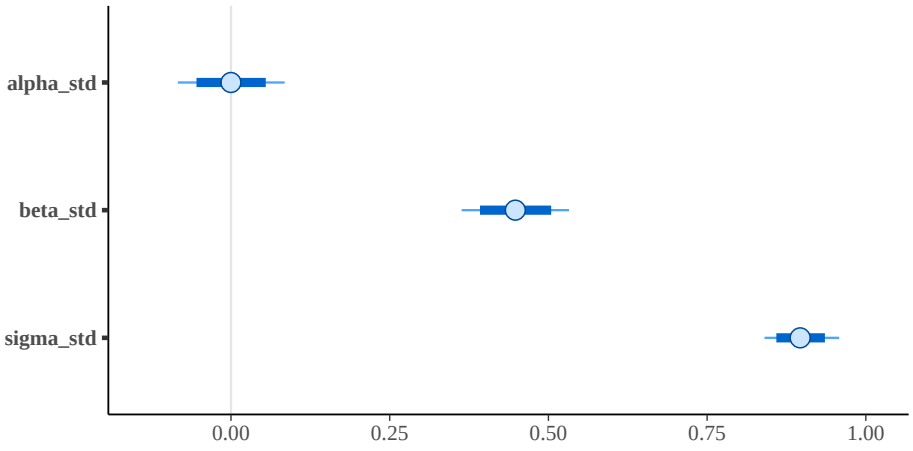
Un grafico che riporta l'intervallo di credibilità ai livelli di probabilità desiderati per β si ottiene con le seguenti istruzioni:

```
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
  )
```



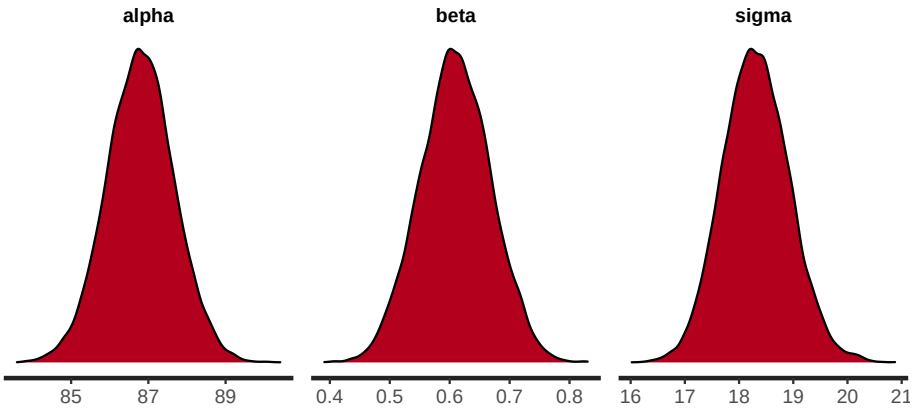
3.3 Rappresentazione grafica della distribuzione a posteriori

Ma non c'è niente di "magico" o necessario relativamente al livello di 0.95: il valore di 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 che stiamo discutendo si ottiene nel modo seguente:

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



3.4 Test di ipotesi

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

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

3.5 Regressione robusta

Spesso i ricercatori devono affrontare il problema degli outlier: un modello statistico basato sulla distribuzione Normale produrrà delle stime dei parametri che non si generalizzano ad altri campioni di dati (ovvero, campioni outlier con le stesse proprietà come nel campione esaminato). Il metodo tradizoinale per affrontare questo problema è quello di eliminare gli outlier prima di eseguire l'analisi statistica. Questo approccio ha però il problema che il criterio utilizzato per eliminare gli outlier, quale esso sia, non può che essere arbitrario. Dunque, usando criteri diversi per eliminare gli outlier i ricercatori finiscno per trovare risultati diversi.

Questo problema trova una semplice soluzione se viene usato l'approccio bayesiano. Nel modello di regressione che abbiamo dicusso finora è stato ipotizzato che gli errori seguono la distribuzione Normale. 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, introduco nel set dei dati un singlo valore anomalo:

```
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 (i risultati sono identici a quelli 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

```
summary(lm(kid_score ~ mom_iq, data = df2))
#>
#> Call:
#> lm(formula = kid_score ~ mom_iq, data = df2)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -599.95  -11.00    4.69   13.64   47.77
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  49.1880    11.0665   4.445 1.12e-05 ***
#> mom_iq        0.3626     0.1093   3.317 0.000987 ***
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 34.38 on 432 degrees of freedom
#> Multiple R-squared:  0.02483,    Adjusted R-squared:  0.02258
#> F-statistic:    11 on 1 and 432 DF,  p-value: 0.0009872
```

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

Un modello che assume una distribuzione Normale dei residui non è l'unico possibile. È altrettanto valido un modello che ipotizza che i residui seguano una diversa distribuzione di densità. Per 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 usiamo la t di Student quale distribuzione dei residui in modello di regressione, ci aspettiamo che le stime dei parametri risultino meno influenzate dalla presenza di outlier di quanto avvenga nel modello Normale.

Per verificare questa intuizione, modifichiamo il codice Stan del modello usato in precedenza. L'unico cambiamento riguarda il fatto che, in questo caso, la distribuzione della y viene ipotizzata seguire una t di Student con un numero ν gradi di libertà stimato dal modello: `student_t(nu, mu, sigma)`.

```
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
#>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>
#> 1 alpha    87.8   87.8   0.887  0.899  86.3   89.3    1.00
#> 2 beta      0.603  0.603 0.0585 0.0577  0.506  0.698  1.00
#> 3 sigma    15.9   15.9   0.806  0.812  14.6   17.3    1.00
#> 4 nu        5.58   5.44  1.15   1.11    3.94   7.65   1.00
#> # ... with 2 more variables: ess_bulk <dbl>, ess_tail <dbl>
```

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

Capitolo 4

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.

4.1 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 alla media 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. 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"). L'inferenza su β, dunque, viene utilizzata per capire quanto "robusto" può essere considerato l'effetto di un trattamento o di una manipolazione sperimentale.

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.

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:

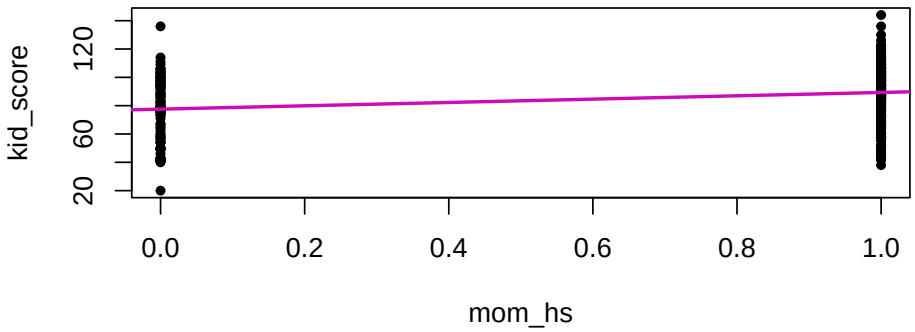
```
file <- file.path("code", "simpleregstd.stan")
mod <- cmdstan_model(file)
```

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
#>   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 alpha     77.5   77.6 2.07  2.06  74.2   81.0  1.00
#> 2 beta      11.8   11.8 2.34  2.33   7.91  15.6  1.00
#> 3 sigma     19.9   19.9 0.679 0.673 18.8   21.0  1.00
#> # ... with 2 more variables: ess_bulk <dbl>, ess_tail <dbl>
```

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).

Possiamo ottenere l'intervallo di credibilità al 95% per `beta`:

```
rstantools::posterior_interval(as.matrix(stanfit), prob = 0.95)
#>                   2.5%         97.5%
#> alpha_std   -0.0903768    0.09155991
#> beta_std     0.1446650    0.32894522
#> sigma_std    0.9120478    1.04369125
#> alpha       73.4853725   81.60923250
#> beta         7.1877393   16.34371250
#> sigma       18.6154975   21.30252500
#> lp__      -209.0430250 -204.32200000
```

Il coefficiente b ci dice 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 è di almeno 7 punti e può arrivare fino a ben 16 punti. Possiamo dunque concludere, con un grado di certezza soggettiva del 95%, che c'è un'associazione tra il livello di scolarità della madre e l'intelligenza del bambino: i bambini tendono ad avere un livello di intelligenza più elevato se le loro madri hanno un livello di istruzione maggiore.

4.2 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] 0.5906051
```

Nei termini del d di Cohen possiamo dunque concludere che la grandezza dell'effetto è di entità "media" [$d > 0.5$; x@sawilowsky2009new].

Bibliografia

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

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*.