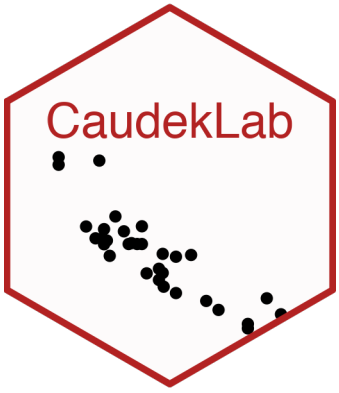
Psicometria

Corrado Caudek

2022-01-15

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: Gennaio 15, 2022.

Prefazione

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

La psicologia e la Data science

> *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 psicologico sensato senza avere almeno una comprensione rudimentale della Data science. Le tematiche della Data science non possono essere ignorate né dai ricercatori in psicologia né da coloro che svolgono la professione di psicologo al di fuori dell'Università. Infatti, anche i professionisti al di fuori dall'università non possono fare a meno di leggere la letteratura psicologica più recente: il continuo aggiornamento delle conoscenze è infatti richiesto dalla deontologia della professione. Ma per potere fare questo è necessario conoscere un bel po' di Data science! Basta aprire a caso una rivista specialistica di psicologia per rendersi conto di quanto ciò sia vero: gli articoli che riportano i risultati delle ricerche psicologiche sono zeppi di analisi statistiche e di modelli formali. E la comprensione della letteratura psicologica rappresenta un requisito minimo nel bagaglio professionale dello psicologo.

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

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

non capiamo i dettagli di qualcosa, è quello di pensare: “non importa, ho capito in maniera approssimativa questo punto, non devo preoccuparmi del resto”. Ma in realtà non è vero: se la nostra comprensione è superficiale, quando il problema verrà presentato in una nuova forma, non riusciremo a risolverlo. Per cui i dubbi che ci vengono quando studiamo qualcosa sono il nostro alleato più prezioso: ci dicono esattamente quali sono gli aspetti che dobbiamo approfondire per potere migliorare la nostra preparazione.

- È utile sviluppare una visione d’insieme degli argomenti trattati, capire l’obiettivo generale che si vuole raggiungere e avere chiaro il contributo che i vari pezzi di informazione forniscono al raggiungimento di tale obiettivo. Questa organizzazione mentale del materiale di studio facilita la comprensione. È estremamente utile creare degli schemi di ciò che si sta studiando. Non aspettate che sia io a fornirvi un riepilogo di ciò che dovete imparare: sviluppate da soli tali schemi e tali riassunti.
- Tutti noi dobbiamo imparare l’arte di trovare le informazioni, non solo nel caso di questo insegnamento. Quando vi trovate di fronte a qualcosa che non capite, o ottenete un oscuro messaggio di errore da un software, ricordatevi: “Google is your friend”.

Corrado Caudek

Capitolo 1

Distribuzione predittiva a posteriori

Oltre ad una sintesi della distribuzione a posteriori attraverso il computo di indici caratteristici e alla verifica di ipotesi, un altro compito dell'analisi bayesiana è la predizione di nuovi dati futuri. Dopo aver osservato i dati di un campione e ottenuto le distribuzioni a posteriori dei parametri, è infatti possibile ottenere una qualche indicazione su come potrebbero essere i dati futuri. L'uso più immediato della stima della distribuzione dei possibili valori futuri della variabile di esito è la verifica del modello. Infatti, il modo più diretto per testare un modello è quello di utilizzare il modello per fare previsioni sui possibili dati futuri per poi confrontare i dati predetti con i dati effettivi. Questa pratica va sotto il nome di controllo predittivo a posteriori.

1.1 La distribuzione dei possibili valori futuri

La distribuzione dei possibili valori futuri della variabile di esito può essere predetta da un modello statistico sulla base della distribuzione a posteriori dei parametri, $p(\theta \mid y)$, avendo già osservato n manifestazioni del fenomeno y. Una tale distribuzione va sotto il nome di *distribuzione predittiva a posteriori* (*posterior predictive distribution*, PPD).

Quando vengono simulate le osservazioni della distribuzione predittiva a posteriori si usa la notazione y^{rep} (dove *rep* sta per *replica*) quando, nella simulazione, vengono utilizzate le stesse osservazioni di X che erano state usate per stimare i parametri del modello. Si usa invece la notazione $\tilde{y}$ per fare riferimento a possibili valori X che non sono contenuti nel campione osservato, ovvero, ad un campione di dati che potrebbe essere osservato in qualche futura occasione.

La distribuzione predittiva a posteriori viene usata per fare inferenze predittive. L'idea è che, se il modello ben si adatta bene ai dati del campione allora, sulla base dei parametri stimati, dovremmo essere in grado di generare nuovi dati non osservati y^{rep} che risultano molto simili ai dati osservati y. I dati y^{rep} vengono concepiti come stime di $\tilde{y}$.

La distribuzione predittiva a posteriorie è data da:

$$p(\tilde{y} \mid y) = \int_{\theta} p(\tilde{y}, \theta \mid y) d\theta = \int_{\theta} p(\tilde{y} \mid \theta, y) p(\theta \mid y) d\theta.$$

Supponendo che le osservazioni passate e future siano condizionalmente indipendenti dato θ, ovvero che $p(\tilde{y} \mid \theta, y) = p(\tilde{y} \mid \theta)$, possiamo scrivere

$$p(\tilde{y} \mid y) = \int_{\theta} p(\tilde{y} \mid \theta) p(\theta \mid y) d\theta. \tag{1.1}$$

La (1.1) descrive la nostra incertezza sulla distribuzione di future osservazioni di dati, data la distribuzione a posteriori di θ, ovvero tenendo conto della scelta del modello e della stima dei parametri mediante i dati osservati. Si noti che, nella (1.1), $\tilde{y}$ è condizionato da y ma non da ciò che è incognito, ovvero θ. La distribuzione predittiva a

posteriori è invece ottenuta mediante marginalizzazione sopra le variabili da "scartare", ovvero sopra i parametri incogniti θ.

Un esempio formulato mediante il codice Stan può chiarire questo concetto. Consideriamo il codice relativo alla distribuzione predittiva a posteriori nel caso di un modello di regressione lineare classico con un solo predittore x. Il blocco *Model* sarà:

```
model {
 y ~ normal(x * beta + alpha, sigma);
}
```

Quello che è di interesse per la discussione presente è il blocco *Generated Quantities*. Tale blocco avrà questa forma:

```
generated quantities {
 real y_rep[N];

 for (n in 1:N) {
   y_rep[n] = normal_rng(x[n] * beta + alpha, sigma);
 }

}
```

La variabile `y_rep` è ciò a cui siamo interessati. Nel codice precedente, `x` è il vettore che contiene i valori della variabile indipendente nel campione di osservazioni esaminato. I parametri del modello di regressione sono `alpha` e `beta`; `sigma` è la stima dell'errore standard della regressione. Supponiamo che questi tre parametri siano degli scalari. Se lo fossero, per il valore `x` n-esimo, l'istruzione `normal_rng()` ritornerebbe un valore a caso dalla distribuzione normale con media $\alpha + \beta x_n$ e deviazione standard σ. Il ciclo `for()` ripete questa operazione N volte, ovvero tante volte quanti sono gli elementi del vettore `x` del campione. Quello che è stato detto sopra ci dà un'idea di quello che succederebbe se `alpha`, `beta` e `sigma` fossero degli scalari. Ma non lo sono. Per ciascuno dei tre paramatri abbiamo un numero molto alto di stime, ovvero l'approssimazione MCMC della distribuzione a posteriori. Poniamo che l'ampiezza campionaria N sia 30. Se `alpha`, `beta` e `sigma` fossero degli scalari, la distribuzione predittiva a posteriori sarebbe costituita da 30 valori y^{rep}, ovvero, non sarebbe nient'altro che $\widehat{y} = \widehat{\alpha} + \widehat{\beta} x$. Ma `alpha`, `beta` e `sigma` non sono degli scalari: per ciascuno di questi parametri abbiamo un grande numero di stime, diciamo 2000. Dunque, quando `normal_rng()` estrae un valore a caso dalla distribuzione normale, i parametri della normale non sono fissi: per determinare μ prendiamo un valore a caso, chiamiamolo `beta'`, dalla distribuzione dei valori `beta` e un valore a caso, chiamiamolo `alpha'`, dalla distribuzione dei valori `alpha`. Avendo questi due valori, calcoliamo il valore $\mu'_n = \alpha' + \beta' x_n$. Lo stesso si può dire per σ'. A questo punto possiamo trovare il valore `y_n'` estraendo un valore a caso dalla distribuzione gaussiana di parametri μ' e σ'. Per l'n-esimo valore x possiamo ripetere questo processo tante volte. Se lo ripetiamo, ad esempio, 2,000 volte, per tutti e 30 i valori x del campione otterremo una matrice $30 \times 2,000$. In questo modo possiamo generare le previsioni del modello, ovvero y^{rep}, che includono due fonti di incertezza:

- la variabilità campionaria, ovvero il fatto che abbiamo osservato uno specifico insieme di valori (x, y); in un altro campione tali valori saranno diversi;
- la variabilità a posteriori della distribuzione dei parametri, ovvero il fatto che di ciascun parametro non conosciamo il "valore vero" ma solo una distribuzione (a posteriori) di valori.

Nel caso dell'esempio presente, l'integrale della (1.1) può essere interpretato dicendo che, nell'esempio della matrice di dimensioni $30 \times 2,000$, noi marginalizziamo rispetto

alle colonne, ovvero, per ciascuna riga facciamo la media dei valori colonna. Otteniamo così un vettore di 30 osservazioni, ovvero y^{rep}.

Quando, con metodi grafici, vengono esaminati i valori della distribuzione predittiva a posteriori, possiamo esaminare un numero arbitrario di previsioni. Per esempio, possiamo rappresentare graficamente 50 rette di regressione predette – o un qualsiasi altro numero. Questa rappresentazione grafica quantifica la nostra incertezza a posteriori relativamente (in questo esempio) all'orientamento della retta di regressione.

Esercizio 1.1. Illustreremo ora il problema di trovare la distribuzione $p(\tilde{y} \mid y)$ in un caso semplice, ovvero quello dello schema Beta-Binomiale. Nell'esempio, useremo un'altra volta i dati del campione di pazienti clinici depressi di Zetsche et al. (2019) – si veda l'Appendice **??**. Supponendo di volere esaminare in futuro altri 30 pazienti clinici, ci chiediamo: quanti di essi manifesteranno una depressione grave?

Se vogliamo fare predizioni su $\tilde{y}$ (il numero di "successi" previsti futuri) dobbiamo innanzitutto riconoscere che i possibili valori $\tilde{y} \in \{0, 1, \dots, 30\}$ non sono tutti egualmente plausibili. Sappiamo che $\tilde{y}$ è una v.c. binomiale con distribuzione

$$p(\tilde{y} \mid \theta) = \binom{30}{\tilde{y}} \theta^{\tilde{y}} (1-\theta)^{30-\tilde{y}} . \tag{1.2}$$

La v.c. $\tilde{y}$ dipende da θ, ma il parametro θ è esso stesso una variabile casuale. Avendo osservato $y = 23$ successi in $n = 30$ prove nel campione (laddove la presenza di una depressione grave è stata considerata un "successo"), e avendo assunto come distribuzione a priori per θ una Beta$(2, 10)$ (per continuare con l'esempio precedente), la distribuzione a posteriori di θ sarà una Beta$(25, 17)$:

```
bayesrules::summarize_beta_binomial(alpha = 2, beta = 10, y = 23, n = 30)
#>       model alpha beta  mean mode    var     sd
#> 1     prior     2   10 0.167  0.1 0.0107 0.1034
#> 2 posterior    25   17 0.595  0.6 0.0056 0.0749
```

Per trovare la distribuzione sui possibili dati previsti futuri $\tilde{y}$ dobbiamo applicare la (1.1):

$$p(\tilde{y} \mid y = 23) = \int_0^1 p(\tilde{y} \mid \theta) p(\theta \mid y = 23) d\theta . \tag{1.3}$$

Per il modello Beta-Binomiale è possibile trovare una soluzione analitica alla (1.1).

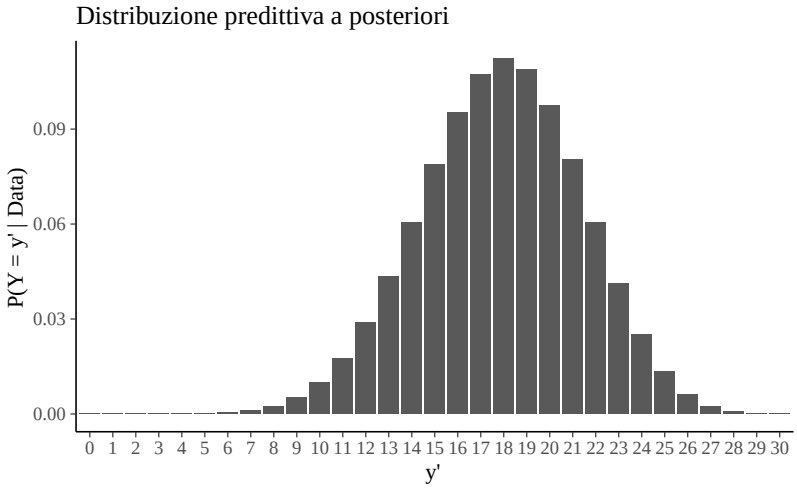
Poniamo di avere osservato y successi in n prove e di utilizzare una distribuzione a priori Beta(a, b). Possiamo scrivere

$$\begin{aligned}
p(\tilde{y} \mid y) &= \int_0^1 p(\tilde{y} \mid \theta) p(\theta \mid y) \, d\theta \\
&= \int_0^1 \binom{\tilde{n}}{\tilde{y}} \theta^{\tilde{y}} (1-\theta)^{\tilde{n}-\tilde{y}} \text{Beta}(a+y, b+n-y) \, d\theta \\
&= \binom{\tilde{n}}{\tilde{y}} \int_0^1 \theta^{\tilde{y}} (1-\theta)^{\tilde{n}-\tilde{y}} \frac{1}{B(a+y, b+n-y)} \theta^{a+y-1} (1-\theta)^{b+n-y-1} \\
&= \binom{\tilde{n}}{\tilde{y}} \frac{1}{B(a+y, b+n-y)} \int_0^1 \theta^{\tilde{y}+a+y-1} (1-\theta)^{\tilde{n}-\tilde{y}+b+n-y-1} \\
&= \binom{\tilde{n}}{\tilde{y}} \frac{B(\tilde{y}+a+y, b+n-y+\tilde{n}-\tilde{y})}{B(a+y, b+n-y)} .
\end{aligned} \tag{1.4}$$

Svolgendo i calcoli in R, per i dati dell'esempio otteniamo:

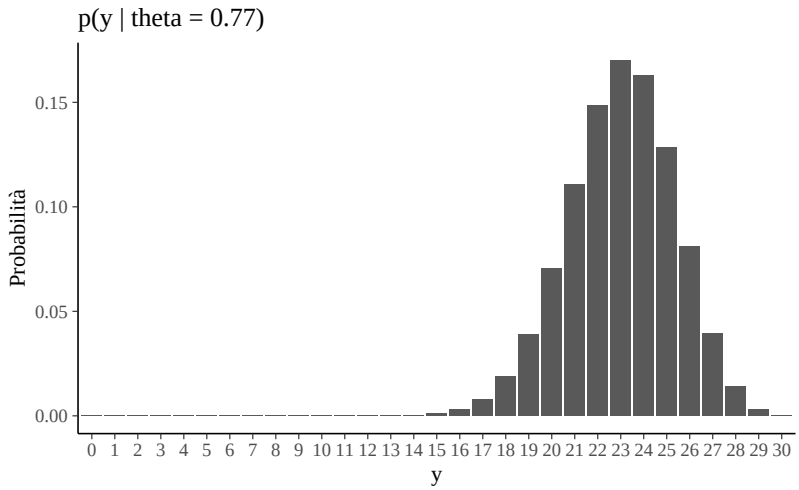
```r
beta_binom <- function(rp) {
  val <- choose(np, rp) *
    beta(rp + a + y, b + n - y + np - rp) /
    beta(a + y, b + n - y)
  val
}

n <- 30
y <- 23
a <- 2
b <- 10
np <- 30
data.frame(
  heads = 0:np,
  pmf = beta_binom(0:np)
) %>%
  ggplot(aes(x = factor(heads), y = pmf)) +
  geom_col() +
  labs(
    title = "Distribuzione predittiva a posteriori",
    x = "y'",
    y = "P(Y = y' | Data)"
  )
```



È facile vedere come, in questo esempio, la distribuzione predittiva a posteriori $p(\tilde{y} \mid y)$ sia diversa dalla binomiale di parametro $\theta = 23/30$:

```r
tibble(
  heads = 0:np,
  pmf = dbinom(x = 0:np, size = np, prob = 23 / 30)
) %>%
  ggplot(aes(x = factor(heads), y = pmf)) +
  geom_col() +
  labs(
    title = "p(y | theta = 0.77)",
    x = "y",
    y = "Probabilità"
  )
```



In particolare, la $p(\tilde{y} \mid y)$ ha una varianza maggiore di $\text{Bin}(y \mid \theta = 0.77, n = 30)$. Questa maggiore varianza riflette le due fonti di incertezza che sono presenti nella (1.1): l'incertezza sul valore del parametro (descritta dalla distribuzione a posteriori) e l'incertezza dovuta alla variabilità campionaria (descritta dalla funzione di verosimiglianza). Possiamo concludere la discussione di questo esempio dicendo che, nel caso di 30 nuovi pazienti clinici, alla luce delle nostre credenze precedenti e dei dati osservati nel campione, ci aspettiamo di osservare 18 pazienti con una depressione severa, anche se è ragionevole aspettarci un numero compreso, diciamo, tra 10 e 25.

Una volta trovata la distribuzione predittiva a posteriori $p(\tilde{y} \mid y)$ diventa possibile rispondere a domande come: qual è la probabilità di depressione grave in almeno 10 dei 30 pazienti futuri? Rispondere a domande di questo tipo è possibile, ma richiede un po' di lavoro. Tuttavia, non è importante imparare scrivere il codice necessario a risolvere problemi di questo tipo perché, in generale, anche per problemi solo leggermente più complessi di quello discusso qui, non sono disponibili espressioni analitiche della distribuzione predittiva a posteriori. Invece, è possibile trovare una approssimazione numerica della $p(\tilde{y} \mid y)$ mediante simulazioni MCMC. Inoltre, se viene utilizzato un tale metodo, risulta facile rispondere a domande simili a quella che abbiamo presentato sopra.

1.2 Metodi MCMC per la distribuzione predittiva a posteriori

Se svolgiamo l'analisi bayesiana con il metodo MCMC, le repliche $p(y^{rep} \mid y)$ (ovvero le stime delle possibili osservazioni future $p(\tilde{y} \mid y)$) possono essere ottenute nel modo seguente:

- campionare $\theta_i \sim p(\theta \mid y)$, ovvero campionare un valore del parametro dalla distribuzione a posteriori;
- campionare $y^{rep} \sim p(y^{rep} \mid \theta_i)$, ovvero campionare il valore di un'osservazione dalla funzione di verosimiglianza condizionata al valore del parametro definito nel passo precedente.

Se i due passaggi descritti sopra vengono ripetuti un numero sufficiente di volte, l'istogramma risultante approssimerà la distribuzione predittiva a posteriori che, in teoria (ma non in pratica) potrebbe essere ottenuta per via analitica (si veda il Paragrafo **??**).

Esercizio 1.2. Generiamo ora $p(y^{rep} \mid y)$ nel caso dell'inferenza su una proporzione.

Riportiamo qui sotto il codice Stan — si veda il Capitolo **??**.

```
modelString = "
data {
  int<lower=0> N;
  int<lower=0, upper=1> y[N];
}
parameters {
  real<lower=0, upper=1> theta;
}
model {
  theta ~ beta(2, 10);
  y ~ bernoulli(theta);
}
generated quantities {
  int y_rep[N];
  real log_lik[N];
  for (n in 1:N) {
    y_rep[n] = bernoulli_rng(theta);
    log_lik[n] = bernoulli_lpmf(y[n] | theta);
  }
}
"
writeLines(modelString, con = "code/betabin23-30-2-10.stan")
```

Si noti che nel nel blocco `generated quantities` sono state aggiunte le istruzioni necessarie per simulare y^{rep}, ovvero, `y_rep[n] = bernoulli_rng(theta)`. I dati dell'esempio sono:

```
data_list <- list(
  N = 30,
  y = c(rep(1, 23), rep(0, 7))
)
```

Compiliamo il codice Stan

```
file <- file.path("code", "betabin23-30-2-10.stan")
mod <- cmdstan_model(file)
```

ed eseguiamo il campionamento MCMC:

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

Per comodità, trasformiamo l'oggetto `fit` in un oggetto di classe `stanfit`:

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

Il contenuto dell'oggetto `stanfit` può essere esaminato nel modo seguente:

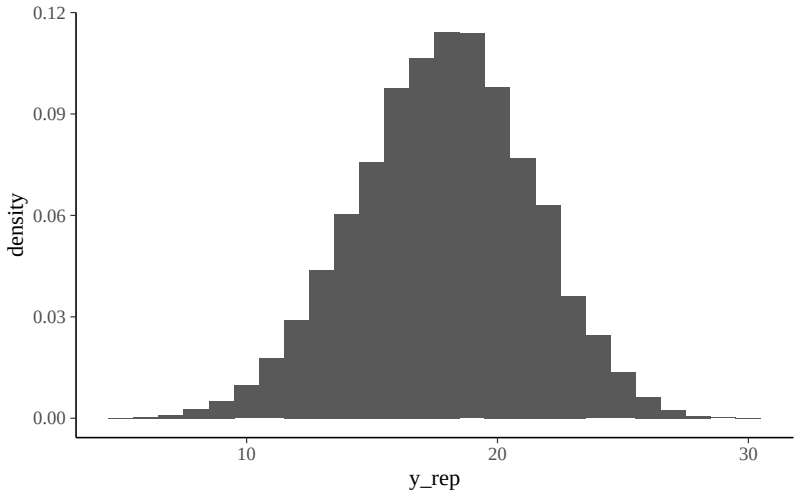
```
list_of_draws <- extract(stanfit)
print(names(list_of_draws))
#> [1] "theta"   "y_rep"   "log_lik" "lp__"
```

Dall'oggetto `list_of_draws` recuperiamo `y_rep`:

```
y_bern <- list_of_draws$y_rep
dim(y_bern)
#> [1] 16000    30
head(y_bern)
#>
#> iterations [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
#>       [1,]    0    1    1    0    0    1    1    1    1     1     1
#>       [2,]    1    0    1    0    1    1    1    1    1     1     1
#>       [3,]    1    1    0    1    1    1    0    1    1     0     0
#>       [4,]    0    1    1    1    0    1    1    1    1     1     1
#>       [5,]    0    1    1    0    0    1    1    0    1     0     1
#>       [6,]    1    1    1    0    0    0    0    1    0     1     1
#>
#> iterations [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
#>       [1,]     0     0     0     1     1     0     1     0     1
#>       [2,]     1     1     1     0     1     0     0     1     1
#>       [3,]     1     1     1     1     1     1     1     1     1
#>       [4,]     1     0     1     0     0     1     1     1     1
#>       [5,]     1     0     1     1     1     0     1     1     1
#>       [6,]     1     1     0     1     1     0     0     1     0
#>
#> iterations [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29]
#>       [1,]     1     0     1     0     0     0     1     1     1
#>       [2,]     1     0     1     1     1     1     0     1     1
#>       [3,]     0     1     0     1     0     1     0     1     1
#>       [4,]     1     1     1     1     1     0     0     1     0
#>       [5,]     0     0     1     1     1     1     0     1     0
#>       [6,]     0     1     0     1     1     1     1     1     1
#>
#> iterations [,30]
#>       [1,]     0
#>       [2,]     1
#>       [3,]     0
#>       [4,]     1
#>       [5,]     0
#>       [6,]     0
```

Dato che il codice Stan definisce un modello per i dati grezzi (ovvero, per ciascuna singola prova Bernoulliana del campione), ogni riga di `y_bern` include 30 colonne, ciascuna delle quali corrisponde ad un campione ($n = 16000$ in questa simulazione) di possibili valori futuri $y_i \in \{0, 1\}$. Per ottenere una stima della distribuzione predittiva a posteriori `p(y_rep)`, ovvero, una stima della probabilità associata a ciascuno dei possibili numeri di "successi" in $N = 30$ nuove prove future, è sufficiente calcolare la proporzione di valori 1 in ciascuna riga:

```
tibble(y_rep = rowSums(y_bern)) %>%
  ggplot(aes(x = y_rep, after_stat(density))) +
  geom_histogram(binwidth = 1)
```



1.3 Posterior predictive checks

La distribuzione predittiva a posteriori viene utilizzata per eseguire i cosiddetti *controlli predittivi a posteriori* (*Posterior Predictive Checks*, PPC). Ricordiamo che la distribuzione predittiva a posteriori corrisponde alla simulazione di un campione di dati generati utilizzando le proprietà del modello adattato. Nei PPC si realizza un confronto grafico tra $p(y^{rep} \mid y)$ e i dati osservati y. Confrontando visivamente gli aspetti chiave dei dati previsti futuri y^{rep} e dei dati osservati y possiamo determinare se il modello è adeguato.

Oltre al confronto tra le distribuzioni $p(y)$ e $p(y^{rep})$ è anche possibile un confronto tra la distribuzione di varie statistiche descrittive, i cui valori sono calcolati su diversi campioni y^{rep}, e le corrispondenti statistiche descrittive calcolate sui dati osservati. Vengono solitamente considerate statistiche descrittive quali la media, la varianza, la deviazione standard, il minimo o il massimo. Ma confronti di questo tipo sono possibili per qualunque statistica descrittiva. Questi confronti sono chiamati PPC.

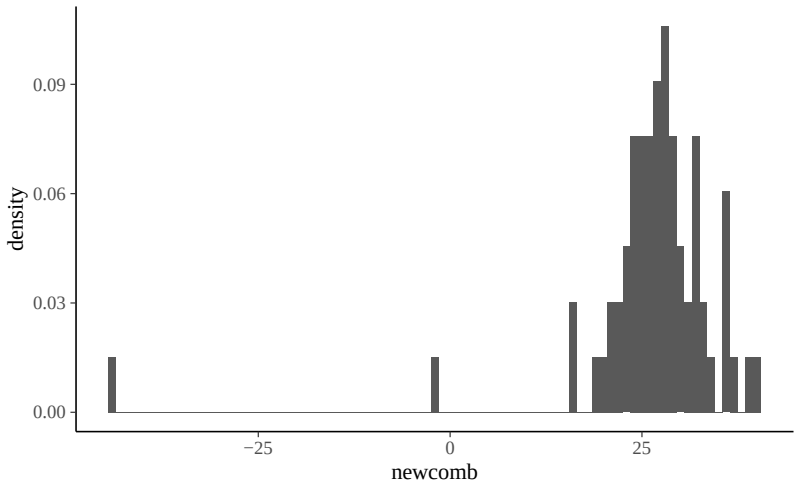
Esercizio 1.3. Esaminiamo ora un set di dati che non seguono la distribuzione normale (Gelman et al., 2020). I dati corrispondono ad una serie di misurazioni prese da Simon Newcomb nel 1882 come parte di un esperimento per stimare la velocità della luce. A questi dati verrà (inappropriatamente) adattata una distribuzione normale. L'obiettivo dell'esempio è quello di mostrare come i PPC possono rivelare la mancanza di adattamento di un modello ai dati.

I PPC mostrano che il modo più semplice per verificare l'adattamento del modello è quello di visualizzare y^{rep} insieme ai dati effettivi. Iniziamo a caricare i dati:

```
library("MASS")
data("newcomb")
```

Generiamo un istogramma per visualizzare i dati:

```
tibble(newcomb) %>%
  ggplot(aes(x = newcomb, after_stat(density))) +
  geom_histogram(binwidth = 1)
```



Creiamo un oggetto di tipo `list` dove inserire i dati:

```
data_list <- list(
  y = newcomb,
  N = length(newcomb)
)
```

Il codice Stan per il modello normale è il seguente:

```
modelString <- "
data {
  int<lower=0> N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=0> sigma;
}
model {
  mu ~ normal(25, 10);
  sigma ~ cauchy(0, 10);
  y ~ normal(mu, sigma);
}
generated quantities {
  vector[N] y_rep;
  for (n in 1:N) {
    y_rep[n] = normal_rng(mu, sigma);
  }
}
"
writeLines(modelString, con = "code/newcomb.stan")
```

Adattando il modello ai dati

```
file <- file.path("code", "newcomb.stan")
mod <- cmdstan_model(file)
fit <- mod$sample(
  data = data_list,
```

```
  iter_sampling = 4000L,
  iter_warmup = 2000L,
  seed = SEED,
  chains = 4L,
  cores = 4L,
  refresh = 0,
  thin = 1
)
```

otteniamo le seguenti stime dei parametri μ e σ:

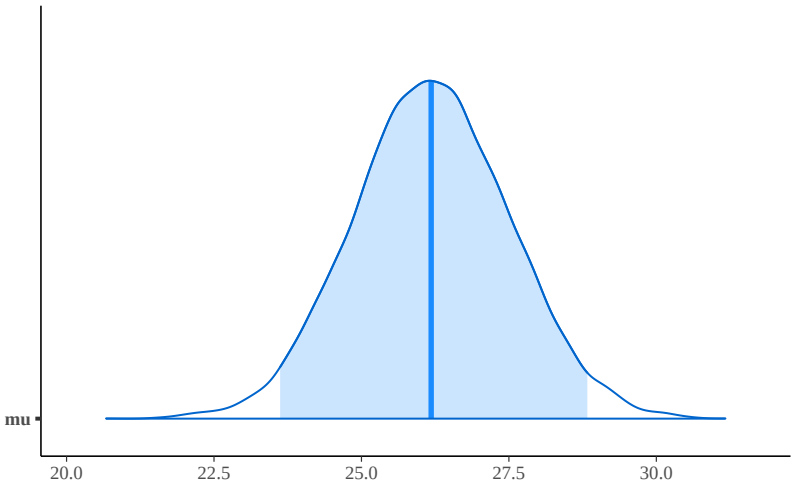
```
fit$summary(c("mu", "sigma"))
#> # A tibble: 2 × 10
#>   variable  mean median    sd   mad    q5   q95  rhat ess_bulk
#>   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
#> 1 mu        26.2   26.2 1.33  1.32  24.0   28.4  1.00   12233.
#> 2 sigma     10.9   10.8 0.973 0.953  9.39  12.6  1.00   12499.
#> # … with 1 more variable: ess_tail <dbl>
```

Trasformiamo `fit` in un oggetto `stanfit`:

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

La distribuzione a posteriori di μ è

```
mu_draws <- as.matrix(stanfit, pars = "mu")
mcmc_areas(mu_draws, prob = 0.95) # color 95% interval
```



Confrontiamo μ con la media di y:

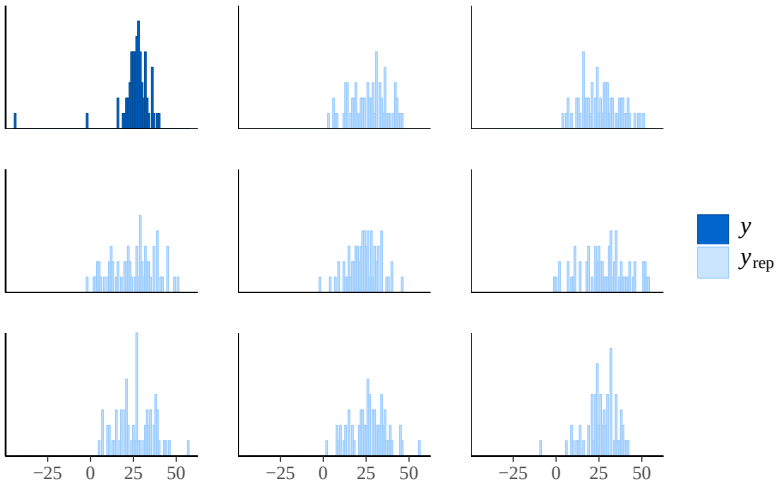
```
mean(newcomb)
#> [1] 26.2
```

Anche se trova la media giusta, il modello non è comunque adeguato a prevedere le altre proprietà della y. Estraiamo y^{rep} dall'oggetto `stanfit`:

```
y_rep <- as.matrix(stanfit, pars = "y_rep")
dim(y_rep)
#> [1] 16000    66
```

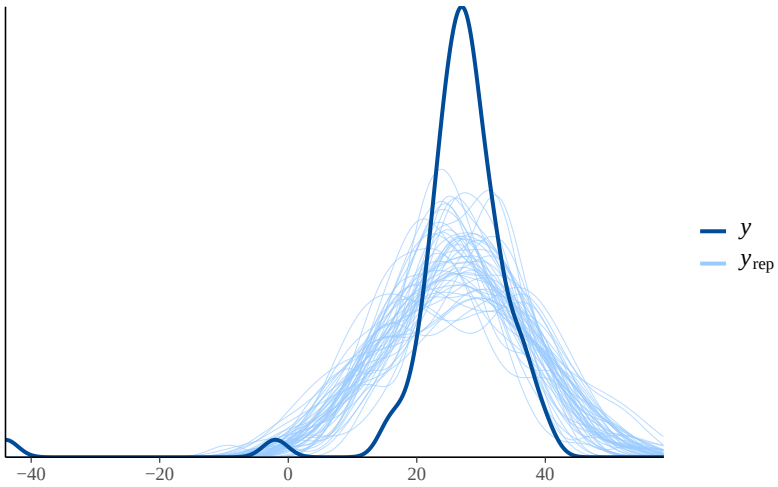
I valori `y_rep` sono i dati della distribuzione predittiva a posteriori che sono stati simulati usando gli stessi valori X dei predittori utilizzati per adattare il modello. Il confronto tra l'istogramma della y e gli istogrammi di diversi campioni y^{rep} mostra una scarsa corrispondenza tra i due:

```
ppc_hist(data_list$y, y_rep[1:8, ], binwidth = 1)
```



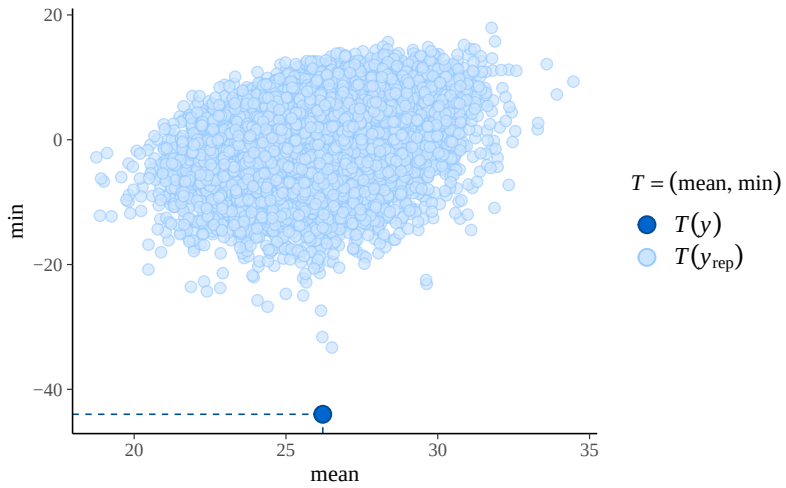
Alla stessa conclusione si giunge tramite un confronto tra la funzione di densità empirica della y e quella di diversi campioni y^{rep}:

```
ppc_dens_overlay(data_list$y, y_rep[1:50, ])
```



Generiamo ora i PPC per la media e il minimo della distribuzione:

```
ppc_stat_2d(data_list$y, y_rep, stat = c("mean", "min"))
```



Mentre la media viene riprodotta accuratamente dal modello (come abbiamo visto sopra), ciò non è vero per il minimo dela distribuzione. L'origine di questa mancanza di adattamento è il fatto che la distribuzione delle misurazioni della velocità della luce è asimmetrica negativa. Dato che ci sono poche osservazioni nella coda negativa della distribuzione, solo per fare un esempio, utilizzeremo ora un secondo modello che ipotizza una distribuzione t di Student:

```
modelString = "
data {
  int<lower=0> N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=0> sigma;
  real<lower=0> nu;
}
model {
  mu ~ normal(25, 10);
  sigma ~ cauchy(0, 10);
  nu ~ cauchy(0, 10);
  y ~ student_t(nu, mu, sigma);
}
generated quantities {
  vector[N] y_rep;
  for (n in 1:N) {
    y_rep[n] = student_t_rng(nu, mu, sigma);
  }
}
"
writeLines(modelString, con = "code/newcomb2.stan")
```

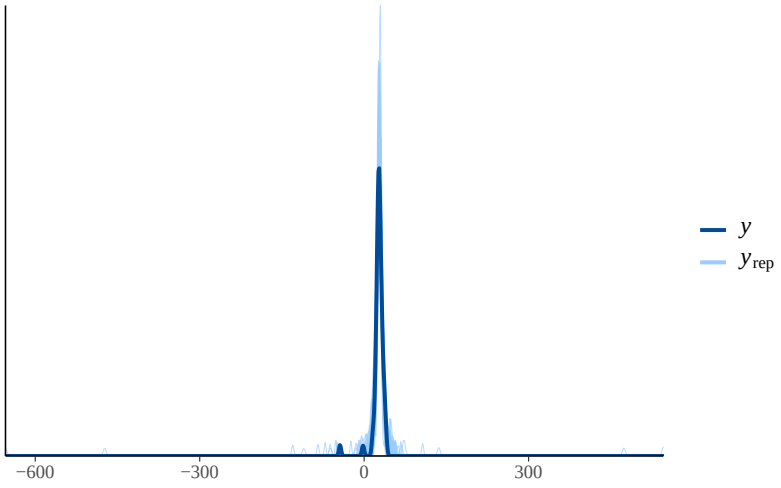
Adattiamo questo secondo modello ai dati.

```
file <- file.path("code", "newcomb2.stan")
mod <- cmdstan_model(file)
fit <- mod$sample(
  data = data_list,
  iter_sampling = 4000L,
```

```
  iter_warmup = 2000L,
  seed = SEED,
  chains = 4L,
  cores = 4L,
  parallel_chains = 2L,
  refresh = 0,
  thin = 1
)
#> Running MCMC with 4 parallel chains...
#>
#> 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: 0.4 seconds.
```

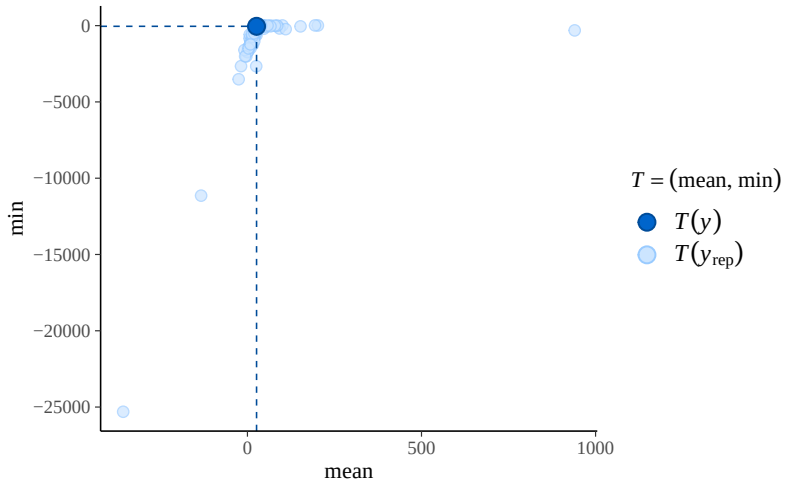
Per questo secondo modello il confronto tra la funzione di densità empirica della y e quella di diversi campioni y^{rep} risulta adeguato:

```
stanfit <- rstan::read_stan_csv(fit$output_files())
y_rep <- as.matrix(stanfit, pars = "y_rep")
ppc_dens_overlay(data_list$y, y_rep[1:50, ])
```



Inoltre, anche la statistica "minimo della distribuzione" viene ben predetta dal modello.

```
ppc_stat_2d(data_list$y, y_rep, stat = c("mean", "min"))
```



In conclusione, per le misurazioni della velocità della luce di Newcomb l'accuratezza predittiva del modello basato sulla distribuzione t di Student è chiaramente migliore di quella del modello normale.

Considerazioni conclusive

Questo capitolo presenta i controlli predittivi a posteriori. A questo proposito è necessario notare un punto importante: i controlli predittivi a posteriori, quando suggeriscono un buon adattamento del modello alle caratterische dei dati previsti futuri y^{rep}, non forniscono necessariamente una forte evidenza della capacità del modello di generalizzarsi a nuovi campioni di dati. Una tale evidenza sulla generalizzabilità del modello può solo essere fornita da studi di *holdout validation*, ovvero da studi nei quali viene utilizzato un *nuovo* campione di dati. Se i PPC mostrano un cattivo adattamento del modello ai dati previsti futuri, però, questo controllo fornisce una forte evidenza di una errata specificazione del modello.

Bibliografia

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

Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press. (Cit. a p. 8).

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

Zetsche, U., Bürkner, P.-C., & Renneberg, B. (2019). Future expectations in clinical depression: Biased or realistic? *Journal of Abnormal Psychology*, *128*(7), 678–688 (cit. a p. 3).

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.