Tecniche di Bootstrap

Analisi Dati e Statistica, 2025–26

Paolo Bosetti
Università di Trento, Dipartimento di Ingegneria Industriale

Ultimo aggiornamento: 17/06/2026

Indice


```
options(width = 60)
set.seed(0)
library(latex2exp)
library(glue)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──── tidyverse
2.0.0 ──
✔ dplyr     1.2.1     ✔ readr     2.2.0
```

```
✔ forcats   1.0.1     ✔ stringr   1.6.0
✔ ggplot2   4.0.3     ✔ tibble    3.3.1
✔ lubridate 1.9.5     ✔ tidyr     1.3.2
✔ purrr     1.2.2
── Conflicts ─────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.
org/>) to force all conflicts to become errors
```

```
library(modelr)
theme_set(theme_gray()+theme(legend.position        =
"bottom"))
```

1 Bootstrap

To pull oneself up by one's bootstraps significa letteralmente *sollevarsi da terra tirandosi per le cinghie degli stivali*

In generale, significa costruire qualcosa da risorse apparentemente minime o inesistenti

In statistica, significa ricostruire una popolazione da un semplice campione

1.1 Tecniche di Bootstrap

Scopo: analizzare una particolare statistica $\hat{\vartheta}$ simulando nuovi campioni a partire da un campione originario per effettuare dell'inferenza sulla statistica in questione

Si punta a inferire da $\hat{\vartheta}$ il corrispondente **parametro** ϑ della distribuzione.

La simulazione può essere fatta in due modi:

- **Bootstrap non-parametrico**: i campioni bootstrap vengono generati dal campione originario mediante campionamento con reinserimento
- **Bootstrap parametrico** i campioni bootstrap vengono generati da distribuzioni aventi una forma nota (e assunta corretta) e parametri stimati dal campione originario (tipicamente media e varianza).

Un **parametro** in statistica, è una costante da cui dipende la *forma* di una distribuzione: ad esempio, la distribuzione $\mathcal{N}(\mu, \sigma^2)$ è una distribuzione a due parametri: il valore atteso μ e la varianza σ^2

1.2 Tecniche di Bootstrap

Ad esempio, supponiamo di voler *stimare* un determinato parametro θ con lo stimatore $\hat{\theta} = s(x)$, dove $s(\cdot)$ è una qualche funzione dei dati campionari

θ potrebbe essere il **valore atteso** e $\hat{\theta}$ la **media campionaria**

Si noti che $\hat{\theta}$ è a sua volta una variabile casuale. In quanto tale, $\hat{\theta}$ avrà una sua distribuzione, ed indicheremo con $G(\theta) = P(\hat{\theta} < \theta)$ la sua funzione di ripartizione (CDF^-).

La distribuzione G è chiamata **distribuzione campionaria** della statistica $\hat{\theta}$.

1.3 Tecniche di Bootstrap

La **forma di** G dipende da:

- la distribuzione originaria F
- la funzione $s(\cdot)$ utilizzata per calcolare la statistica $\hat{\theta}$
- la dimensione del campione n

In alcuni casi—cioè per alcune (poche) combinazioni di F, $s(\cdot)$ e n—la distribuzione campionaria è nota in maniera esatta. Tuttavia, nella maggior parte dei casi essa è **nota solo asintoticamente**, cioè quando $n \to +\infty$. In alcuni casi, addirittura, G può non essere nota nemmeno asintoticamente.

1.4 Esempio: media campionaria, distribuzione normale

Replichiamo 10 000 volte la media di un **campione casuale normale** di n elementi

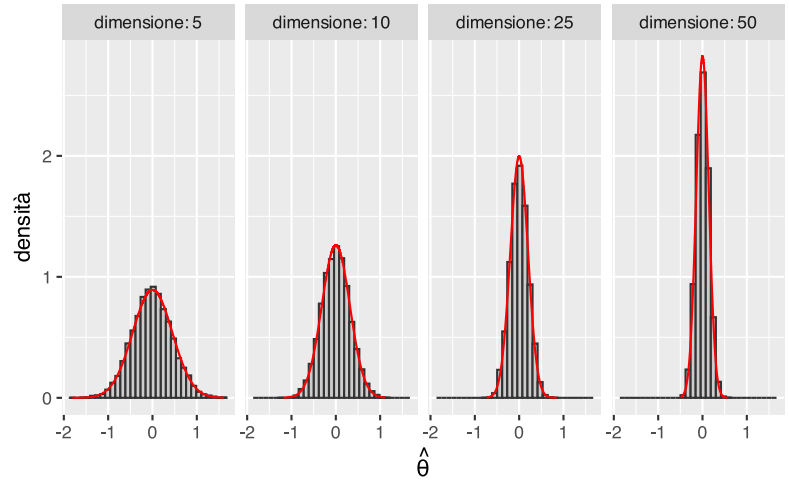
Valutiamo la distribuzione delle 10 000 stime con un istogramma

Ripetiamo il processo per dimensioni del campione $n \in \langle 5, 10, 25, 50 \rangle$

L'istogramma è pressoché perfettamente sovrapposto alla PDF di $\mathcal{N}(\theta, \sigma^2/n)$, quale che sia il valore di n

```
N<-10000
n<-c(5, 10, 25, 50)
set.seed(10)
df <- tibble(.rows=N)
for (i in n) {
  df[as.character(i)] = replicate(N, mean(rnorm(i)))
}
df <- df %>%
  pivot_longer(seq_along(n), names_to = "dimensione",
values_to = "thetahat") %>%
  mutate(
    size_n=as.numeric(dimensione),
    dimensione=factor(dimensione, levels=n)
  )
asymp.se <- function(n) 1/sqrt(n)

df %>%
  ggplot(aes(x=thetahat, y=after_stat(density))) +
      geom_histogram(bins   =   31,   fill=gray(0.8),
color=gray(0.2)) +
                      geom_line(aes(y=dnorm(thetahat,
sd=asymp.se(size_n))), color="red") +
  labs(x=TeX("\\hat{\\theta}"), y="densità") +
 facet_wrap(~dimensione, labeller=label_both, nrow=1)
```



1.5 Esempio: media campionaria, distribuzione uniforme

Se il campione proviene da una distribuzione non-normale (ad esempio uniforme) allora, grazie al **teorema del limite centrale**, la distribuzione della statistica campionaria è **asintoticamente normale**, cioè $G(\theta) \to \mathcal{N}(\theta, \sigma^2/n)$ quando $n \to +\infty$.

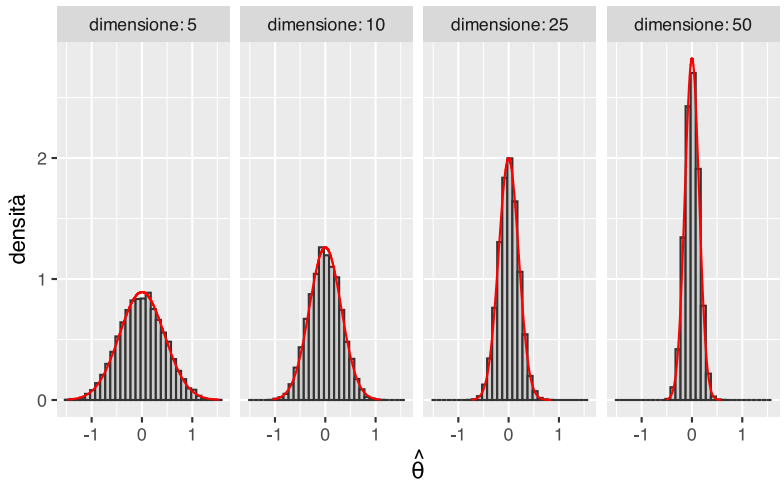
Ripetiamo l'analisi sopra fatta per il campione normale

Questa volta gli istogrammi, come previsto dal **teorema del limite centrale** diventano gradualmente più normali all'aumentare del numero di elementi di ciascun campione

```
set.seed(10)
df <- tibble(.rows=N)
for (i in n) {
   df[as.character(i)] = replicate(N, mean(runif(i,
min=-sqrt(3), max=sqrt(3))))
}
df <- df %>%
  pivot_longer(seq_along(n), names_to = "dimensione",
values_to = "thetahat") %>%
  mutate(
    size_n=as.numeric(dimensione),
    dimensione=factor(dimensione, levels=n)
  )

df %>%
  ggplot(aes(x=thetahat, y=after_stat(density))) +
```

```
      geom_histogram(bins   =   31,   fill=gray(0.8),
color=gray(0.2)) +
                    geom_line(aes(y=dnorm(thetahat,
sd=asymp.se(size_n))), color="red") +
  labs(x=TeX("\\hat{\\theta}"), y="densità") +
  facet_wrap(~dimensione, labeller=label_both, nrow=1)
```



1.6 Esempio: mediana campionaria

Consideriamo la **mediana** $\tilde{x}$ come statistica, a partire da campioni **uniformi**

Si può dimostrare analiticamente come in questo caso la distribuzione campionaria di $\hat{\theta} = \tilde{x}$ tende ad una distribuzione normale $\mathcal{N}(\theta, 1/(4nf(\theta)^2))$, dove $f(\cdot)$ è la PDF della normale standard, quando $n \to +\infty$

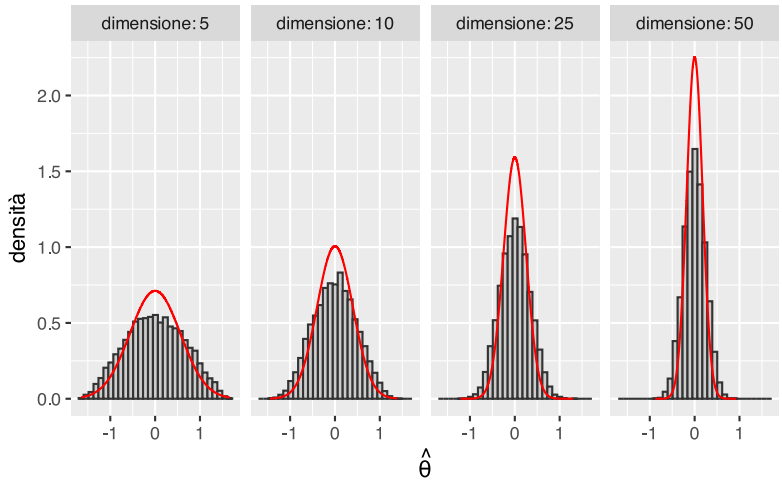
In questo caso, la convergenza è ancora più lenta della convergenza della media su un campione uniforme

La convergenza è comunque garantita dal **teorema del limite centrale**

```
set.seed(10)
df <- tibble(.rows=N)
for (i in n) {
  df[as.character(i)] = replicate(N, median(runif(i,
min=-sqrt(3), max=sqrt(3))))
}
df <- df %>%
```

```
  pivot_longer(seq_along(n), names_to = "dimensione",
values_to = "thetahat") %>%
  mutate(
    size_n=as.numeric(dimensione),
    dimensione=factor(dimensione, levels=n)
  )

# asymp.se <- function(n) 1 / sqrt(4 * n * dunif(0,
min = -sqrt(3), max = sqrt(3))^2)
asymp.se <- function(n) 1 / sqrt(4 * n * dnorm(0)^2)
df %>%
  ggplot(aes(x=thetahat, y=after_stat(density))) +
      geom_histogram(bins   =   31,   fill=gray(0.8),
color=gray(0.2)) +
                     geom_line(aes(y=dnorm(thetahat,
sd=asymp.se(size_n))), color="red") +
  labs(x=TeX("\\hat{\\theta}"), y="densità") +
 facet_wrap(~dimensione, labeller=label_both, nrow=1)
```



1.7 E Quindi?

- Effettuare inferenza su una statistica non sempre è analiticamente possibile
- È possibile generare molti campioni in modo **parametrico**, cioè assumendo una distribuzione nota sulla base della quale generare campioni casuali
- Tuttavia cambiare la distribuzione di partenza può modificare sensibilmente il risultato

- **Quindi**: sia i metodi analitici che un bootstrap parametrico presentano alcuni limiti

1.8 Boostrap Non-Parametrico

Si assume che la distribuzione del campione sia **ignota**

Si generano R campioni a partire dal campione originario mediante **campionamento con reinserimento**

Dato che tutti gli elementi hanno la stessa probabilità di essere estratti dal campione originario, ogni campione di boostrap mantiene la stessa distribuzione (**incognita**)

Se R è grande ($R > 10000$) è possibile studiare la distribuzione del parametro in studio sui campioni di bootstrap per inferirne le proprietà

In particolare sarà possibile calcolarne:

- il valore atteso
- la varianza
- l'intervallo di confidenza

1.9 Procedura Generale

Procedura

Se x è un campione di n elementi provenienti da una distribuzione ignota:

1. si ricampiona x con reinserimento, ottenendo $x_i^* = \left\langle x_{i,1}^*, x_{i,2}^*, ..., x_{1,n}^* \right\rangle$
2. si calcola la statistica $\hat{\theta}_1^* = s(x_1^*)$
3. si ripetono i primi due passi R volte, ottenendo un *campione di* $\hat{\theta}_i^*, i = 1, 2, ..., R$

La **distribuzione di bootstrap** consiste quindi di R stime di θ più la stima del campione originale $\hat{\theta}$, cioè si ha il campione di $R + 1$ stime $\left\langle \hat{\theta}, \hat{\theta}_1^*, \hat{\theta}_2^*, ...,, \hat{\theta}_R^* \right\rangle$.

Si possono ora stimare le proprietà del parametro θ sulla base del campione di bootstrap:

- il valore atteso di θ è stimata con la media del campione di bootstrap
- la varianza di θ è stimata con la varianza del campione di bootstrap
- l'intervallo di confidenza di θ è stimato dai quantili del campione di boostrap

1.10 Esempio

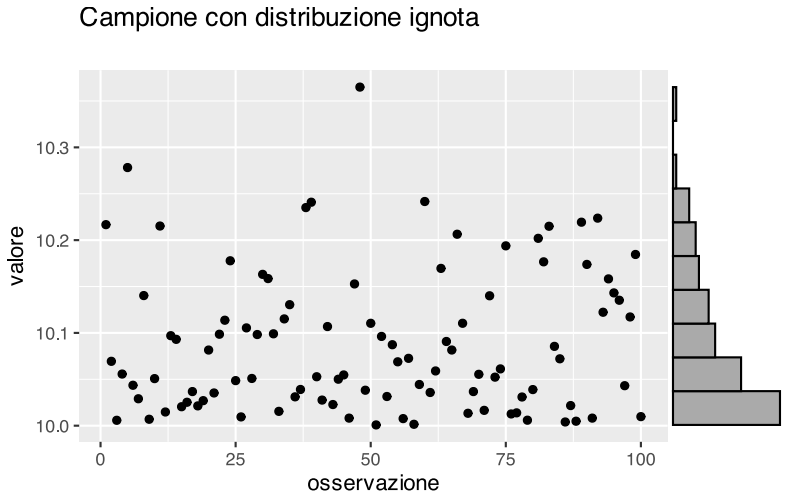
```
N <- 50000
n <- 100
```

Consideriamo un campione di $n = 100$ elementi provenienti da una popolazione con distribuzione ignota

Dal grafico è evidente che la **distribuzione non è normale**

Vogliamo calcolare il valore atteso della media campionaria e **il suo intervallo di confidenza**

```
library(ggExtra)
library(boot)
set.seed(1)
data <- rbeta(n, 1, 10) + 10
data.b <- boot(data, \(x, i) mean(x[i]), R=N)
p <- tibble(i=1:n, v=data) %>% ggplot(aes(x=i, y=v)) +
  geom_point() +
   labs(x="osservazione", y="valore", title="Campione
con distribuzione ignota")
ggMarginal(p, type="histogram", margins="y", bins=10,
fill=gray(2/3))
```



NOTA: se valesse l'ipotesi di normalità, l'intervallo di confidenza potrebbe essere calcolato dal T-test a un campione

1.11 Esempio

```
data.ci <- boot.ci(data.b, type=c("perc"))
data.test <- t.test(data)
```

Costruiamo un campione di boostrap ricampionando $R = 5 \times 10^4$ volte il campione originario, calcolando $R + 1$ volte la media campionaria

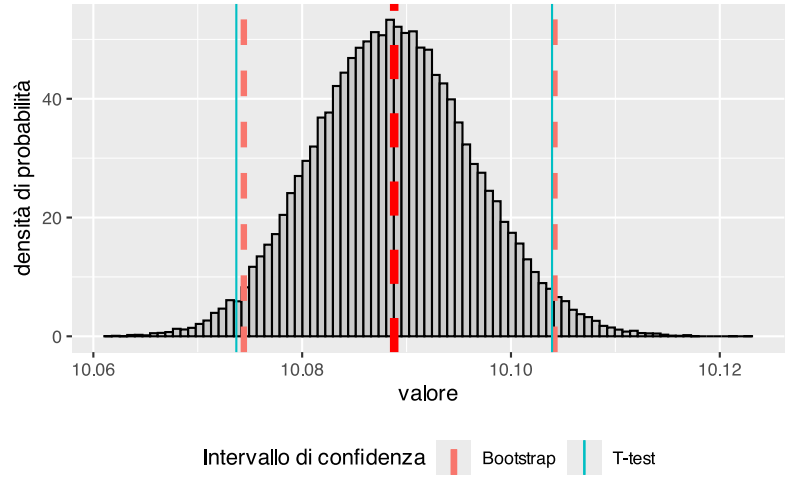
Riportiamo in istogramma le $R + 1$ stime della media

Il **valore atteso della media** è $E(\bar{x}) = \bar{\hat{\theta}} = 10.0888$

L'**intervallo di confidenza per la media** al 95% è calcolato dai quantili empirici di $\hat{\theta}$:

$$L = Q^{-}(\hat{\theta}, (1 - 0.95)/2), \;\; U = Q^{+}(\hat{\theta}, (1 - 0.95)/2)$$

```
tibble(t=data.b$t) %>%
  ggplot(aes(x = t, y=after_stat(density))) +
  geom_histogram(bins = nclass.scott(data.b$t),
                 fill = grey(0.8),
                 color = grey(0.)) +
  geom_vline(aes(color = "Bootstrap",
                 xintercept = data.ci$percent[4]),
                 linetype = 2, linewidth = 1.5) +
  geom_vline(aes(color = "Bootstrap",
                 xintercept = data.ci$percent[5]),
                 linetype = 2, linewidth = 1.5) +
  geom_vline(aes(color = "T-test",
                xintercept = data.test$conf.int[1])) +
  geom_vline(aes(color = "T-test",
                xintercept = data.test$conf.int[2])) +
 geom_vline(xintercept=mean(data.b$data), linetype=2,
color="red", linewidth=2) +
     labs(x="valore",  y="densità  di  probabilità",
color="Intervallo di confidenza") +
  theme(legend.position = "bottom")
```



NOTA: confrontando con l'intervallo calcolato dal T-test si nota che il limite inferiore è diverso: ciò è dovuto all'**assimmetria della distribuzione** del campione di base

1.12 Vantaggi della Tecnica Bootstrap

Il primo vantaggio della tecnica bootstrap rispetto alle tecniche analitiche lo abbiamo visto nell'esempio precedente:

- consente di effettuare inferenza su una statistica **senza alcuna assunzione sulla distribuzione dei dati**

È anche evidente, tuttavia, che per come è costruita

- la tecnica boostrap può essere applicata a **qualsiasi statistica**, incluse statistiche costruite per via numerica o comunque non esprimibili analiticamente

L'unico svantaggio delle tecniche di bootstrap è che sono tecniche esclusivamente computazionali e sono quindi più "costose"

1.13 Esempio

Supponiamo di voler calcolare gli intervalli di confidenza su una statistica calcolata per via numerica: la **regressione non-lineare** mediante il metodo dei minimi quadrati

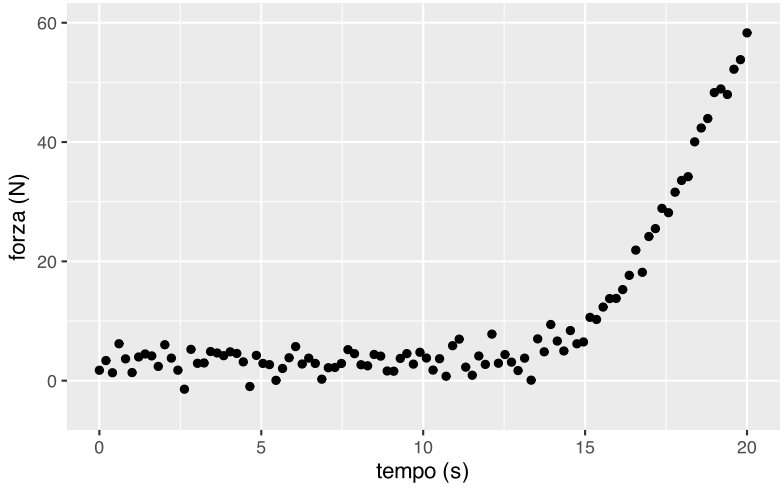
Il caso di interesse è la misura dell'istante in cui avviene il contatto tra due corpi, identificato mediante una misura di forza (rumorosa)

Il modello da regredire è:

$$f = \begin{cases} f_0 & t < t_0 \\ at^2 + bt + ct \geq t_0 \end{cases}$$

```r
f <- function(t, t0 = 0, bias = 0, a = 1) {
  b <- -2*a*t0
  c <- bias + a*t0^2
  y <- a*t^2+b*t+c
  return(ifelse(t<t0, bias, y))
}
set.seed(1)
onset <- 12.5
bias <- 3
a <- 1
data <- tibble(
  t = seq(0, 20, length.out=100),
  yn = f(t, onset, bias, 1),
  y = yn + rnorm(length(t), 0, 2)
)
fit <- nls(y~f(t, t0, bias, a), data=data,
start=list(t0=0, bias=0, a=1))
data <- add_predictions(data, fit)

data %>% ggplot(aes(x=t, y=y)) +
  geom_point() +
  labs(x="tempo (s)", y="forza (N)") +
  coord_cartesian(ylim=c(-5, 60))
```



Dove f è la forza, f_0 è il livello della forza prima del contatto, t_0 è l'istante del contatto, e b, c parametri di forma della curva di contatto

1.14 Esempio

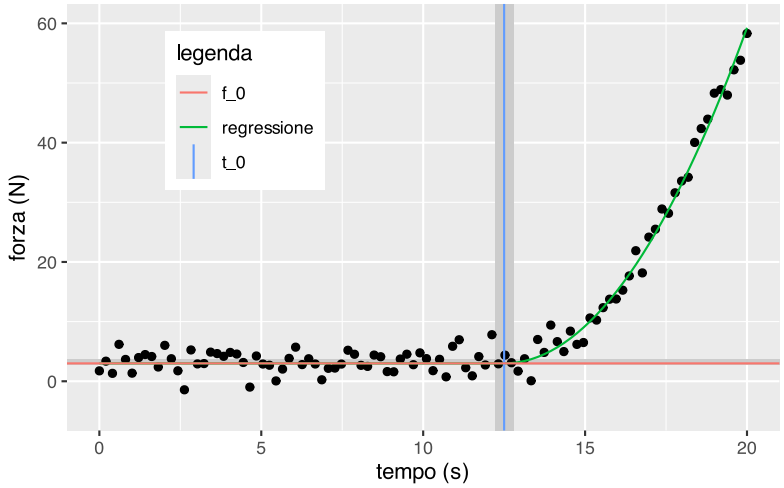
Siccome $f(\cdot)$ deve essere continua con derivata continua, risulta $2at_0 + b = 0$ e $at_0^2 + bt_0 + c = f_0$ e quindi si hanno **tre parametri**:

$$f = \begin{cases} f_0 & t < t_0 \\ at^2 - (2at_0)t + at_0^2 + f_0 t \geq t_0 \end{cases}$$

Mediante regressione ai minimi quadrati e successivo **bootstrap** è possibile identificare l'**istante di contatto** e il relativo intervallo di confidenza per un assegnato livello di confidenza (in questo caso 95%)

```
stats <- function(y, data) {
    fit  <-  nls(y~f(t,  t0,  bias,  a),  data=data,
start=list(t0=0, bias=0, a=1))
  pars <- fit$m$getPars()
  return(pars)
}
data.b <- boot(data, \(x, i) stats(x[i,"y"],x[i,]),
R=10000)
ci <- list(
         t0    =    boot.ci(data.b,    type="perc",
index=1)$percent[4:5],
        bias    =    boot.ci(data.b,    type="perc",
index=2)$percent[4:5],
         c    =    boot.ci(data.b,    type="perc",
index=3)$percent[4:5]
)
```

```
data %>% ggplot(aes(x=t, y=y)) +
  geom_rect(aes(xmin=ci$t0[1], xmax=ci$t0[2],
                ymin=-Inf, ymax=Inf), fill=gray(0.8)) +
  geom_rect(aes(ymin=ci$bias[1], ymax=ci$bias[2],
                xmin=-Inf, xmax=Inf), fill=gray(0.8)) +
  geom_point() +
  geom_line(aes(y=yn, color="regressione")) +
  geom_vline(aes(xintercept=onset, color="t_0")) +
  geom_hline(aes(yintercept=bias, color="f_0")) +
  coord_cartesian(ylim=c(-5, 60)) +
 labs(x="tempo (s)", y="forza (N)", color="legenda") +
  theme(legend.position = c(1/4, 3/4))
```



1.15 Esempio

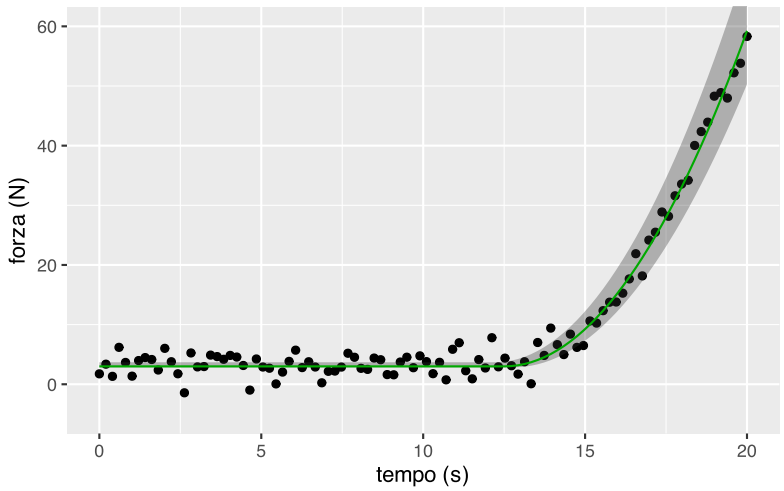
È anche possibile calcolare, per ogni valore del predittore t, gli estremi di f considerando tutte le possibili combinazioni dei tre parametri f_0, t_0, a

In questo modo è possibile identificare una **banda di confidenza** sulla funzione interpolante (per un assegnato livello di confidenza)

Ricordare che un livello di confidenza del 95% corrisponde a un limite sulla **probabilità di errore di tipo I** del test associato pari a $100\% - 95\% = 5\%$

```
f_conf <- function(t, f, ci, upper=T) {
  df <- expand.grid(ci)
  df$f <- f(t, df$t0, df$bias, df$c)
  return(ifelse(upper, max(df$f), min(df$f)))
}
data %>% mutate(
  upper = map_dbl(t, ~f_conf(., f, ci)),
  lower = map_dbl(t, ~f_conf(., f, ci, upper=F))
) %>%
  ggplot(aes(x=t, y=y)) +
  geom_point() +
  geom_ribbon(aes(ymin=lower, ymax=upper), alpha=1/3)
+
  geom_line(aes(y=yn), color=rgb(0, 2/3, 0)) +
  coord_cartesian(ylim=c(-5, 60)) +
  labs(x="tempo (s)", y="forza (N)", color="")
```

```
Ignoring unknown labels:
• colour : ""
```


60
40
20
0
forza (N)
0
5
10
15
20
tempo (s)