Regressione

Analisi Dati e Statistica, 2025–26

Paolo Bosetti
Università di Trento, Dipartimento di Ingegneria Industriale

Ultimo aggiornamento: 17/06/2026

Indice

1 Regressione

Ogni **modello** di un sistema o processo fisico dipende da **parametri**. Questi parametri devono essere calcolati **adattando** il modello alle osservazioni sperimentali (misure)

L'operazione di adattamento (*fit*) di un modello è effettuata mediante **regressione**

1.1 Basi

- Un modello di un sistema fisico può essere espresso come:

  $$y = f(x_1, x_2, ..., x_n, c_1, c_2, ..., c_m)$$

  dove x_i sono le variabili (aleatorie) fisiche, dette **regressori** o **predittori**, mentre le c_i sono i **parametri** (costanti) del modello
- Se $n = 1$ c'è un unico predittore e il modello si dice **semplice**
- Ad esempio, $y = a + bx + cx^2$ è un modello semplice con parametri a, b, c
- **regredire il modello** significa effettuare delle misurazioni di y per diversi valori di x e determinare i valori dei parametri a, b, c che **minimizzano la distanza** tra il modello e le osservazioni sperimentali

1.2 Tipi di regressione

Prenderemo in considerazione tre tipi di regressione:

- Regressione lineare: il modello è una **combinazione lineare dei parametri**
- Regressione lineare generalizzata
- Regressione ai minimi quadrati: i parametri sono combinati in modo non lineare

2 Regressione Lineare

Valida per ogni modello **lineare nei parametri** (mentre i predittori possono comparire con grado diverso da 1)

2.1 Basi

Definizioni

Si definisce il **modello statistico** del processo da regredire come segue:

$$y_i = f(x_{1i}, x_{2i}, ..., x_{ni}, c_1, c_2, ..., c_{n+1}) + \varepsilon_i = \hat{y_i} + \varepsilon_i, i = 1, 2, ..., N$$

dove i è l'indice di osservazione ($N \geq n+1$ in totale), $\hat{y}_i$ è il **valore regredito**, o **predizione**, in corrispondenza dell'osservazione i, e ε_i sono i **residui**

- Il valore regredito corrisponde alla **componente deterministica**
- Il residuo è la **componente aleatoria**
- Si assume l'ipotesi di **normalità dei residui**, cioè che $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$

Se la $f(\cdot)$ è una funzione analitica e lineare nei coefficienti, allora possiamo esprimerla come $\mathbf{Ak} = \mathbf{y}$, dove

- $\mathbf{y}$ è il vettore delle y_i, $i = 1...N$
- $\mathbf{k}$ è il vettore dei parametri c_j, $j = 1...n+1$
- $\mathbf{A}$ una matrice $N \times (n+1)$ così composta:

$$\mathbf{A} = \begin{bmatrix} x_1^{n-1} & x_1^{n-2} & ... & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & ... & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^{n-1} & x_N^{n-2} & ... & x_N & 1 \\ & & & & \end{bmatrix}$$

2.2 Basi

L'equazione lineare può essere risolta con il metodo della **pseudo-inversa**:

$$\begin{aligned} \mathbf{A}^T\mathbf{A} \cdot \mathbf{k} &= \mathbf{A}^T \cdot \mathbf{y} \\ (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A} \cdot \mathbf{k} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \cdot \mathbf{y} \\ \mathbf{k} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \cdot \mathbf{y} \end{aligned}$$

Questa relazione rende evidente cosa si intende per **regressione lineare**: non ha nulla a che fare con il **grado della funzione regredita**, ma solo con l'equazione lineare nei parametri che rappresenta il modello

NOTE:

- dalla equazione matriciale precedente è evidente che la regressione può essere eseguita se e solo se $N \geq n+1$, cioè se il numero di osservazioni è almeno pari al numero di parametri
- anche nel caso di una $f(\cdot)$ con più predittori, se essa è lineare nei coefficienti è sempre possibile esprimerla come $\mathbf{Ak} = \mathbf{y}$ e quindi risolverla con il metodo della pseudo-inversa

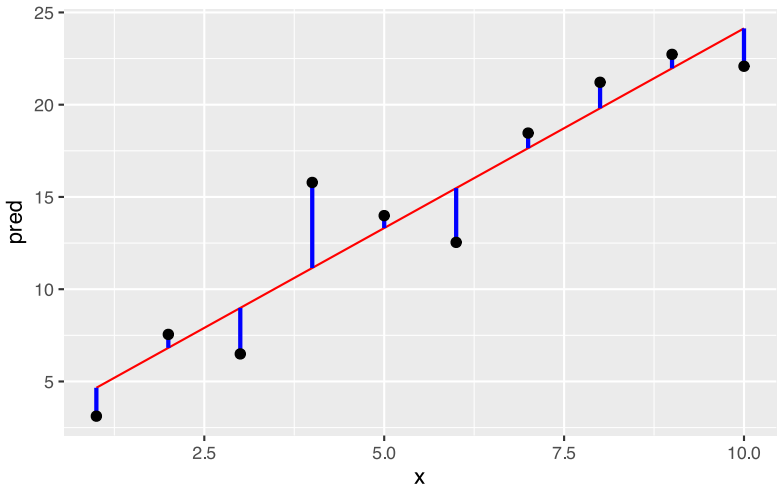
2.3 Esempio

Sia il modello da regredire del tipo $y_i = (ax_i + b) + \varepsilon_i$; allora può essere rappresentato come:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

```
library(modelr)
set.seed(1)
df <- tibble(x=1:10, y=3+2*x+rnorm(length(x), sd=3))
df.lm <- lm(y~x, data=df)
df <- add_predictions(df, df.lm)
df <- add_residuals(df, df.lm)
ggplot(df, aes(x=x, y=y)) +
  geom_line(aes(y=pred), color="red") +
    geom_linerange(aes(ymin=pred,  ymax=pred+resid),
color="blue", size=1) +
  geom_point(size=2)
```

```
Warning: Using `size` aesthetic for lines was
deprecated in ggplot2
3.4.0.
ℹ Please use `linewidth` instead.
```



La figura mostra le osservazioni come punti di coordinate (x_i, y_i), il modello regredito come una **retta rossa** (modello lineare nei

coefficienti a, b e di primo grado del predittore x) e i residui ε_i come segmenti blu che rappresentano la differenza tra le y_i e il corrispondente valore regredito $\hat{y}_i$

3 Regressione ai Minimi Quadrati

Se il modello non è lineare nei parametri, è comunque possibile effettuarne la regressione con il metodo dei **minimi quadrati**

3.1 Regressione ai Minimi Quadrati

Si cerca cioè l'insieme di valori dei parametri che **minimizza la distanza** tra il modello e le osservazioni. Questa minimizzazione può essere realizzata definendo un **indice di merito** che rappresenta la distanza tra modello e osservazioni **in funzione dei parametri**:

$$\Phi(c_1, c_2, ..., c_m) = \sum_{i=1}^{N} \left(y_i - f(x_{1i}, x_{2i}, ..., x_{ni}, c_1, c_2, ..., c_m)\right)^2$$

- Se la $f(\cdot)$ è analitica e differenziabile, allora possiamo **minimizzare** $\Phi(\cdot)$ per derivazione, cioè risolvendo il sistema di m equazioni

$$\frac{\partial \Phi}{\partial c_i}(c_1, c_2, ..., c_m) = 0$$

- Se la $f(\cdot)$ **non è differenziabile**, il minimo di $\Phi(\cdot)$ può comunque essere calcolato per via numerica (ad es. metodo di Newton-Raphson)

4 Qualità della regressione

Ad un dato insieme di osservazioni è possibile adattare **un numero infinito di modelli**

È possibile definire alcuni **parametri di merito** e alcuni **metodi di verifica** che consentono di valutare la qualità di una regressione e, quindi, individuare il modello che meglio si adatta alle osservazioni

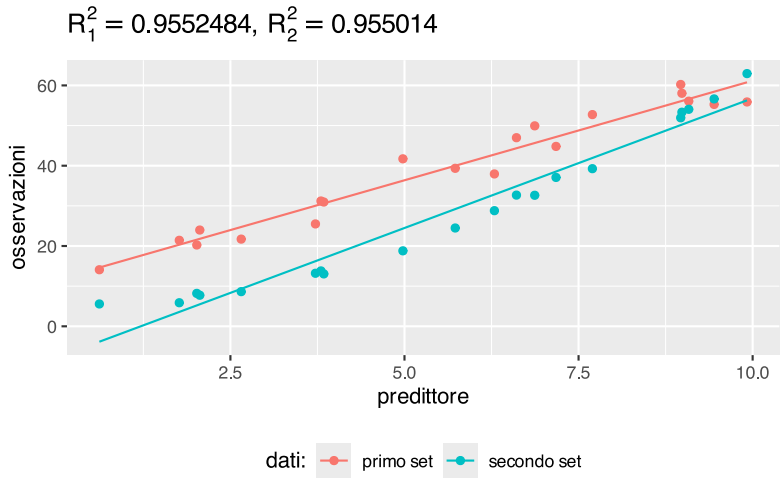
4.1 Coefficiente di Determinazione

- Il coefficiente di merito più utilizzato per valutare una regressione è il **coefficiente di determinazione** R^2
- È definito come $R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$, dove $SS_{\text{res}} = \sum \varepsilon_i^2$ e $SS_{\text{tot}} = \sum (y_i - \bar{y})^2$

- Se i valori regrediti corrispondono ai valori osservati $y_i = \hat{y}_i$, allora i residui sono tutti nulli e vale $R^2 = 1$
- La qualità della regressione diminuisce al diminuire di R^2

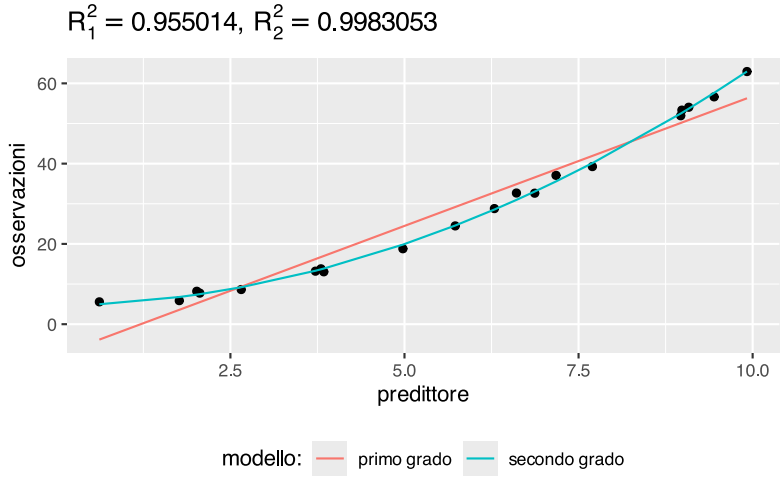
4.1.a Due set di dati

```
set.seed(0)
N <- 20
df <- tibble(
  x=runif(N, 0, 10),
  y1=5*x+12 + rnorm(N,sd=4.45),
  y2=0.5*x^2 + 1*x + 3 + rnorm(N,sd=1)
)
lm1 <- lm(y1~x, data=df)
df <- add_predictions(df, lm1, var="y1_hat")
lm2 <- lm(y2~x, data=df)
df <- add_predictions(df, lm2, var="y2_hat")
lm3 <- lm(y2~poly(x, degree=2, raw=T), data=df)
df <- add_predictions(df, lm3, var="y3_hat")
r <- c(summary(lm1)$r.squared, summary(lm2)$r.squared,
summary(lm3)$r.squared)

df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y1, color="primo set")) +
  geom_line(aes(y=y1_hat, color="primo set")) +
  geom_point(aes(y=y2, color="secondo set")) +
  geom_line(aes(y=y2_hat, color="secondo set")) +
  labs(color="dati:", title=TeX(glue("$R^2_1={r[1]}$,
$R^2_2={r[2]}$")), x="predittore", y="osservazioni") +
  theme(legend.position = "bottom")
```



4.1.b Stesso set, due modelli

```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y2)) +
  geom_line(aes(y=y2_hat, color="primo grado")) +
  geom_line(aes(y=y3_hat, color="secondo grado")) +
                              labs(color="modello:",
title=TeX(glue("$R^2_1={r[2]}$,    $R^2_2={r[3]}$")),
x="predittore", y="osservazioni") +
  theme(legend.position = "bottom")
```



Nota: come si osserva, non è necessario che i valori dei regressori siano equispaziati!

4.2 Sotto-adattamento

Si ha **sotto-adattamento** (o *under-fitting*) quando il modello ha un grado inferiore all'apparente comportamento delle osservazioni

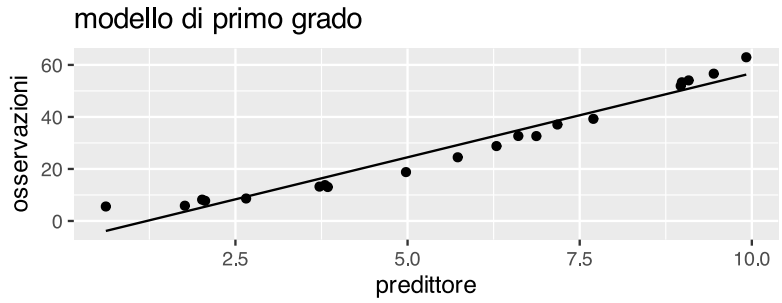
Può essere evidenziato, oltre che da un R^2 basso, studiando la distribuzione dei residui: se c'è sotto-adattamento i residui possono essere non-normali e, soprattutto, mostrare degli **andamenti**, o *pattern*

Un *pattern* è un andamento regolare dei residui in funzione dei regressori

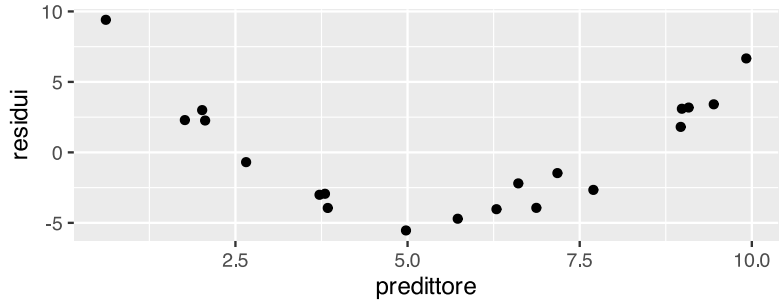
Dal numero di massimi e minimi presenti nell'eventuale *pattern* è possibile stimare **quanti gradi mancano**

4.2.a Sotto-adattamento

```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y2)) +
  geom_line(aes(y=y2_hat)) +
          labs(x="predittore",     y="osservazioni",
title="modello di primo grado")
```

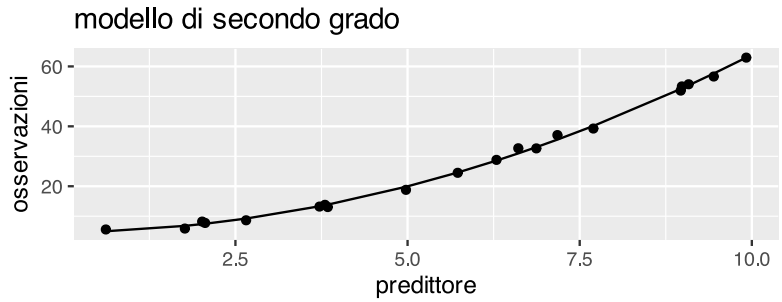



```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=lm2$residuals)) +
  labs(x="predittore", y="residui")
```

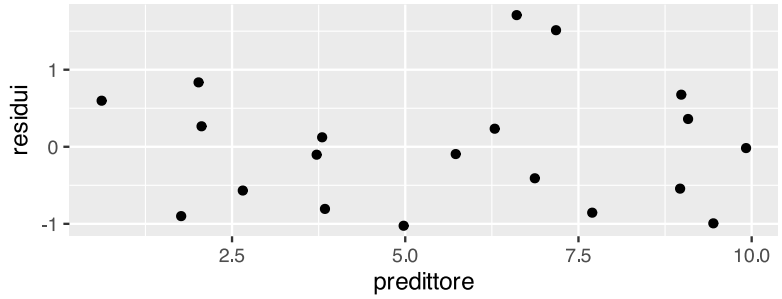


4.2.b Adattamento corretto

```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y2)) +
  geom_line(aes(y=y3_hat)) +
          labs(x="predittore",     y="osservazioni",
title="modello di secondo grado")
```




```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=lm3$residuals)) +
  labs(x="predittore", y="residui")
```



4.3 Sovra-adattamento

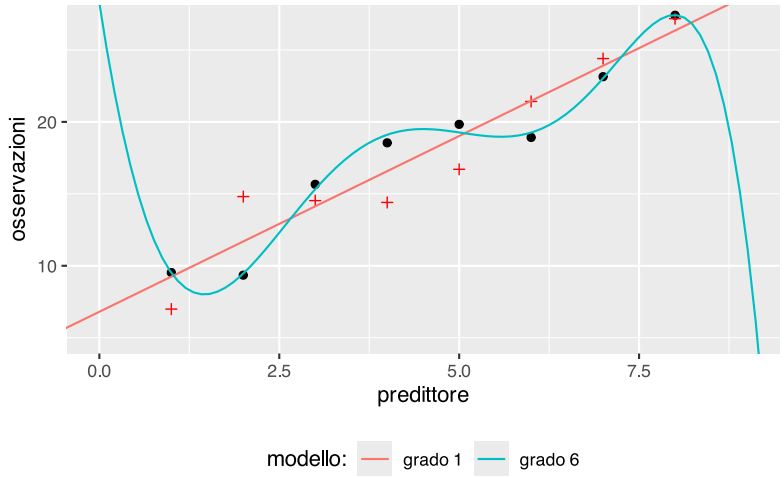
Se il grado del modello è eccessivo, il modello tende a **inseguire i singoli punti**

Il valore di R^2 cresce, raggiungendo 1 quando il grado è uguale al numero di osservazioni meno 1

Tuttavia il modello **perde di generalità** e non riesce più a predirre correttamente **nuovi valori** acquisiti in un secondo momento (le crocette rosse in figura)

Il sovra-adattamento ha effetti particolarmente drammatici in caso di **estrapolazione**, cioè quando si valuta il modello al di fuori dell'intervallo in cui è stato regredito

```r
set.seed(0)
N <- 8
df <- tibble(
  x=1:N,
  y=3*x+4 + rnorm(N, sd=2),
  y2=3*x+4 + rnorm(N, sd=2)
)
dfp <- tibble(x=seq(-1, N+2, length.out=101))
lm1 <- lm(y~x, data=df)
dfp <- add_predictions(dfp, lm1, var="pred.lm1")
lm2 <- lm(y~poly(x, deg=6, raw=T), data=df)
dfp <- add_predictions(dfp, lm2, var="pred.lm2")
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y)) +
  geom_point(aes(y=y2), color="red", shape=3) +
   geom_line(aes(x=x,  y=pred.lm1,  color="grado  1"),
data=dfp) +
   geom_line(aes(x=x,  y=pred.lm2,  color="grado  6"),
data=dfp) +
  coord_cartesian(ylim=c(5,27), xlim=c(0,9)) +
          labs(color="modello:",      x="predittore",
y="osservazioni") +
  theme(legend.position = "bottom")
```



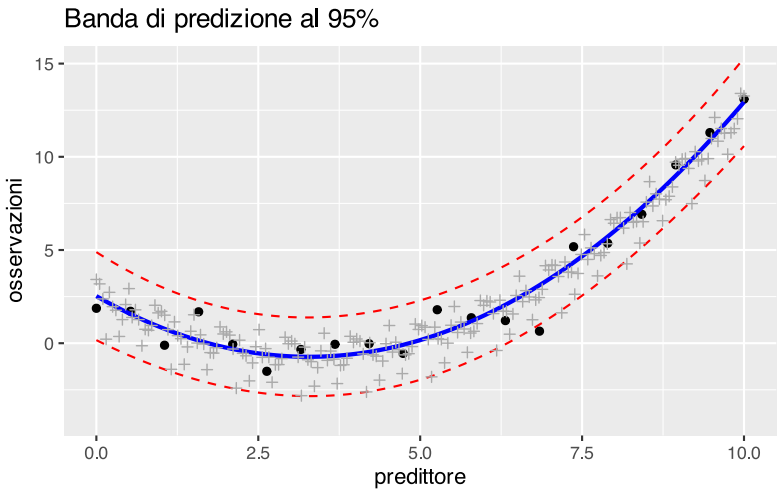
4.4 Bande di Predizione

È una banda **simmetrica rispetto alla regressione** all'interno della quale le osservazioni (presenti e future) hanno una probabilità assegnata di ricadere

Il metodo più robusto per identificare il sovra-adattamento è detto ***K-fold cross-validation***, vedi anche https://paolobosetti.quarto.pub/kfold.html

In generale, per un numero di osservazioni sufficientemente grande (> 50) la banda di predizione al 95% contiene il 95% delle osservazioni

```
set.seed(1)
f <- function(x, offset) 0.3*(x-offset)^2+1*(x-offset)
N <- 20
offset <- 5
df <- tibble(
  x=seq(0, 10, length.out=N),
  y= f(x, offset) + rnorm(N, sd=1)
)
dfp <- tibble(
  x=seq(0, 10, length.out=N*10),
  y=f(x, offset) + rnorm(N, sd=1)
)
lm <- lm(y~poly(x, deg=2, raw=T), data=df)
dfp      <-      cbind(dfp,      predict(lm,      dfp,
interval="prediction"), level=0.95)

df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y)) +
   geom_line(aes(x=x, y=fit), data=dfp, color="blue",
linewidth=1) +
   geom_line(aes(x=x, y=upr), data=dfp, color="red",
linetype=2) +
   geom_line(aes(x=x, y=lwr), data=dfp, color="red",
linetype=2) +
  geom_point(aes(x=x, y=y), data=dfp, color=gray(2/3),
shape=3) +
  coord_cartesian(ylim=c(-4,15)) +
  labs(x="predittore", y="osservazioni", title="Banda
di predizione al 95%")
```



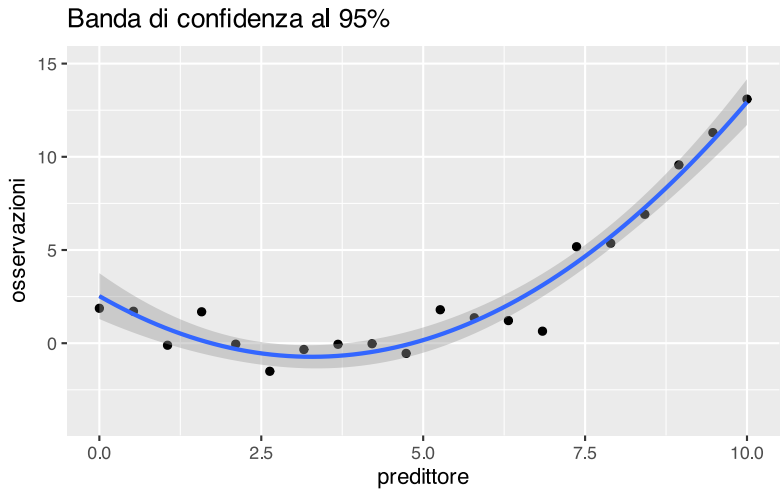
4.5 Bande di Confidenza

È una banda simmetrica rispetto alla regressione all'interno della quale **il valore atteso del modello** ha una probabilità assegnata di ricadere

È sempre più stretta rispetto alla banda di predizione

È l'equivalente multi-dimensionale dell'intervallo di confidenza per un T-test: come questo è l'intervallo all'interno del quale ha una assegnata probabilità di rientrare il valore corrispondente all'ipotesi nulla, qui possiamo assumere che il modello "*vero*" rientri con una certa probabilità nella banda di confidenza

```
df %>% ggplot(aes(x=x)) +
  geom_point(aes(y=y)) +
  geom_line(aes(x=x, y=fit), data=dfp) +
             geom_smooth(aes(y=y),       method="lm",
formula=lm$call$formula, level=0.95) +
  coord_cartesian(ylim=c(-4,15)) +
  labs(x="predittore", y="osservazioni", title="Banda
di confidenza al 95%")
```



È ottenuto calcolando gli intervalli di confidenza sui parametri della regressione, calcolando poi—per ogni valore del predittore—il valore massimo e minimo della regressione corrispondente ai valori estremi dei parametri nei loro intervalli di confidenza

5 Regressione Lineare Generalizzata

La regressione lineare classica assume l'ipotesi di **normalità dei residui**

Quando quest'ipotesi non è vera, ma il **modello è comunque lineare nei parametri**, si può utilizzare la regressione lineare generalizzata

5.1 Basi

Nel caso della regressione lineare:

$$y_i = f(\mathbf{x}_i, \mathbf{k}) + \varepsilon_i = \hat{y}_i + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Nel caso della **regressione lineare generalizzata**:

$$y_i = \hat{y}_i + \varepsilon_i$$
$$\varepsilon_i \sim D(p_1, p_2, ..., p_k)$$

dove D è una generica distribuzione a k parametri facente parte della **famiglia delle distribuzioni esponenziali** (normale, binomiale, gamma, normale inversa, Poisson, quasinormale, quasibinomiale e quasipoissoniana)

Il problema può essere risolto con l'introduzione di una **funzione di collegamento** che riscala i residui proiettandoli su una ditribuzione normale

5.2 Basi

La **funzione di collegamento** (*link function*) $g(\cdot)$ è tale per cui:

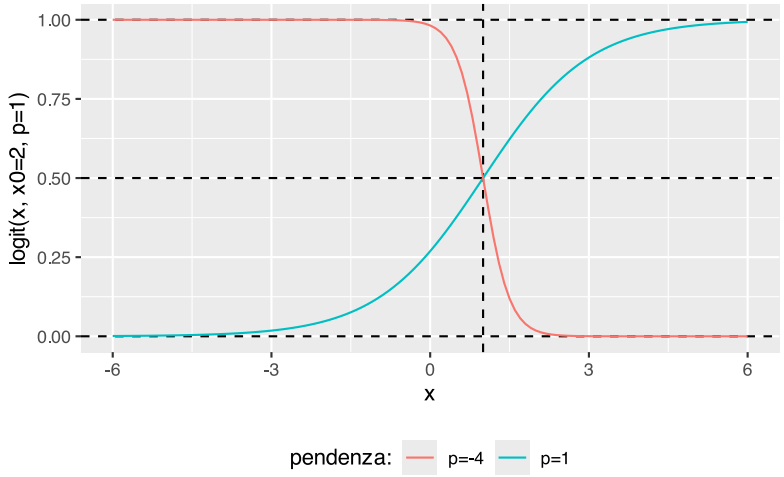
$$y_i = \hat{y}_i + g(\varepsilon_i)$$
$$\varepsilon_i \sim D(p_1, p_2, ..., p_k); g(\varepsilon_i) \sim \mathcal{N}(0, \sigma^2)$$

Le funzioni di collegamento per le distribuzioni più comuni sono:

Distribuzione	Funzione di collegamento
Normale	$g(x) = x$
Binomiale	$g(x) = \text{logit}(x)$
Poisson	$g(x) = \log(x)$
Gamma	$g(x) = 1/x$

In particolare, vale: $\text{logit}(x) = \frac{1}{1+e^{-p(x-x_0)}}$

```
logit <- function(x, p=1, x0=0) 1/(1+exp(-p*(x-x0)))
tibble(x=seq(-6,  6,  0.1),  y1=logit(x,  1,  1),
y2=logit(x, -4, 1)) %>%
  ggplot(aes(x=x)) +
  geom_hline(yintercept=c(0, 0.5, 1), lty=2) +
  geom_vline(xintercept=1, lty=2) +
  geom_line(aes(y=y1, color="p=1")) +
  geom_line(aes(y=y2, color="p=-4")) +
  labs(y="logit(x, x0=2, p=1)", color="pendenza:") +
  theme(legend.position = "bottom")
```



5.3 Regressione Logistica

Il caso tipico di regressione logistica è il classificatore di **eventi binomiali**

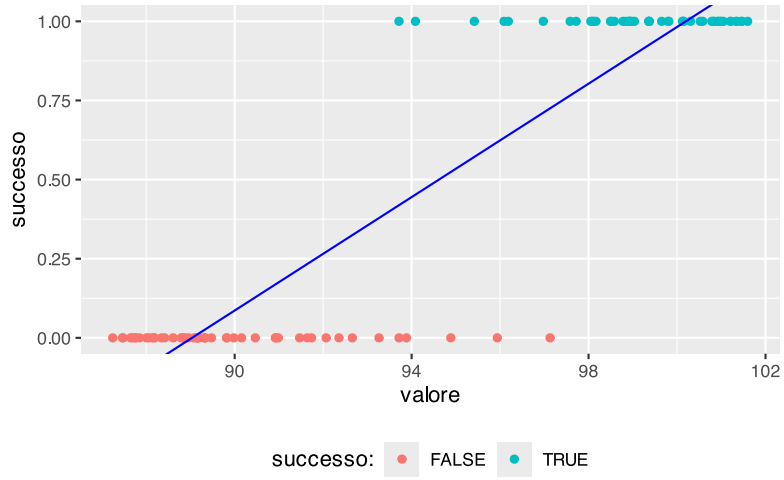
consideriamo un processo che, in funzione di uno o più predittori, possa fornire un risultato che può valere solo una di due alternative (successo|fallimento, rotto|integro, vero|falso, 1|0). Vogliamo **identificare la soglia dei predittori** che commuta il risultato

Una regressione lineare non è adatta alla situazione: è evidente che i **residui non sono normali** e che la pendenza della regressione dipende molto da quanti punti sono raccolti nelle zone "sicure"

5.3.a Dati

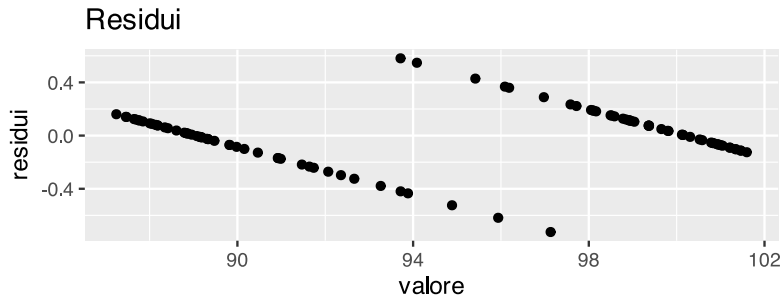
```
library(PearsonDS)
set.seed(0)
N <- 50
N2 <- 0
v0 <- c(90, 102)
sd <- 5
m1 <- list(m=90, variance=5, skewness=1.1, kurtosis=4)
m2   <-   list(m=99,   variance=5,   skewness=-1.1,
kurtosis=4)
df <- tibble(
  OK=c(rep(c(T, F), N), rep(T, N2))
) %>%
  arrange(OK) %>%
  mutate(OKv = as.numeric(OK), val=c(
    rpearson(N, moments=m1), rpearson(N, moments=m2),
```

```
runif(N2, 100, 120)))

df.lm <- lm(OKv~val, data=df)
df <- add_predictions(df, df.lm, var="pred.lm")
df <- add_residuals(df, df.lm, var="res.lm")

df %>% ggplot(aes(x=val,y=OKv)) +
  geom_point(aes(color=OK)) +
  geom_line(aes(y=pred.lm), color="blue") +
  coord_cartesian(ylim=c(0,1)) +
  labs(x="valore", y="successo", color="successo:") +
  theme(legend.position = "bottom")
```

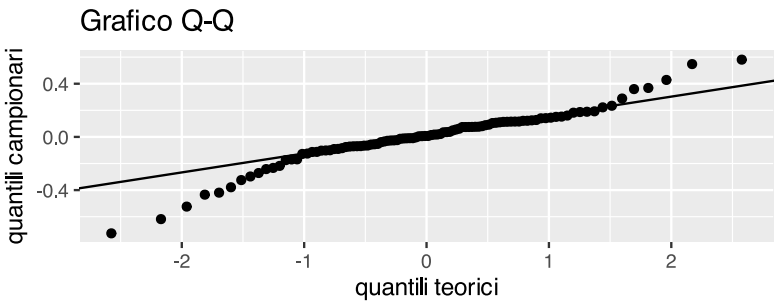


5.3.b Residui modello lineare

```
df %>% ggplot(aes(x=val, y=res.lm)) +
  geom_point() +
  labs(x="valore", y="residui", title="Residui")
```



```
df %>% ggplot(aes(sample=res.lm)) +
  geom_qq() +
  geom_qq_line() +
  labs(x="quantili teorici", y="quantili campionari",
title="Grafico Q-Q")
```



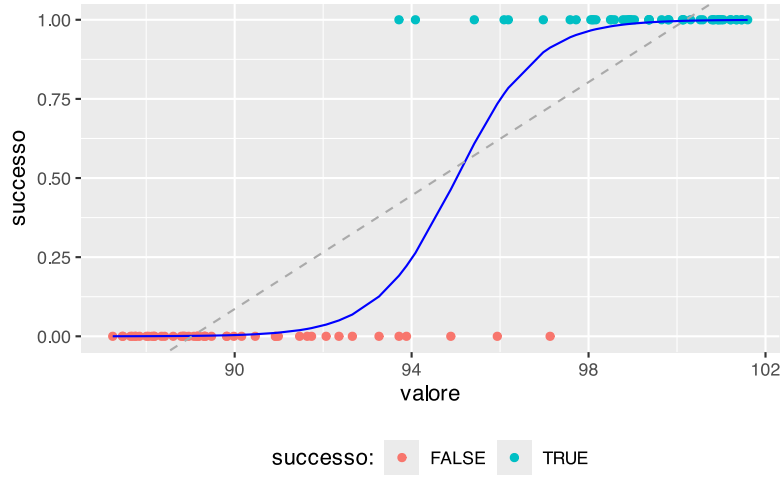
5.4 Regressione Logistica

La funzione logistica regredita fornisce il parametro x_0 che identifica il valore che separa una **uguale quantità di falsi positivi e falsi negativi**

Inoltre, è possibile individuare la soglia opportuna per ottenere una prefissata probabilità di falsi positivi (o falsi negativi)

Questo è il tipo più semplice di **machine learning**: un classificatore binomiale

```
df.glm <- glm(OKv~val, family=binomial(), data=df)
df  <-  add_predictions(df,  df.glm,  var="pred.glm",
type="response")
df <- add_residuals(df, df.glm, var="res.glm")

df %>% ggplot(aes(x=val, y=OKv)) +
  geom_point(aes(color=OK)) +
  geom_line(aes(x=val, y=pred.glm), color="blue") +
        geom_line(aes(y=pred.lm),    color=gray(2/3),
linetype=2) +
  coord_cartesian(ylim=c(0,1)) +
  labs(x="valore", y="successo", color="successo:") +
  theme(legend.position = "bottom")
```



6 Presentazione dei dati

Il concetto di **banda di confidenza** è essenziale nella presentazione grafica di dati provenienti da più serie

6.1 Confronto grafico di serie

```
set.seed(0)
r <- 8
alpha <- 0.05
df.n <- tibble(
  x=seq(100, 900, 100),
  S1=c(300, 310, 318, 330, 290, 305, 358, 290, 180),
  S2=c(315, 315, 322, 340, 272, 315, 340, 285, 185),
  n=c(10, 10, 25, 10, 10, 7, 20, 10, 10)
) %>% mutate(S2=S2-15)

df <- expand.grid(
  r=1:r,
  x=df.n$x
) %>% mutate(
  S1=map_dbl(x, ~df.n[df.n$x==.,]$S1),
  S2=map_dbl(x, ~df.n[df.n$x==.,]$S2)
  ) %>%
  tibble() %>%
              pivot_longer(S1:S2,       names_to="s",
values_to="nominal", cols_vary = "slowest")

df <- df %>%
  group_by(s, x) %>%
```

```
              group_modify(~       {.x$noise=rnorm(8,
sd=df.n[df.n$x==.y$x,]$n); .x}) %>%
  mutate(value=nominal+noise)

df.s <- df %>% group_by(s, x) %>% summarise(
  n=n(),
  ci=sd(value)/sqrt(n) * qt(alpha/2, n - 1, lower.tail
= F),
  value=mean(value),
  lwr=value-ci,
  upr=value+ci
)
```

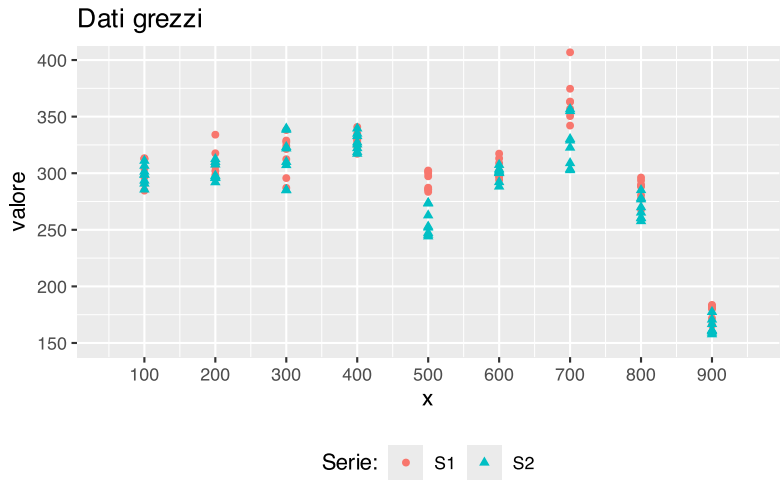
```
`summarise()` has regrouped the output.
ℹ Summaries were computed grouped by s and x.
ℹ Output is grouped by s.
ℹ Use `summarise(.groups = "drop_last")` to silence
this
  message.
ℹ Use `summarise(.by = c(s, x))` for per-operation
grouping
  (`?dplyr::dplyr_by`) instead.
```

Supponiamo di avere un processo il cui valore dipende da una variabile x

Supponiamo che un parametro S di processo possa influire sul valore in uscita. Ad esempio:

- il valore è la durezza di un metallo, x è la temperatura, il parametro S è la quantità di un elemento in lega
- il valore è la produttività di un impianto, x è un parametro quantitativo di processo, il parametro S è il tipo di macchina utilizzato

```
df %>% ggplot(aes(x=x)) +
 geom_point(aes(y=value, group=s, color=s, shape=s)) +
   labs(color="Serie:",  shape="Serie:",  y="valore",
title="Dati grezzi") +
  coord_cartesian(xlim=c(50,950), ylim=c(150,400)) +
  scale_x_continuous(breaks=df.n$x) +
  theme(legend.position = "bottom")
```



Supponiamo di ripetere 8 volte una misurazione del valore in uscita per le varie combinazioni di x e di S, ottenendo i risultati in figura: **quali differenze sono significative**?

6.2 Senza un modello di riferimento

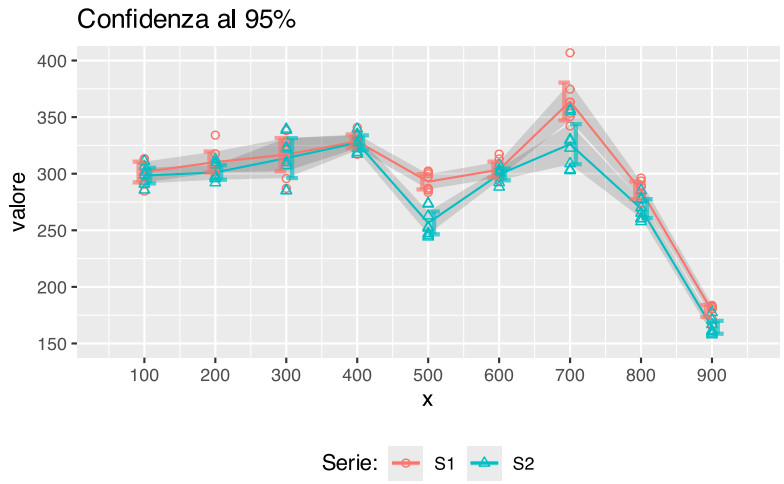
Senza un modello di riferimento che esprima $v = f(x, S)$ **non ha senso effettuare una regressione**

Tuttavia posso riportare, per ogni **trattamento**

- il valor medio, unendo le serie con una spezzata al solo scopo di raggruppare visivamente i dati
- i limiti dell'intervallo di confidenza per ogni serie e per ogni trattamento
- oppure, unire i limiti con una banda che rappresenta la **confidenza sulla media**

```
df.s %>% ggplot(aes(x=x)) +
     geom_ribbon(aes(ymin=lwr,   ymax=upr,   group=s),
alpha=1/6) +
       geom_line(aes(y=value,    group=s,    color=s),
linewidth=0.5) +
    geom_point(aes(x=x,  y=value,  group=s,  color=s,
shape=s), data=df) +
   labs(color="Serie:",  shape="Serie:",  y="valore",
title="Confidenza al 95%") +
     geom_errorbar(aes(ymin=lwr,   ymax=upr,   group=s,
color=s),   alpha=2/3,   position="dodge",   width=33,
linewidth=1) +
  coord_cartesian(xlim=c(50,950), ylim=c(150,400)) +
```

```
  scale_x_continuous(breaks=df.n$x) +
  scale_shape_manual(values=c(1, 2)) +
  theme(legend.position = "bottom")
```



Zone in cui le bande sono sovrapposte sono statisticamente indistinguibili

6.3 Con un modello

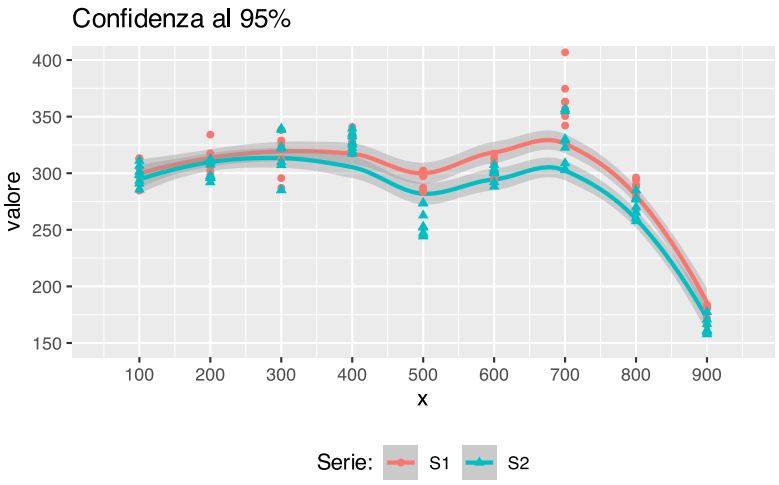
Solo se ho un modello $v = f(x, S)$ posso effettuare una regressione

Anche in questo caso, la regressione va accompagnata con bande di confidenza

Di nuovo, **zone in cui le bande sono sovrapposte sono statisticamente indistinguibili**

```
df %>% ggplot(aes(x=x)) +
  geom_smooth(aes(y=value, group=s, color=s)) +
 geom_point(aes(y=value, group=s, color=s, shape=s)) +
   labs(color="Serie:",  shape="Serie:",  y="valore",
title="Confidenza al 95%") +
  coord_cartesian(xlim=c(50,950), ylim=c(150,400)) +
  scale_x_continuous(breaks=df.n$x) +
  theme(legend.position = "bottom")
```

```
`geom_smooth()` using method = 'loess' and formula =
'y ~
x'
```


Confidenza al 95%
400
350
300
250
200
150
valore
100
200
300
400
500
600
700
800
900
x
Serie:
S1
S2