

Regressione polinomiale

La regressione polinomiale utilizza lo stesso metodo della regressione lineare, ma assume che la funzione che meglio descrive l'andamento dei dati non sia una retta, ma un polinomio. Quindi è adatta quando lo scatterplot di una relazione bivariata, ad esempio, mostra una forma diversa da quella della retta, ad esempio una curva, come nell'esempio che segue.

I dati

Tab. 1: Dati (Dataset *mtcars*)

mpg	miglia per gallone
hp	potenza

```
# per i grafici e il piping  
library(tidyverse)
```

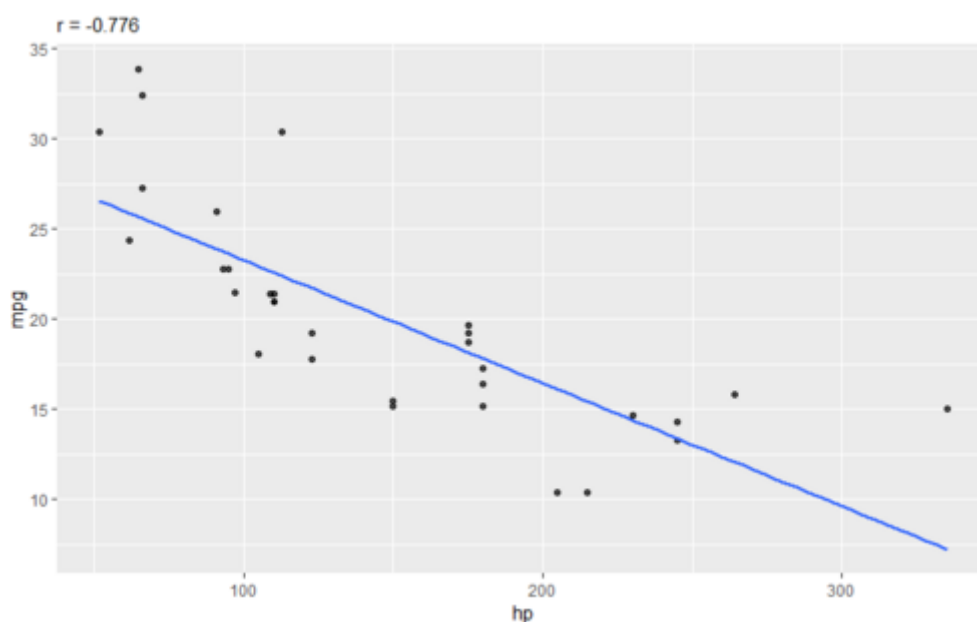


Fig. 1: Scatterplot con retta di regressione

La relazione fra le due variabili non sembra ben rappresentata da una retta (anche se potrebbe dipendere dagli outliers, visto che i punti in coda sono pochi).

La funzione

$$\hat{Y} = a + b_1X + b_2X^2$$

```
lm(mpg ~ hp + I(hp^2), data = mtcars) %>%  
summary()
```

equivale a:

```
lm(mpg ~ poly(hp, 2, raw=TRUE), data = mtcars) %>%  
summary()
```

```
## Call:  
## lm(formula = mpg ~ hp + I(hp^2), data = mtcars)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.5512 -1.6027 -0.6977  1.5509  8.7213   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  4.041e+01  2.741e+00  14.744 5.23e-15 ***  
## hp          -2.133e-01  3.488e-02  -6.115 1.16e-06 ***  
## I(hp^2)      4.208e-04  9.844e-05   4.275 0.000189 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 3.077 on 29 degrees of freedom  
## Multiple R-squared:  0.7561, Adjusted R-squared:  0.7393   
## F-statistic: 44.95 on 2 and 29 DF,  p-value: 1.301e-09
```

$\hat{\text{mpg}} = 4.041e{+01} - 2.13e{-01} \text{ hp} + 4.20e{-04} \text{ hp}^2$

Ovvero:

```
x = mtcars$hp  
fitted.y = 4.041e+01 + (-2.133e-01 * x) + (4.208e-04 * x^2)
```

Per avere polinomi di grado superiore, è sufficiente scrivere:

- `poly(hp, 3, raw=TRUE)`; oppure
- `hp + I(hp^2) + I(hp^3)`

La prima forma è quindi più sintetica e comoda da scrivere.

Regressione polinomiale ortogonale

Se non specifichiamo `raw = TRUE`, con `poly(x, n)` otteniamo una regressione ortogonale, in cui cioè i coefficienti sono indipendenti l'uno dall'altro:

```
fit1 <- lm(mpg ~ poly(hp, 2), data = mtcars)  
summary(fit1)
```

```
## Call:
## lm(formula = mpg ~ poly(hp, 2), data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.5512 -1.6027 -0.6977  1.5509  8.7213
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20.091      0.544  36.931 < 2e-16 ***
## poly(hp, 2)1   -26.046      3.077  -8.464 2.51e-09 ***
## poly(hp, 2)2    13.155      3.077   4.275 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.077 on 29 degrees of freedom
## Multiple R-squared:  0.7561, Adjusted R-squared:  0.7393
## F-statistic: 44.95 on 2 and 29 DF,  p-value: 1.301e-09
```

La rappresentazione grafica con ggplot

Vedi:

- [Basi dei grafici con ggplot](#);
- il grafico alla pagina [Regressione lineare bivariata](#)

Per rappresentare la curva di interpolazione:

```
ggplot(mtcars, aes(hp, mpg)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = F) +                # retta dei
minimi quadrati
  geom_smooth(method = "lm",
              formula = y ~ poly(x, 2), col = "red")  # geom_smooth
con la formula
```

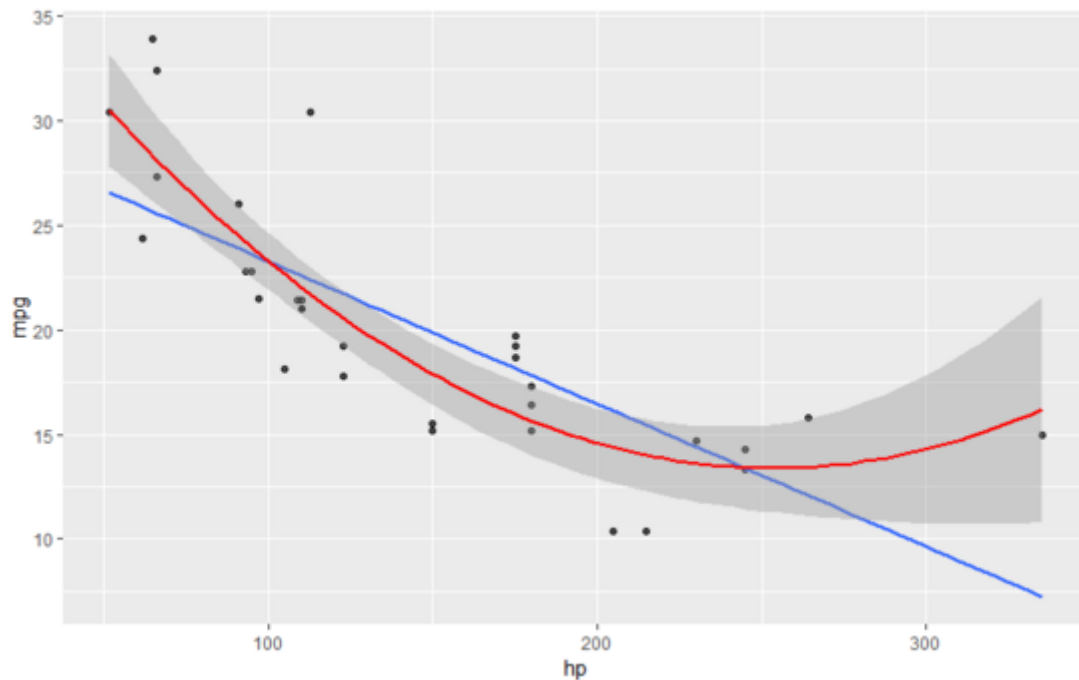


Fig. 2: Scatterplot con retta di regressione (blu), curva di regressione (rossa) e area di errore

Script di esempio

E' possibile scaricare ed eseguire lo script dell'esempio:

[Es-regr-poli.R](#)

```
library(tidyverse)

# con I()
lm(mpg ~ hp + I(hp^2), data = mtcars) %>%
summary()

# con poly()
lm(mpg ~ poly(hp, 2, raw=TRUE), data = mtcars) %>%
summary()

# ortogonale, salvando il modello 'fit'
fit <- lm(mpg ~ poly(hp, 2), data = mtcars)
summary(fit)

# grafico della curva
ggplot(mtcars, aes(hp, mpg)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = F) + # retta
  # dei minimi quadrati
```

```
geom_smooth(method = "lm",  
             formula = y ~ poly(x,2), col = "red") #  
geom_smooth con la formula
```

Analisi bivariata, Regressione lineare, Modelli lineari

From:

<https://www.agnesevardanega.eu/wiki/> - Ricerca Sociale con R

Permanent link:

https://www.agnesevardanega.eu/wiki/r/modelli/regressione_polinomiale?rev=1754923148

Last update: **11/08/2025 14:39**

