Regressione lineare semplice

STATISTICA NUMERICA, CAP. 6.4

Regressione lineare: introduzione

La regressione lineare è il cuore della statistica.

Risponde alla domanda:

« come posso utilizzare i dati che ho misurato per fare previsoni su dati che non conosco?»

utilizzando in particolare un modello di tipo lineare.

Regressione lineare: introduzione

La *regressione lineare* è la parte della statistica che studia la relazione fra due o più variabili, che sono legate in modo NON DETERMINISTICO, per fare inferenze sul modello.

In particolare, si usano relazioni fra due o più variabili in modo da potere avere informazioni su una di esse conoscendo i valori dell'altra. Esempi di variabili che non sono legate fra loro da una relazione deterministica: x= l'età di un bambino e Y=la sua altezza, x=il volume di un motore e Y=il suo consumo di carburante, x=tempo di studio e Y= voto all'esame, ecc. Poiché x non è una variabile casuale la indichiamo con la lettera minuscola, mentre Y, che è un variabile casuale, viene indicata con la lettera maiuscola. Nel caso lineare supponiamo una relazione appunto lineare fra le due variabili x e Y:

$$Y = \beta_0 + \beta_1 x$$

Regressione lineare: introduzione

. Questa relazione, di per sè deterministica, viene generalizzata a una relazione probabilistica. Date quindi informazioni su x e Y, l'obiettivo è quello di *predire* un valore futuro di Y per un particolare valore di x.

In questo modello, x viene detta *variabile indipendente* e Y viene detta *variabile dipendente*.

Il modello viene costruito a partire da alcune osservazioni $(x_i, Y_i), i = 1, \ldots n.$.

L'estensione al modello probabilistico è necessaria nel momento in cui le due variabili non hanno una relazione deterministica. In pratica, in corrispondenza di n variabili indipendenti $x_1, x_2, \ldots x_n$ si hanno n valori $Y_1, Y_2, \ldots Y_n$, che sono legati dalla relazione:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots n$$

quindi differiscono, rispetto al modello lineare esatto, di una quantità ϵ_i.

I valori Y_i sono in generale variabili aleatorie.

Regressione lineare: introduzione

Regressione lineare: introduzione

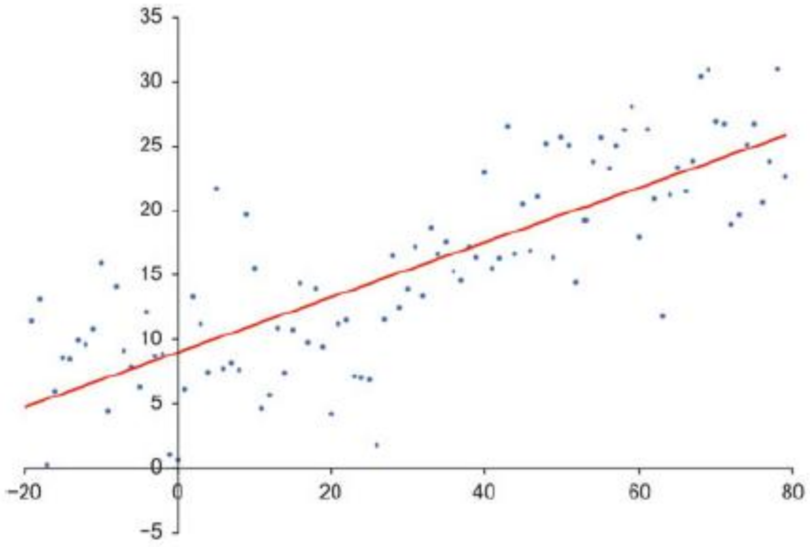


Fig. 11.2 Best-fit linear regression line to a given set of data

Regressione lineare: introduzione

Modello di regressione lineare semplice. Esistono parametri $\beta_0, \beta_1, \sigma^2$ tali che, per ogni valore fissato della variabile indipendente x, la variabile dipendente è una variabile aleatoria legata ad x dal modello:

$$Y = \beta_0 + \beta_1 x + \epsilon,$$

dove ϵ è una variabile aleatoria, detta *errore casuale*, che si assume con distribuzione `norm(0,σ)`.

Stima dei parametri :MLE

'*Come calcolare stime dei parametri β_0 e β_1 della retta di regressione lineare assegnate le coppie $(x_i, Y_i), i = 1, \ldots n$? Cioé come determinare, fra le infinite rette del piano, una* buona *retta? Esiste una retta* migliore *delle altre?*'

Visto che gli errori ϵ_i hanno distribuzione `norm(mean=0, sd=`σ`)`, allora la variabile aleatoria $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ ha distribuzione normale con deviazione standard σ. La funzione di verosimiglianza è:

$$
\begin{aligned}
L(\beta_0, \beta_1) &= \prod_{i=1}^{n} f_{Y_i}(x_i) \\
&= \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} exp \left\{ \frac{-(Y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2} \right\} \\
&= (2\pi\sigma^2)^{-n/2} exp \left\{ \frac{-\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2} \right\};
\end{aligned}
$$

Stima dei parametri: MLE

facendo il logaritmo naturale di $L(\beta_0, \beta_1)$ si ha:

$$F(\beta_0, \beta_1) = ln(L(\beta_0, \beta_1)) = -\frac{n}{2} ln(2\pi\sigma^2) - \frac{\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}.$$

Per minimizzare questa funzione rispetto alle variabili β_0 e β_1:

$$\frac{\partial F}{\partial \beta_0} = 0, \quad \frac{\partial F}{\partial \beta_1} = 0.$$

Quindi:

$$\frac{\partial F}{\partial \beta_0} = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} 2(Y_i - \beta_0 - \beta_1 x_i)(-1),$$

da cui:

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} Y_i.$$

Stima dei parametri: MLE

$$i=1 \qquad\qquad i=1$$

Per quanto riguarda l'altra derivata:

$$\begin{aligned}
\frac{\partial lnL}{\partial \beta_1} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} 2(Y_i - \beta_0 - \beta_1 x_i)(-x_i) \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i Y_i - \beta_0 x_i - \beta_1 x_i^2),
\end{aligned}$$

da cui:

$$\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i Y_i.$$

Stima dei parametri: MLE

Quindi devo risolvere il sistema costituito dalle due seguenti equazioni:

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} Y_i$$

$$\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i Y_i$$

che dà come soluzione:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i Y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} Y_i\right)/n}{\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2/n}$$

e

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x},$$

dove $\bar{Y}$ e $\bar{x}$ sono, rispettivamente, la media dei valori Y_i e x_i.

Stima dei parametri: Minimi Quadrati

Una formulazione differente ma equivalente (i risultati sono i medesimi) è quella dei Minimi Quadrati.

Principio dei minimi quadrati. Detto *residuo i-esimo* la differenza verticale fra l'osservazione i-esima e la retta di regressione lineare:

$$E_i = Y_i - (\beta_0 + \beta_1 x_i),$$

e detta $f(\beta_0, \beta_1)$ la funzione somma dei quadrati dei residui:

$$f(\beta_0, \beta_1) = \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 x_i)]^2 = \sum_{i=1}^{n} E_i^2$$

le stime $\hat{\beta}_0$ e $\hat{\beta}_1$ si ottengono minimizzando la funzione $f(\beta_0, \beta_1)$.

Significato dei parametri della retta

Significato dei parametri della retta

Se abbiamo trovato per esempio una retta che rappresenta il modello di crescita di un bambino (in centimetri) in funzione della sua eta’ espressa in mesi e l a retta ha equazione:

$$y(x)= 50+0.753\ x$$

- Il coefficiente angolare indica la pendenza della retta. In questo esempio rappresenta di quanto aumenta l’ altezza in funzione dei mesi di eta’. Per ogni mese aumenta di 0.753 centimetri.

- L’ intercetta rappresenta l’ altezza a 0 mesi, cioe’ alla nascita. In base ai dati a disposizione e’ stata stimata in 50 cm.

- Cosa significa calcolare y(5)? L’ altezza stimata dal modello a 5 mesi, cioe’ 50+0.753*5.

Inferenza sui parametri

I parametri $\hat{\beta}_0$ e $\hat{\beta}_1$ sono anch'essi variabili aleatorie e dipendono dal campione considerato. Su di essi, pertanto, si possono fare le inferenze di tipo statistico che abbiamo visto nei paragrafi precedenti, quali stime di intervalli di confidenza e test di ipotesi.

Per il test di verifica di ipotesi, il parametro più importante è sicuramente $\hat{\beta}_1$ rispetto a $\hat{\beta}_0$. Per $\hat{\beta}_1$ il test di ipotesi più frequente è quello che verifica se $\hat{\beta}_1 \neq 0$, cioè se la retta di regressione è parallela all'asse x oppure no. Una retta di regressione parallela all'asse x significa che il valore della variabile aleatoria Y NON cambia al variare della variabile indipendente x. Quindi il test di ipotesi è formulato come:

- H_0: $\beta_1 = 0$
- H_a: $\beta_1 \neq 0$

Valori predetti

Il modello serve per predire il valore della variabile aleatoria in corrispondenza di un o piu'
Valori della variabile indipendente.
I valori predetti possono essere:

- **In sample.** I valori della variabile indipendente in cui si fa la predizione sono nell' insieme dei dati a disposizione . Posso confrontare le predizioni con i valori osservati nelle stesse ascisse.

- **Out of sample**. I valori della variabile indipendente NON sono nell' insieme dei dati a disposizione. Non ho quindi nessun termine di confronto per i valori che vengono predetti.

Il coefficiente R^2

È possibile avere un *singolo numero* che mi dà indicazioni sulla bontà del modello regressione lineare semplice rispetto al campione di dati a disposizione?

Il *coefficiente semplice di determinazione* viene calcolato appunto per questo scopo. Esso è definito dalla formula:

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2},$$

dove, ricordiamo:

- $Y_i, i = 1, \ldots n$ sono i valori 'osservati' del campione;
- $\hat{Y}_i, i = 1, \ldots n$ sono i valori 'fittati', cioè i valori del modello di regressione lineare semplice in corrispondenza delle ascisse x_i ($\hat{Y}_i = \beta_0 + \beta_1 x_i, i = 1, \ldots n$);
- $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ è la media dei valori osservati.

Il coefficiente R^2

Si ha che $0 \leq r^2 \leq 1$. Tanto più r^2 è vicino a 1, tanto più il modello di regressione lineare è *buono*; tanto più r^2 è vicino a 0, tanto più il modello non è rappresentativo del campione dei dati. In quest'ultimo caso, l'analista cerca un modello differente da quello lineare per rappresentare i dati (una regressione non lineare o multivariata che coinvolga più di una variabile per esempio).

Associato al coefficiente semplice di determinazione r^2 si utilizza il *coefficiente semplice di correlazione r* che si ottiene come:

$$|r| = \sqrt{r^2}.$$

Per quanto riguarda il segno di r, si assume il segno della stima di β_1 calcolata.

Il coefficiente R^2

Il valore del coefficiente R^2 che indica un buon modello non si puo definire a priori.

Dipende dalla disciplina, di solito nelle discipline scientifiche R^2 è maggiore rispetto alle discipline sociali.
In finanza e marketing, dipende da quali dati stiamo considerando.

Attenzione! Il coefficiente R^2 da solo non ha significato se non c'è effettivamente una dipendenza lineare fra i dati.

Librerie Python

Librerie Python per fare regressione lineare semplice:

Statsmodel (Introduction — statsmodels)-

Esempio file: *example_statsmodels.py*

Esempio file: *example_statsmodels_simul.py*

Scikit-learn (sklearn.linear_model.LinearRegression — scikit-learn 1.2.2 documentation)

Esempio file: *example_sklearn.py*

Esempio file: *example_salary_sklearn.py*

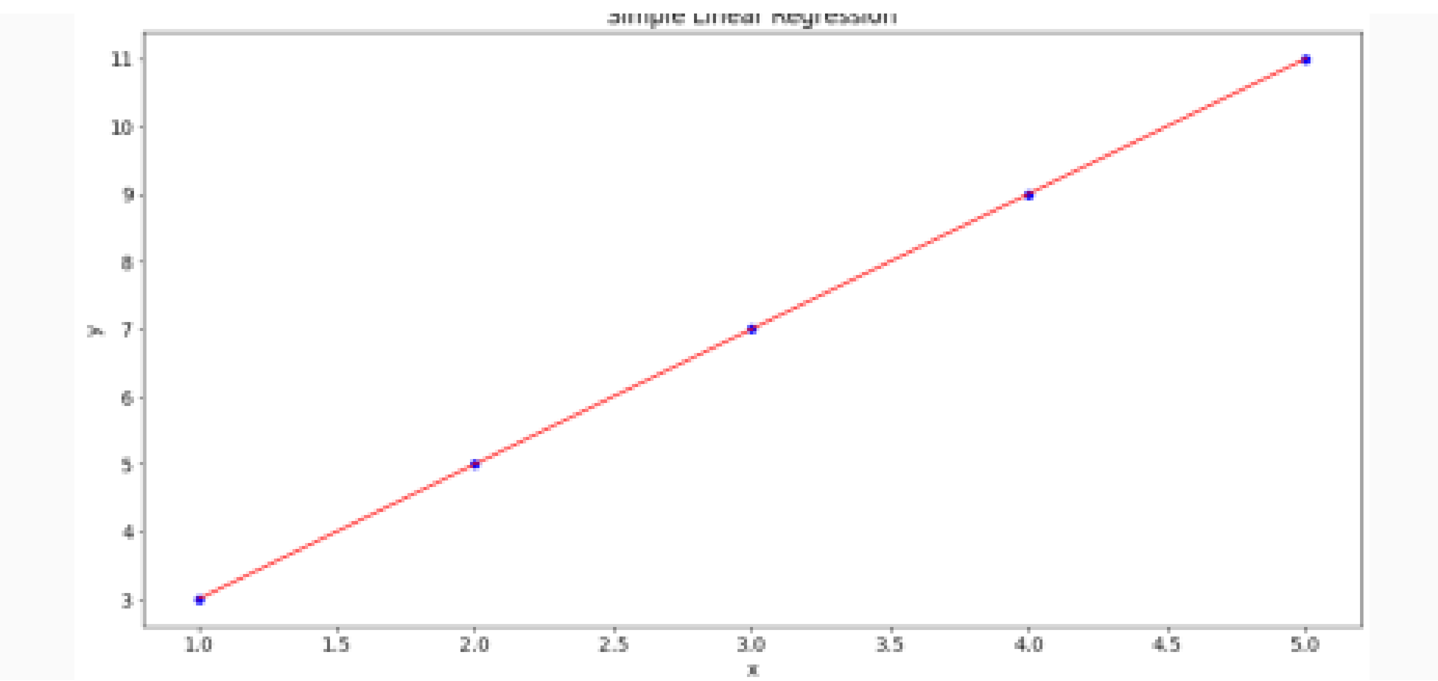
Esempio file: *simple_linear_regression_for _salary_data.py*

Scipy.stats: esempio (file simul_linregress.py)

```
import matplotlib.pyplot as plt
from scipy import stats
from numpy.random import randn
from numpy.random import seed

seed(1)

x = randn(10) # genero valori x casuali
y = 1.6*x + randn(10) # genero i valori y dipendenti da x in modo aleatorio
res = stats.linregress(x, y) # calcolo la regressione lineare di y rispetto ad x
print(f"R-squared: {res.rvalue**2:.6f}") #stampo il valore r^2

#grafico dati e retta
plt.plot(x, y, 'o', label='original data')
plt.plot(x, res.intercept + res.slope*x, 'r', label='fitted line')
plt.legend()
plt.show()
```

Scikitlearn: esempio simulazione (file example_sklearn.py)

```
from sklearn.linear_model import LinearRegression

# Generate sample data
x = np.array([1, 2, 3, 4, 5])
y = np.array([3, 5, 7, 9, 11])

# Reshape data
x = x.reshape(-1, 1)
y = y.reshape(-1, 1)

# Create linear regression object and fit the model
reg = LinearRegression().fit(x, y)

# Predict the y-values using the trained model
y_pred = reg.predict(x)
```



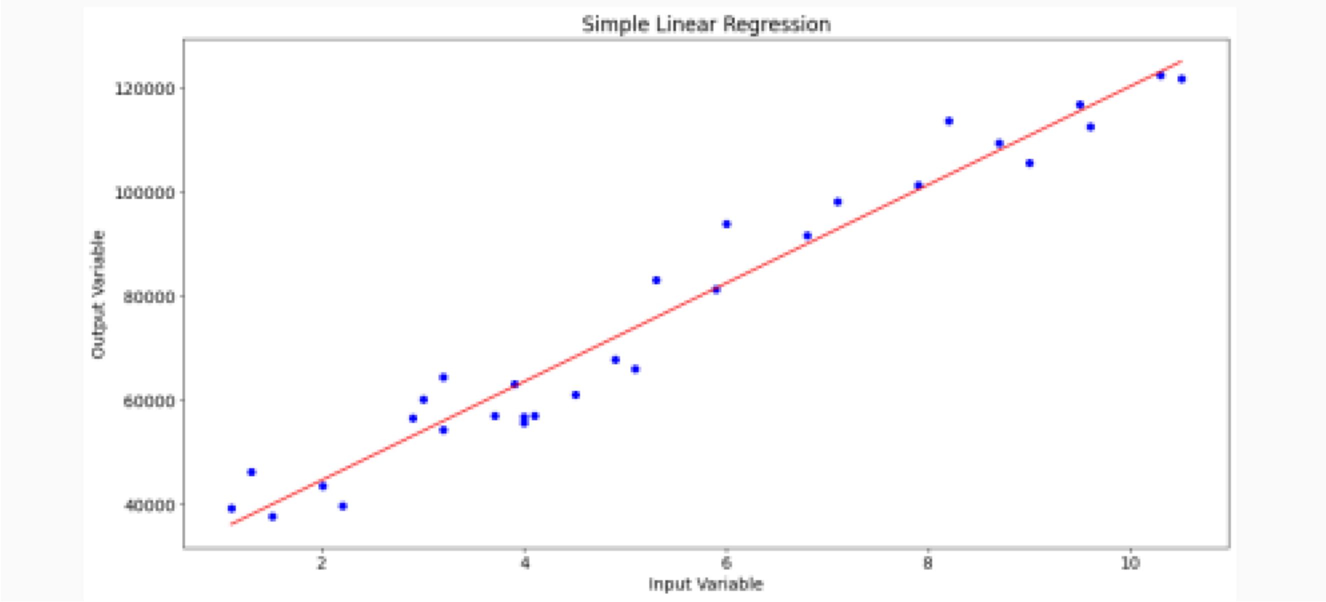
Scikitlearn: esempio simulazione

```
# Plot the data points and the linear regression line

plt.scatter(x, y, color='blue')
plt.plot(x, y_pred, color='red')

# Add labels and a title to the plot
plt.xlabel('x')
plt.ylabel('y')
plt.title('Simple Linear Regression')

# Display the plot
plt.show()
```

Scikitlearn: esempio su data set (file example_salary_sklearn.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Load data from Kaggle CSV file

#df = pd.read_csv('https://www.kaggle.com/vihansp/salary-data-simple-linear-regression')
df = pd.read_csv("Salary_Data.csv")

# Extract input and output variables
x = df['YearsExperience'].values.reshape(-1, 1)

y = df['Salary'].values.reshape(-1, 1)
print(reg.intercept_,reg.coef_)
```

Scikitlearn: esempio su data set




```
# Create linear regression object and fit the model
reg = LinearRegression().fit(x, y)

# Predict the y-values using the trained model
y_pred = reg.predict(x)

# Plot the data points and the linear regression line

plt.scatter(x, y, color='blue')
plt.plot(x, y_pred, color='red')

# Add labels and a title to the plot

plt.xlabel('Input Variable')
plt.ylabel('Output Variable')
plt.title('Simple Linear Regression')
```

Scikitlearn: esempio su data set (file linear_regression_for_salary_data.py)

In questo codice il data set viene partizionato in training set e test set.
I coefficienti vengono stimati utilizzando il training set e le metriche vengono calcolate sul test set.

```
# Split the data for train and test
X_train,X_test,y_train,y_test =
train_test_split(X,y,train_size=0.7,random_state=100)

# Importing Linear Regression model from scikit learn
from sklearn.linear_model import LinearRegression

# Fitting the model
lr = LinearRegression()
lr.fit(X_train,y_train)
```

Analisi dei residui

I residui sono le differenze fra i valori osservati y_i e i valori predetti $\hat{y}_i$
relativi alla stessa ascissa x_i.
I residui possono essere analizzati semplicemnte graficaente per vedere il loro andamento, tramite *scatterplot* o *istogramma in frequenza*.

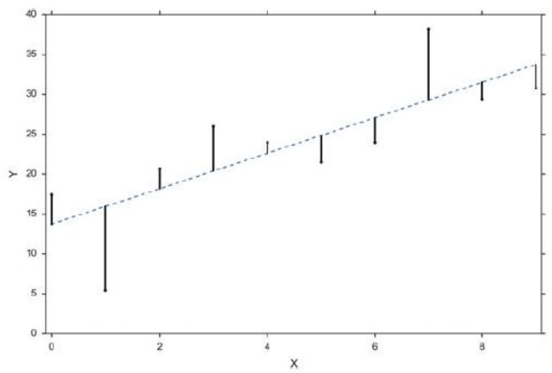
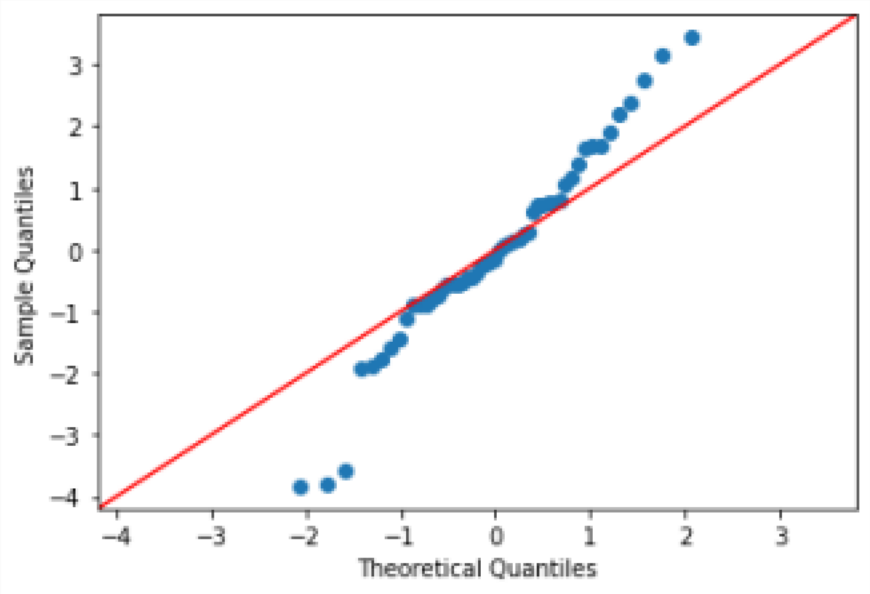


Fig. 11.3 Best-fit linear regression line (*dashed line*) and residuals (*solid lines*)

Funzioni Python: Esempio file *example_residuals.py*



Analisi dei residui

Sui residui viene fatta l'ipotesi di normalità, cioè si suppone, nel modello di regressione lineare, che i residui abbiano distribuzione normale con media 0 e deviazione standard σ

Non nota.

Per verificare l'ipotesi di normalità dei residui, posso fare il grafico QQ-plot.

Se le due distribuzioni sono simili, I punti devono stare molto vicini alla retta.

Python:

```
import statsmodels.api as sm
fig = sm.qqplot(data, line='45')
```

Test di ipotesi di normalità

Ci sono diversi test di ipotesi di normalita' basati sul confronto della distribuzione
stimata dei dati rispetto alla distribuzione normale.

Uno dei piu famosi e' il test di Shapiro-Wilk, che si basa sulla matrice di covarianza delle statitsiche ordinate delle osservazioni e puo' essere utilizzato anche con un numero ridotto (<+50) di osservazioni.

H_0: residui normali
H_a: residui non normali.

Test di ipotesi di normalità

Esempio:

```
from scipy.stats import shapiro

gfg_data = randn(500)
shapiro(gfg_data)
```

Output:

```
(0.9977102279663086, 0.7348126769065857)
```

Interpretazione dell'output: Poiché il p-value 0.73 >0.5 (livello di confidenza del test) Non c'è evidenza x rigettare l'ipotesi nulla, cioè per dire che i residui NON hanno distribuzione normale .

[scipy.stats.shapiro — SciPy v1.10.1 Manual]()