

Uso di Python per il Trattamento dei Dati Sperimentali in Fisica

Una guida per studenti delle scuole superiori

July 27, 2024

Contents

1	Introduzione	2
2	Concetti Fondamentali di Python	2
2.1	Variabili e Tipi di Dato	2
2.2	Liste	2
2.3	Cicli	3
2.4	Funzioni	3
2.5	Importazione di Librerie	3
3	Analisi dei Dati: Concetti di Base	3
3.1	Media e Deviazione Standard	3
3.2	Regressione Lineare	4
4	Esempi Pratici di Analisi dei Dati in Fisica	4
4.1	Calcolo della Media e della Deviazione Standard	4
4.2	Regressione Lineare	4

1 Introduzione

Il trattamento dei dati sperimentali è una parte fondamentale del lavoro di ogni fisico. Per analizzare i dati in modo efficiente e preciso, l'uso di strumenti informatici è diventato indispensabile. Python è uno dei linguaggi di programmazione più popolari e potenti per l'analisi dei dati, grazie alla sua semplicità e alla vasta gamma di librerie disponibili. In questo articolo, esploreremo l'uso di Python e di alcune delle sue librerie principali per l'analisi dei dati sperimentali in fisica.

2 Concetti Fondamentali di Python

Prima di addentrarci nell'analisi dei dati, è importante conoscere alcuni concetti fondamentali di Python. Vediamo alcuni esempi pratici.

2.1 Variabili e Tipi di Dato

In Python, le variabili possono contenere diversi tipi di dati, come numeri interi, numeri in virgola mobile, stringhe e booleani.

```
# Esempi di variabili
intero = 10
virgola_mobile = 10.5
stringa = "Ciao, mondo!"
booleano = True

print(intero)
print(virgola_mobile)
print(stringa)
print(booleano)
```

2.2 Liste

Le liste sono usate per memorizzare più valori in una singola variabile. Le liste in Python sono ordinate e modificabili.

```
# Creazione di una lista
numeri = [1, 2, 3, 4, 5]

# Accesso agli elementi della lista
print(numeri[0]) # Output: 1

# Modifica di un elemento della lista
numeri[1] = 10
print(numeri) # Output: [1, 10, 3, 4, 5]

# Aggiunta di un elemento alla lista
numeri.append(6)
print(numeri) # Output: [1, 10, 3, 4, 5, 6]
```

2.3 Cicli

I cicli sono utilizzati per iterare su una sequenza di elementi, come una lista.

```
# Ciclo for
for numero in numeri:
    print(numero)

# Ciclo while
contatore = 0
while contatore < 5:
    print(contatore)
    contatore += 1
```

2.4 Funzioni

Le funzioni sono blocchi di codice che eseguono un compito specifico e possono essere richiamate quando necessario.

```
# Definizione di una funzione
def saluta(nome):
    return f"Ciao, {nome}!"

# Chiamata della funzione
saluto = saluta("Alice")
print(saluto) # Output: Ciao, Alice!
```

2.5 Importazione di Librerie

Python ha molte librerie che estendono le sue funzionalità. Possiamo importare queste librerie per utilizzare le loro funzioni.

```
import numpy as np

# Uso della libreria numpy per creare un array
array = np.array([1, 2, 3, 4, 5])
print(array)
```

3 Analisi dei Dati: Concetti di Base

Prima di addentrarci nei dettagli delle librerie Python, è utile rivedere alcuni concetti fondamentali dell'analisi dei dati in fisica.

3.1 Media e Deviazione Standard

La media aritmetica di un insieme di dati è la somma dei valori divisa per il numero dei valori. La deviazione standard misura la dispersione dei dati rispetto alla media.

3.2 Regressione Lineare

La regressione lineare è una tecnica statistica utilizzata per modellare la relazione tra una variabile dipendente e una o più variabili indipendenti.

4 Esempi Pratici di Analisi dei Dati in Fisica

4.1 Calcolo della Media e della Deviazione Standard

Consideriamo un esperimento di misura del tempo di caduta di un oggetto. Abbiamo i seguenti tempi di caduta (in secondi): 2.3, 2.5, 2.4, 2.6, 2.4.

```
import numpy as np

# Dati sperimentali
tempi = np.array([2.3, 2.5, 2.4, 2.6, 2.4])

# Calcolo della media
media = np.mean(tempi)
print(f"La media dei tempi di caduta è: {media:.2f} s")

# Calcolo della deviazione standard
deviazione_standard = np.std(tempi)
print(f"La deviazione standard dei tempi di caduta è:
→ {deviazione_standard:.2f} s")
```

4.2 Regressione Lineare

Consideriamo un esperimento di misura della velocità in funzione del tempo per un oggetto in moto rettilineo uniforme. I dati sono i seguenti:

Tempo (s)	Velocità (m/s)
0	0
1	2
2	4
3	6
4	8

Table 1: Dati di velocità in funzione del tempo

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Dati sperimentali
tempo = np.array([0, 1, 2, 3, 4])
velocita = np.array([0, 2, 4, 6, 8])

# Aggiunta degli errori di misura
errori_tempo = np.array([0.1, 0.1, 0.1, 0.1, 0.1])
errori_velocita = np.array([0.2, 0.2, 0.2, 0.2, 0.2])

# Calcolo della regressione lineare
slope, intercept, r_value, p_value, std_err = linregress(tempo, velocita)

# Stampa dei risultati
print(f"Coefficiente angolare (velocità): {slope:.2f} m/s")
print(f"Intercetta: {intercept:.2f} m")
print(f"R-quadrato: {r_value**2:.2f}")

# Grafico dei dati e della retta di regressione con barre di errore
plt.errorbar(tempo, velocita, xerr=errori_tempo, yerr=errori_velocita, fmt='o', label=
plt.plot(tempo, slope*tempo + intercept, color='red', label='Retta di regressione')
plt.xlabel('Tempo (s)')
plt.ylabel('Velocità (m/s)')
plt.legend()
plt.title('Regressione Lineare della Velocità in Funzione del Tempo con Barre di Erro
plt.show()

```