



Esercitazione Visualizzazione

Python

In queste slide andremo ad avere degli esercizi con le correzioni per allenarci in MatPlot e SeaBorn

è consigliato aver già affrontato i moduli inerenti alle librerie prima di approcciarsi a questa esercitazione

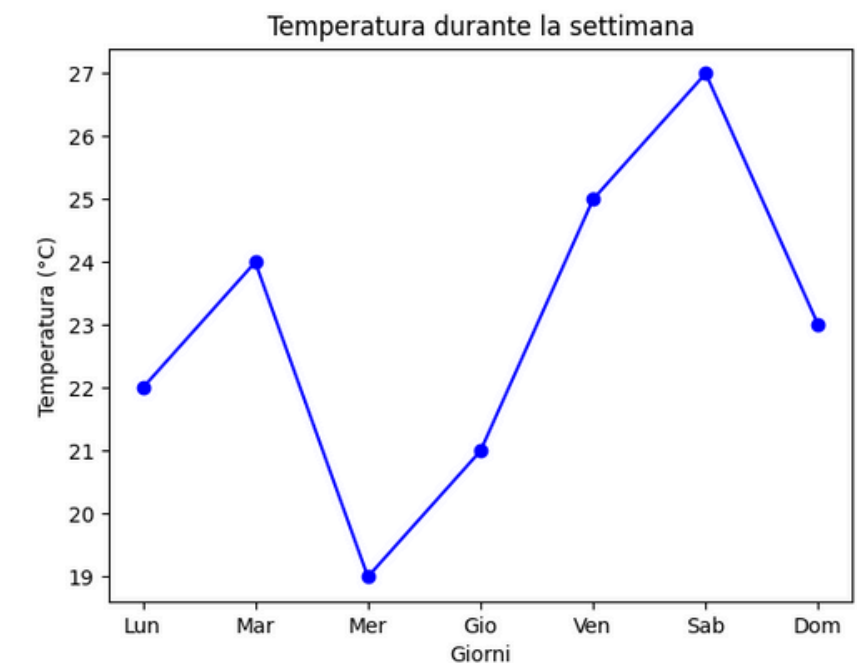


Esercizio Matplotlib:

Iniziamo con un esercizio semplice, disegnare un semplice grafico a linee che mostri la variazione della temperatura in una settimana, con dati di mockup.



```
1. import matplotlib.pyplot as plt
2.
3. # Dati esempio
4. giorni = ['Lun', 'Mar', 'Mer', 'Gio', 'Ven', 'Sab', 'Dom']
5. temperature = [22, 24, 19, 21, 25, 27, 23]
6.
7. # Creazione del grafico
8. plt.plot(giorni, temperature, marker='o', linestyle='-', color='b')
9.
10. # Aggiungere titolo e etichette
11. plt.title("Temperatura durante la settimana")
12. plt.xlabel("Giorni")
13. plt.ylabel("Temperatura (°C)")
14.
15. # Mostra il grafico
16. plt.show()
```



Correzione esercizio Matplotlib:

- Ho aggiunto la funzione `plt.plot()` per generare un grafico a linee.
- Ho usato `marker='o'` per evidenziare i punti.
- Le etichette per gli assi sono state definite con `xlabel()` e `ylabel()`, mentre il titolo è stato aggiunto con `title()`.
- I colori e gli stili del grafico sono stati definiti per renderlo più chiaro.



Esercizio Seaborn:

Similarmente a quanto fatto sopra ora utilizziamo Seaborn per creare un boxplot che rappresenta la distribuzione delle temperature della settimana.



```
1.import seaborn as sns  
2.import matplotlib.pyplot as plt  
3.  
4.# Dati esempio  
5.temperature_settimanali = [22, 24, 19, 21, 25, 27, 23]  
6.  
7.# Creazione del boxplot  
8.sns.boxplot(data=temperature_settimanali)  
9.  
10.# Aggiungere titolo  
11.plt.title("Distribuzione delle temperature settimanali")  
12.  
13.# Mostra il grafico  
14.plt.show()  
15.
```



Esercizio Matplotlib - Grafico a barre con annotazioni:

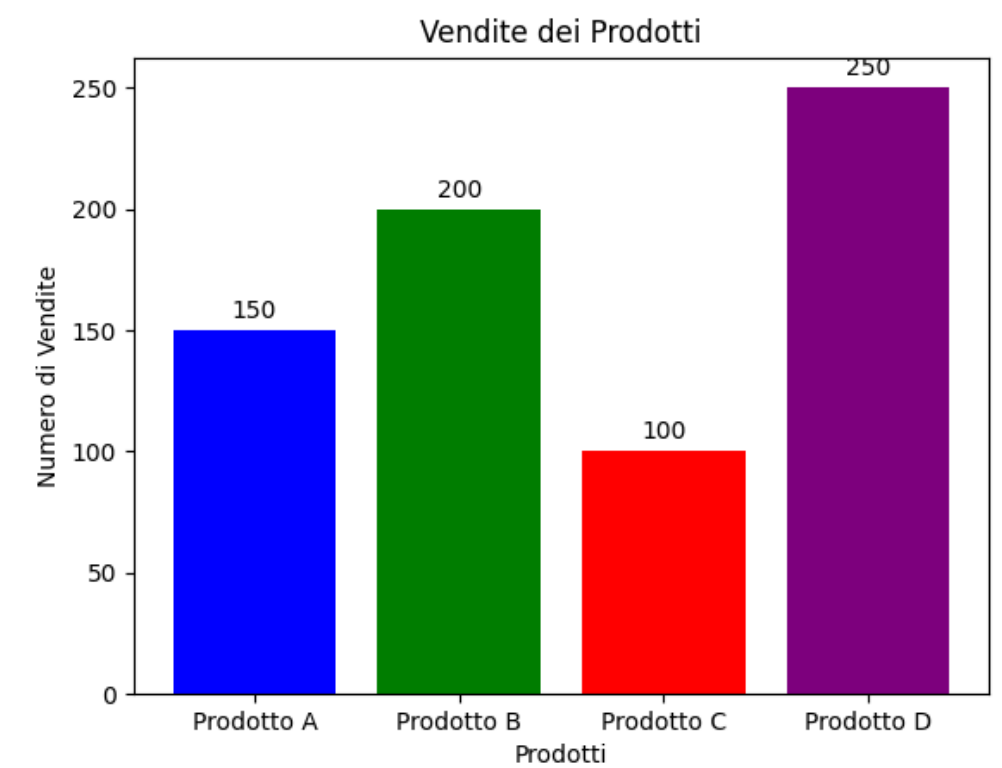
Il grafico a barre è uno dei grafici più utilizzati per rappresentare dati categoriali.

In questo esercizio, creiamo un grafico a barre che rappresenta le vendite di diversi prodotti in un negozio e aggiungiamo delle annotazioni sopra le barre per indicare il numero esatto delle vendite.

Le annotazioni rendono il grafico più leggibile e informativo.



```
1. import matplotlib.pyplot as plt
2.
3. # Dati di esempio
4. prodotti = ['Prodotto A', 'Prodotto B', 'Prodotto C', 'Prodotto D']
5. vendite = [150, 200, 100, 250]
6.
7. # Creazione del grafico a barre
8. plt.bar(prodotti, vendite, color=['blue', 'green', 'red', 'purple'])
9.
10. # Aggiungere titolo e etichette
11. plt.title("Vendite dei Prodotti")
12. plt.xlabel("Prodotti")
13. plt.ylabel("Numero di Vendite")
14.
15. # Aggiungere annotazioni sulle barre
16. for i in range(len(vendite)):
17.     plt.text(i, vendite[i] + 5, str(vendite[i]), ha='center')
18.
19. # Mostra il grafico
20. plt.show()
21.
```



Spiegazione:

- **Dati:** Abbiamo quattro prodotti e il loro numero di vendite.
- **plt.bar():** Utilizziamo `bar()` per creare un grafico a barre. Ogni barra rappresenta un prodotto.
- **Colori personalizzati:** Impostiamo colori diversi per ciascuna barra utilizzando il parametro `color`.
- **Annotazioni:** Utilizziamo la funzione `plt.text()` per aggiungere il numero di vendite sopra ogni barra. Il parametro `ha='center'` serve a centrare il testo sopra la barra.
- **Aspetti estetici:** Aggiungiamo titolo e etichette sugli assi per rendere il grafico più chiaro.



Esercizio Seaborn Heatmap (Mappa di calore):

Una heatmap è uno strumento molto potente per visualizzare matrici di dati.

In una mappa di calore, i valori numerici vengono rappresentati attraverso variazioni di colore, consentendo di identificare facilmente le tendenze o le anomalie.

In questo esercizio, creiamo una heatmap che rappresenta la correlazione tra diverse variabili in un dataset.



```
1. import seaborn as sns
2. import matplotlib.pyplot as plt
3. import numpy as np
4.
5. # Creazione di un set di dati fittizio (matrice di correlazione)
6. np.random.seed(0)
7. dati = np.random.rand(10, 10) # Matrice 10x10 di valori casuali
8.
9. # Creazione della heatmap
10. plt.figure(figsize=(8, 6))
11. sns.heatmap(dati, annot=True, cmap='coolwarm', linewidths=.5)
12.
13. # Aggiungere titolo
14. plt.title("Mappa di Calore dei Valori Casuali")
15.
16. # Mostra il grafico
17. plt.show()
```

Spiegazione:

- **Dati:** Utilizziamo numpy per creare una matrice di dati casuali (10x10) che verranno rappresentati sulla mappa di calore. Questa matrice potrebbe essere, ad esempio, una matrice di correlazione tra variabili in un dataset.
- **sns.heatmap():** La funzione heatmap() di Seaborn è utilizzata per creare la mappa di calore. Impostiamo annot=True per visualizzare i valori numerici all'interno di ciascuna cella e scegliamo una scala di colori personalizzata tramite cmap='coolwarm'.
- **Parametri estetici:** Utilizziamo linewidths=.5 per aggiungere delle linee di separazione tra le celle, rendendo la visualizzazione più pulita
- **Visualizzazione:** Impostiamo una dimensione specifica per la figura tramite plt.figure(figsize=(8, 6)) per evitare che le celle risultino troppo piccole o troppo grandi.



Esercizio Matplotlib Grafico 3D di una superficie:

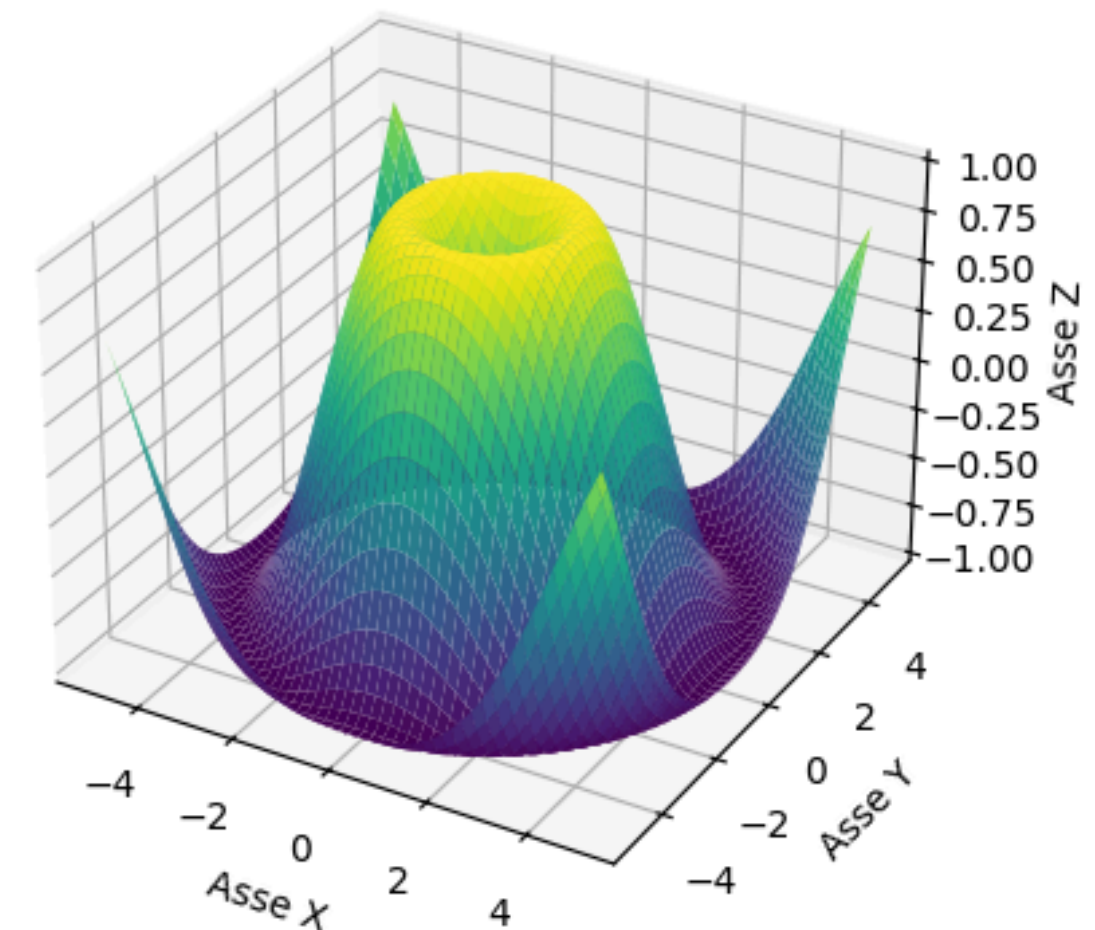
I grafici 3D sono utili per rappresentare dati che variano su due assi indipendenti, consentendo di visualizzare l'andamento di funzioni su una superficie.

In questo esercizio, disegneremo un grafico tridimensionale che rappresenta una funzione matematica, come ad esempio la funzione $z = \sin(x^2 + y^2)$.



```
1.import matplotlib.pyplot as plt
2.from mpl_toolkits.mplot3d import Axes3D
3.import numpy as np
4.
5.# Creazione dei dati
6.x = np.linspace(-5, 5, 100)
7.y = np.linspace(-5, 5, 100)
8.x, y = np.meshgrid(x, y)
9.z = np.sin(np.sqrt(x**2 + y**2))
10.
11.# Creazione del grafico 3D
12.fig = plt.figure()
13.ax = fig.add_subplot(111, projection='3d')
14.
15.# Plot della superficie
16.ax.plot_surface(x, y, z, cmap='viridis')
17.
18.# Aggiungere etichette agli assi
19.ax.set_title("Grafico 3D di una superficie")
20.ax.set_xlabel("Asse X")
21.ax.set_ylabel("Asse Y")
22.ax.set_zlabel("Asse Z")
23.
24.# Mostra il grafico
25 plt.show()
```

Grafico 3D di una superficie



Spiegazione:

- **Dati:** Generiamo una griglia di valori x e y utilizzando `np.meshgrid()`. La funzione $z = \sin(\sqrt{x^2 + y^2})$ rappresenta una superficie ondulata che varia in funzione di x e y .
- **Grafico 3D:** Usiamo `Axes3D` di `mpl_toolkits.mplot3d` per creare il grafico tridimensionale. La funzione `plot_surface()` è utilizzata per disegnare la superficie.
- **Colormap:** Applichiamo una mappa di colori (in questo caso 'viridis') per rappresentare i valori di z attraverso variazioni di colore sulla superficie.
- **Etichette degli assi:** Impostiamo le etichette per tutti e tre gli assi, x , y e z , per una maggiore chiarezza.



Esercizio Matplotlib Subplot multipli con istogramma, grafico a dispersione e grafico a linee:

In molte applicazioni, potresti voler mostrare più grafici su una singola finestra per confrontare diversi aspetti dei dati.

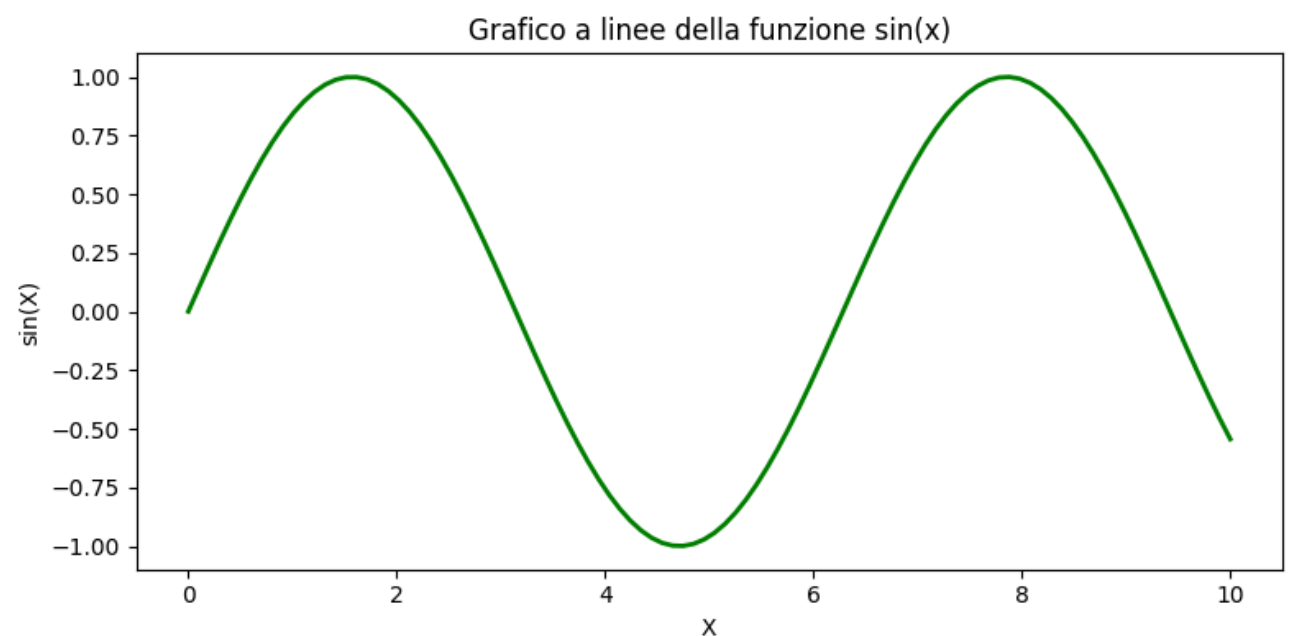
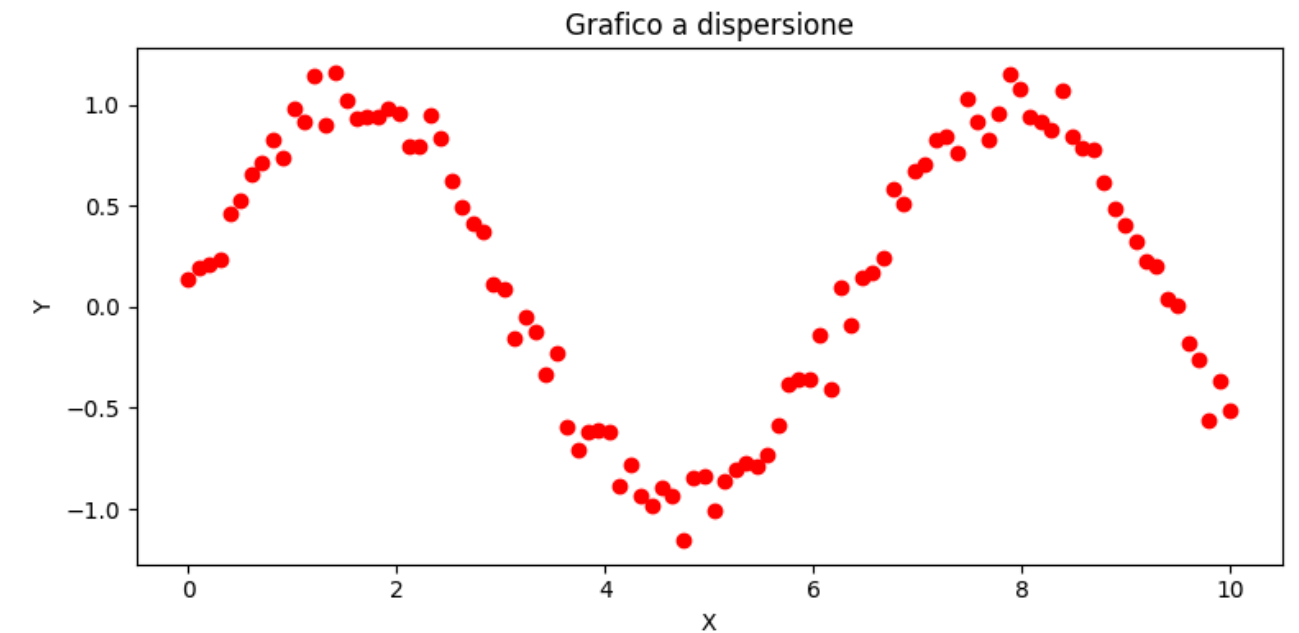
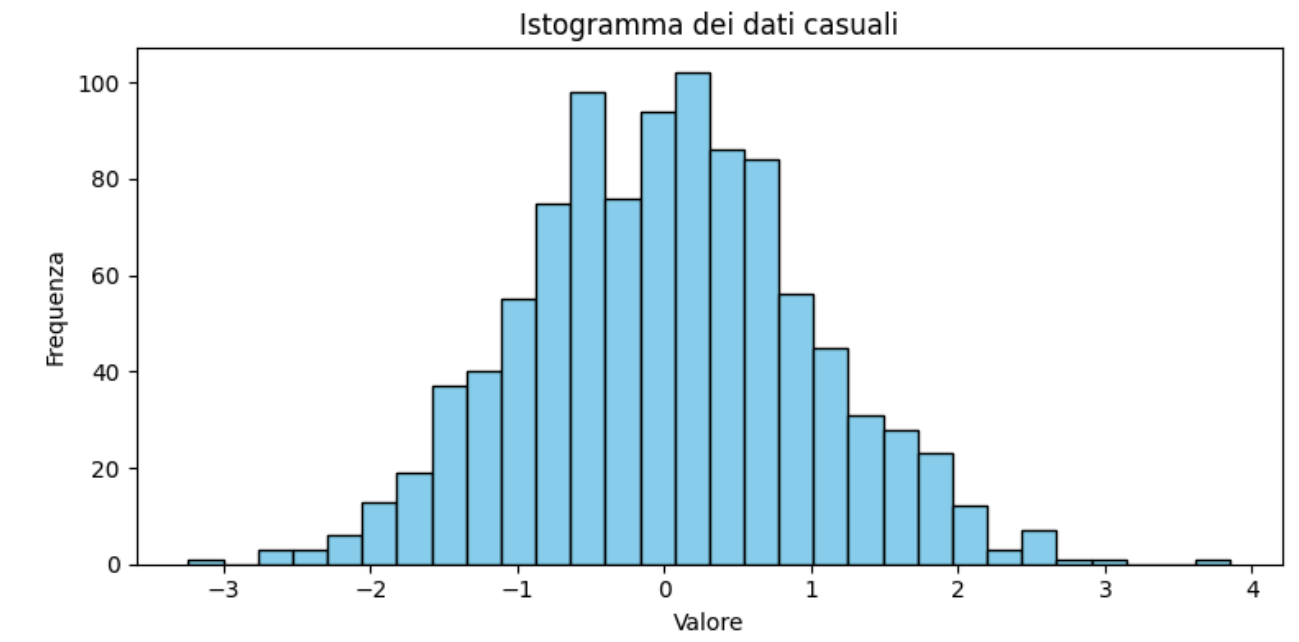
In questo esercizio, creiamo una figura con tre subplots: un istogramma, un grafico a dispersione e un grafico a linee, utilizzando lo strumento subplot di Matplotlib.



```

1. import matplotlib.pyplot as plt
2. import numpy as np
3.
4. # Generazione dei dati
5. np.random.seed(42)
6. dati = np.random.randn(1000)
7. x = np.linspace(0, 10, 100)
8. y = np.sin(x) + 0.1 * np.random.randn(100)
9.
10. # Creazione della figura e dei subplot
11. fig, axs = plt.subplots(3, 1, figsize=(8, 12))
12.
13. # Subplot 1: Istogramma
14. axs[0].hist(dati, bins=30, color='skyblue', edgecolor='black')
15. axs[0].set_title("Istogramma dei dati casuali")
16. axs[0].set_xlabel("Valore")
17. axs[0].set_ylabel("Frequenza")
18.
19. # Subplot 2: Grafico a dispersione (scatter plot)
20. axs[1].scatter(x, y, color='red', marker='o')
21. axs[1].set_title("Grafico a dispersione")
22. axs[1].set_xlabel("X")
23. axs[1].set_ylabel("Y")
24.
25. # Subplot 3: Grafico a linee
26. axs[2].plot(x, np.sin(x), color='green', linestyle='-', linewidth=2)
27. axs[2].set_title("Grafico a linee della funzione sin(x)")
28. axs[2].set_xlabel("X")
29. axs[2].set_ylabel("sin(X)")
30.
31. # Regolazione dello spazio tra i subplots
32. plt.tight_layout()
33.
34. # Mostra il grafico
35. plt.show()

```



Spiegazione:

- **Generazione dei dati:**
 - **dati:** Generiamo 1000 numeri casuali distribuiti normalmente per l'istogramma.
 - **x e y:** Generiamo dati per il grafico a dispersione e il grafico a linee. Il grafico a dispersione mostra la funzione $\sin(x)$ con rumore aggiunto, mentre il grafico a linee traccia la funzione $\sin(x)$ pura.
- **Subplot 1 - Istogramma:** Il primo subplot è un istogramma dei dati casuali, che mostra la distribuzione dei valori. Usiamo 30 bin per suddividere i dati e `edgecolor` per migliorare la visibilità dei bordi.
- **Subplot 2 - Grafico a dispersione:** Il secondo subplot è un grafico a dispersione (scatter plot) di x e y. Qui, y è calcolato come $\sin(x)$ con del rumore casuale aggiunto per simulare una serie di dati reali. Ogni punto è rappresentato da un marcatore rosso.
- **Subplot 3 - Grafico a linee:** Il terzo subplot è un semplice grafico a linee della funzione $\sin(x)$. Usiamo `plot()` per tracciare una curva liscia che rappresenta la funzione $\sin(x)$, senza rumore.
- **Gestione degli spazi:** Utilizziamo `plt.tight_layout()` per gestire lo spazio tra i subplot, in modo che i titoli e le etichette non si sovrappongano.

