Matplotlib w Pythonie: Praktyczne przykłady wizualizacji danych

W tej prezentacji poznasz podstawowe i zaawansowane przykłady kodu do tworzenia wykresów w bibliotece Matplotlib. Skupimy się na praktycznych przykładach, które pomogą Ci zrozumieć, jak efektywnie wizualizować dane w Pythonie. Nauka zacznie się od prostych wykresów liniowych, a następnie przejdziemy do różnych typów wykresów oraz sposobów ich personalizacji.

Po każdej części znajdziesz fragmenty kodu oraz wizualizacje, które wzbogacą Twoją wiedzę i umiejętności tworzenia grafik w Matplotlib.

by r janiak

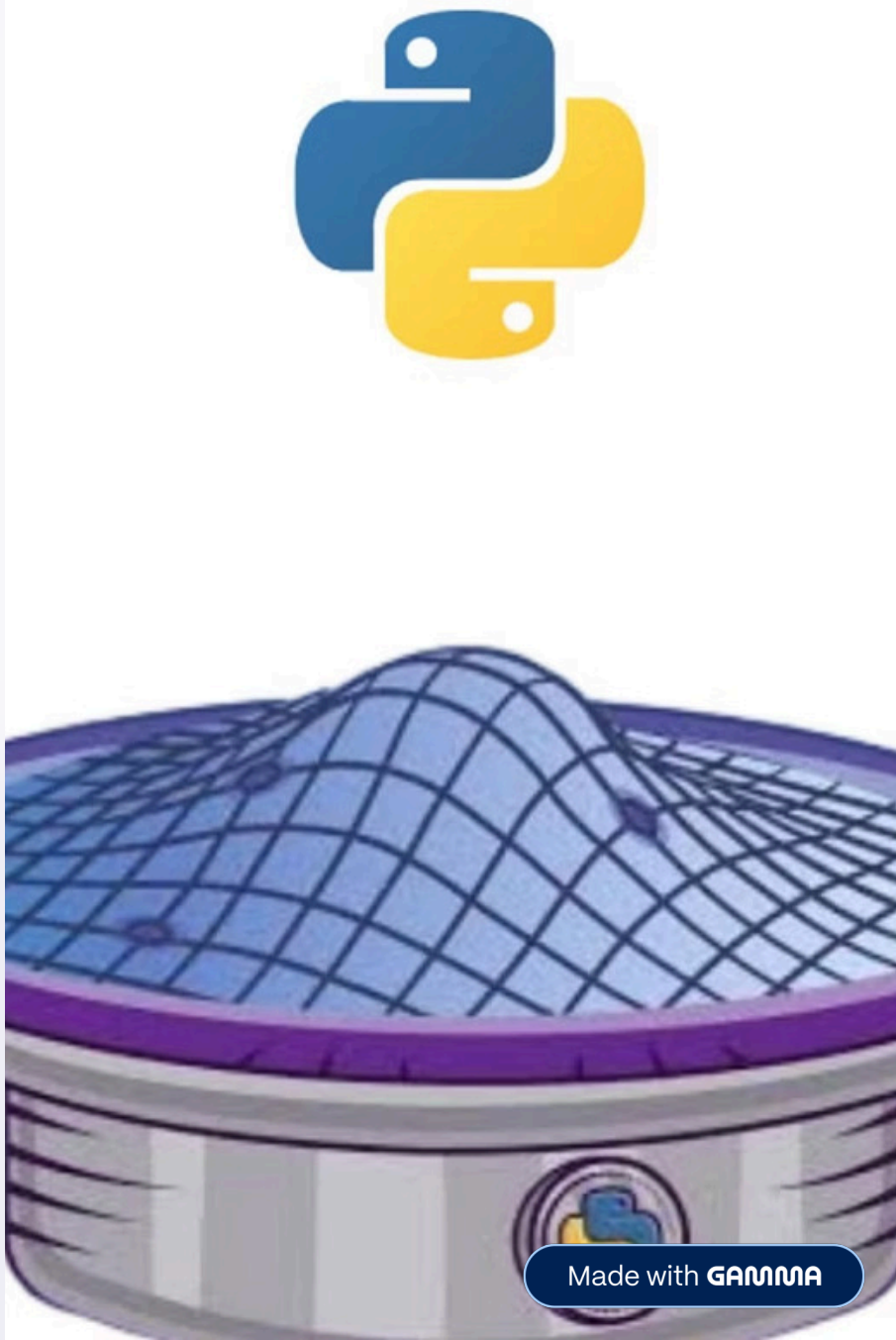
Wykres liniowy – Podstawy i pierwsze przykłady

Podstawowy wykres liniowy

Funkcja plt.plot() tworzy wykres liniowy na podstawie danych x i y. Przykład: plt.plot([1,3,2],[2,4,2]) rysuje linię łączącą punkty o podanych współrzędnych.

Domyślne wartości osi X

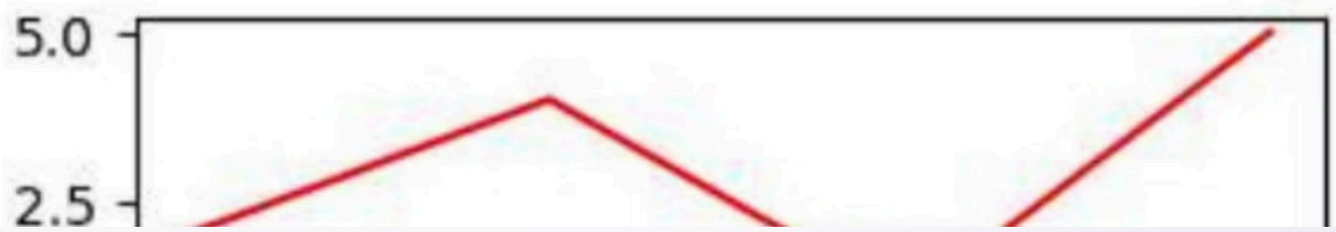
Jeśli nie podasz wartości osi X, matplotlib użyje domyślnych indeksów 0, 1, 2, ...



Formatowanie wykresu – kolory, style i znaczniki

```
plt.plot(ypoints,'r')

plt.show()
```



Zmiana koloru linii

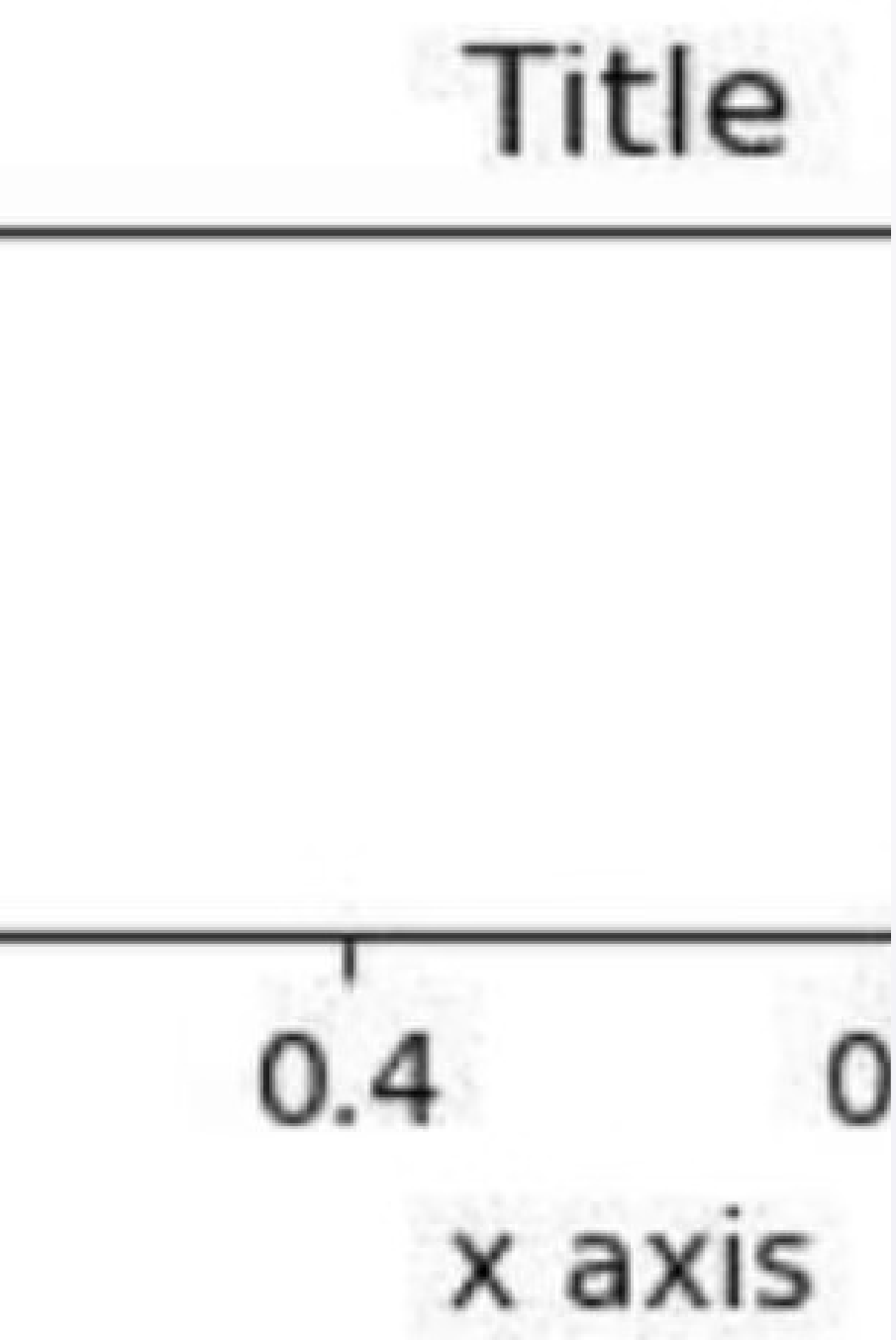
Możesz użyć nazwy koloru, skrótu (np. 'r' dla czerwonego) lub kodów RGB i HEX w trzecim argumencie funkcji plt.plot.

Style linii i znaczniki

Parametry linestyles ('--', '-.', ':') i markery ('o', 'x') pozwalają urozmaicić wygląd wykresu. Przykład: plt.plot(x, y, 'o:r') tworzy czerwoną linię kropkowaną z kółkami.

Rozmiar i kolor markerów

Parametry markersize, markeredgecolor i markerfacecolor pozwalają kontrolować wielkość oraz kolor krawędzi i wypełnienia markerów.



Dodawanie etykiet i tytułów – personalizacja napisów

Etykiety osi

Funkcje plt.xlabel() i plt.ylabel() pozwalają dodać opisy osi X i Y, zwiększając czytelność wykresu.

Tytuł wykresu

plt.title() wstawia tytuł, który można sformatować za pomocą fontdict – kontrolując rodzinę czcionki, kolor i rozmiar liter.

Dostosowanie fontów

Korzystając z parametru fontdict możesz indywidualnie ustawić styl napisów, np. niebieski tytuł z podpisem osi w ciemnej czerwieni.

Siatka na wykresie – ułatwienie czytania danych

Aktywacja siatki

Funkcja plt.grid() rysuje linie siatki na wykresie, co pomaga w odczytywaniu dokładnych wartości.

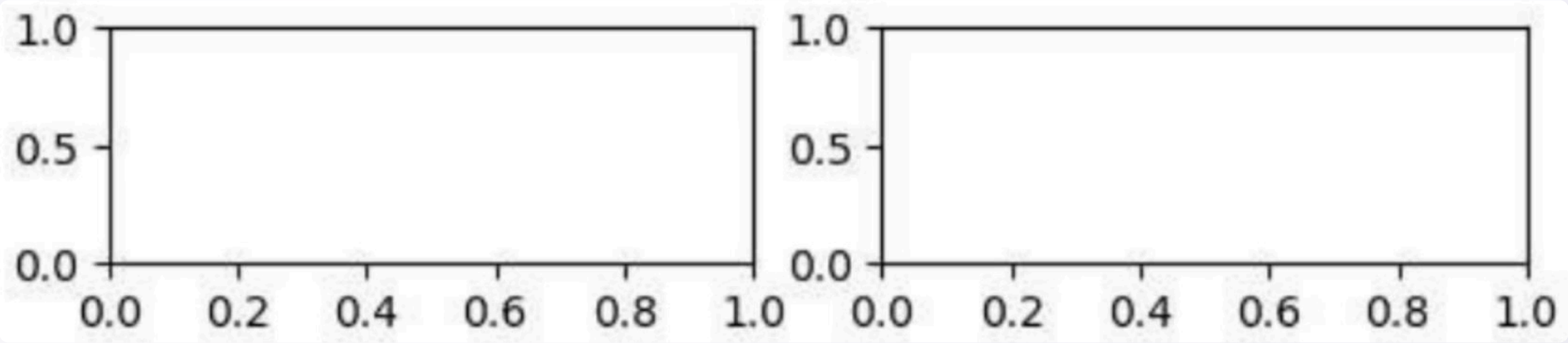
Siatka na wybranych osiach

Możesz wybrać osie, na których linie siatki mają się pojawić: plt.grid(axis='x') lub plt.grid(axis='y').

Właściwości linii siatki

Parametry koloru, stylu i szerokości np. plt.grid(color='green', linestyle='--', linewidth=1.5) pozwalają dostosować wygląd siatki.

Wykresy wielopanelowe – subplots dla wielu wykresów



Podział na subploty

plt.subplot(rows, columns, indeks) pozwala umieścić wiele wykresów w jednej figurze – np. 1x2 dla dwóch wykresów obok siebie.

Metoda subplots()

fig, ax = plt.subplots(m, n) zwraca figurę i tablicę osi, z którymi możesz operować niezależnie – idealne do złożonych layoutów wykresów.

Przykłady wykresów wielopanelowych z podziałem na osie

1

Różne typy wykresów

Możesz tworzyć różne typy wykresów np. liniowy i słupkowy, korzystając z ax[0].plot() oraz ax[1].bar()

2

Układy pionowe i poziome

Subploty można układać pionowo (rows > 1) lub poziomo (columns > 1) dostosowując kompozycję do potrzeb wizualizacji.

3

Wiele osi na figurze

Przy pomocy tuple unpacking np. fig, (ax1, ax2) = plt.subplots(1, 2) wygodnie zarządzasz wieloma osiami.

Różne typy wykresów: słupkowy, scatter i kołowy

Wykres słupkowy – bar()

plt.bar(x, y) tworzy wykres słupkowy, gdzie y oznacza wysokość słupka. Możesz zmieniać kolor i szerokość słupków.

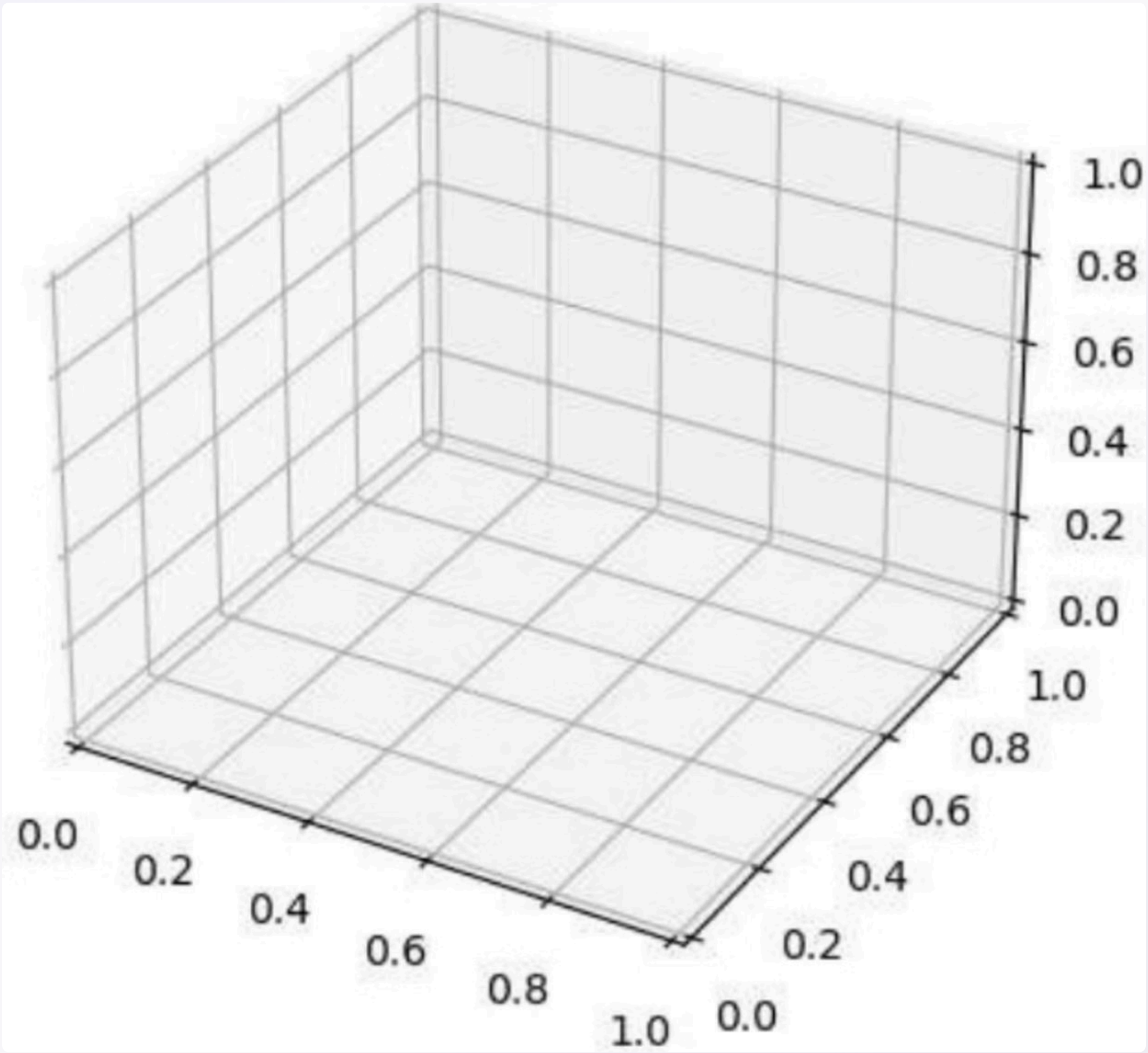
Wykres rozrzutu – scatter()

plt.scatter(x, y) rysuje punkty, przydatne do analizy korelacji. Parametr s zmienia rozmiar punktów.

Wykres kołowy – pie()

Wykres kołowy przedstawia skład procentowy danych. plt.pie(x, labels=y) tworzy wykres z podpisami.

Wykresy 3D i praca z grafikami rastrowymi



Wykresy 3D

Wykorzystaj plt.axes(projection='3d') aby tworzyć wykresy trójwymiarowe, np. linie czy powierzchnie, dla danych z trzema osiami.

Praca z obrazami

Moduł PIL umożliwia ładowanie i wyświetlanie obrazów na wykresie poprzez plt.imshow(). Możesz dodawać efekty, np. koloryzację i konwersję do skali szarości.

Kluczowe wnioski i kolejne kroki w nauce Matplotlib

Zrozum podstawy

Opanuj podstawy tworzenia wykresów, personalizacji kolorów, stylów i etykiet dla jasnej prezentacji danych.

Zastosuj wykresy wielopanelowe

Eksperymentuj z funkcją subplots, by tworzyć bogate wizualizacje z wieloma wykresami w jednym oknie.

Poszerzaj wiedzę

Pracuj z zaawansowanymi typami wykresów, 3D oraz obrazami, a także poznawaj dodatkowe funkcje planowania i animacji.

Pamiętaj, że praktyka to klucz! Ćwicz pisanie kodu i eksperymentuj z parametrami, aby tworzyć wyraziste, czytelne i funkcjonalne wykresy.