

Matplotlib - это основная библиотека для построения научных графиков в Python. [1] Включает функции для создания высококачественных визуализаций: линейных диаграмм, гистограмм и т.д. Визуализация данных и результатов - цель использования библиотеки matplotlib. При работе в среде можно вывести рисунок на экран с помощью встроенных команд:

%matplotlib notebook для визуализации графика в интерактивном режиме; %matplotlib inline для получения статичного изображения.

1. Рисунок (Figure) Любой рисунок в matplotlib имеет вложенную структуру. Рисунок - это объект самого верхнего уровня, на котором располагаются: области рисования (Axes); элементы рисунка Artists (заголовки, легенда и т.д.); основа-холст (Canvas). На рисунке может быть несколько областей рисования Axes, но данная область рисования Axes может принадлежать только одному рисунку Figure.
2. Область рисования (Axes) Объект среднего уровня. Это часть изображения с пространством данных. Каждая область рисования Axes содержит две (или три в случае трёхмерных данных) координатных оси (Axis объектов), которые упорядочивают отображение данных.
3. Координатная ось (Axis) Координатная ось является объектом среднего уровня, которая определяет область изменения данных. На них наносятся: деления ticks; подписи к делениям ticklabels. Расположение делений определяется объектом Locator, а подписи делений обрабатывает объект Formatter. Конфигурация координатных осей заключается в комбинировании различных свойств объектов Locator и Formatter.
4. Элементы рисунка (Artists) Практически всё, что отображается на рисунке является элементом рисунка (Artist), даже объекты Figure, Axes и Axis. Элементы рисунка Artists включают в себя такие простые объекты как: текст (Text); плоская линия (Line2D); фигура (Patch) и другие. Когда происходит отображение рисунка (figure rendering), все элементы рисунка Artists наносятся на основу-холст (Canvas). Большая часть из них связывается с областью рисования Axes. Также элемент рисунка не может совместно использоваться несколькими областями Axes или быть перемещён с одной на другую
5. Самые простые графические команды plt.scatter() - маркер или точечное рисование; plt.plot() - ломаная линия; plt.text() - нанесение текста.
6. Диаграммы plt.bar(), plt.barh(), plt.barbs(), broken\_barh() - столбчатая диаграмма; plt.hist(), plt.hist2d(), plt.hlines - гистограмма; plt.pie() - круговая диаграмма; plt.boxplot() - "ящик с усами" (boxwhisker); plt.errorbar() - оценка погрешности, "усы".
7. Изображения в изолиниях plt.contour() - изолинии; plt.contourf() - изолинии с послойной окраской.
8. Отображения plt.pcolor(), plt.pcolormesh() - псевдоцветное изображение матрицы (2D массива); plt.imshow() - вставка графики (пиксели + сглаживание); plt.matshow() - отображение данных в виде квадратов.
9. Заливка plt.fill() - заливка многоугольника; plt.fill\_between(), plt.fill\_betweenx() - заливка между двумя линиями.
10. Векторные диаграммы plt.streamplot() - линии тока; plt.quiver() - векторное поле.

Всё пространство рисунка Figure (прямоугольной или иной формы) можно использовать для нанесения других элементов рисунка, например, контейнеров Axes, графических примитивов в виде линий, фигур, текста и так далее. В любом случае каждый рисунок можно структурно представить следующим образом:

1. Область рисования Axes Заголовок области рисования -> `plt.title()`;
2. Ось абсцисс Xaxis Подпись оси абсцисс OX -> `plt.xlabel()`; `fig = plt.figure()` `ax = fig.add_axes([0, 0, 1, 1], polar=True)` `plt.scatter(0.0, 0.5)` `plt.show()`
3. Ось ординат Yaxis Подпись оси ординат OY -> `plt.ylabel()`;
4. Легенда -> `plt.legend()`;
5. Цветовая шкала -> `plt.colorbar()` Подпись горизонтальной оси абсцисс OY -> `cbar.ax.set_xlabel()`; Подпись вертикальной оси абсцисс OY -> `cbar.ax.set_ylabel()`;
6. Деления на оси абсцисс OX -> `plt.xticks()`;
7. Деления на оси ординат OY -> `plt.yticks()`.