

Практика 1_4-1. Основные компоненты matplotlib

1. Начало работы

Для начала работы выполним следующий код:

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

Пример 1. Создадим область

```
fig = plt.figure()
ax = fig.add_subplot(111)

plt.show()
```

Пример 2. Покажем цветом области *Figure* и *Axes*:

```
fig = plt.figure()
ax = fig.add_subplot(111)

fig.set(facecolor = 'green')
ax.set(facecolor = 'red')

plt.show()
```

Пример 3. Установим некоторые параметры области *Axes*

```
fig = plt.figure()
ax = fig.add_subplot(111) # We'll explain the "111" later.
                             Basically, 1 row and 1 column.

fig.set_facecolor('green')

ax.set_facecolor('red')
ax.set_xlim([-10, 10])
ax.set_ylim([-2, 2])
ax.set_title('Основы анатомии matplotlib')
ax.set_xlabel('ось абсцисс (X Axis)')
ax.set_ylabel('ось ординат (Y Axis)')

plt.show()
```

Пример 4. То же самое выполним более коротким способом:

```
fig = plt.figure()
ax = fig.add_subplot(111)

fig.set(facecolor = 'green')
ax.set(facecolor = 'red',
       xlim = [-10, 10],
       ylim = [-2, 2],
       title = 'Основы анатомии matplotlib',
       xlabel = 'ось абсцисс (X Axis)',
```

```

        ylabel = 'ось ординат (Y Axis)')

plt.show()

```

Пример 5. Изменим цвет и размер `title` двумя способами:

```

# Способ №1:
fig = plt.figure()
ax = fig.add_subplot(111)

fig.set(facecolor = 'green')
ax.set_title('Основы анатомии matplotlib', color = 'white', size
= 20)
ax.set(xlim = [-10, 10], ylim = [-2, 2], xlabel = 'ось абцис (X
Axis)',
        ylabel = 'ось ординат (Y Axis)')
plt.show()

```

Пример 6

```

# Способ №2:
fig = plt.figure()
ax = fig.add_subplot(111)

fig.set(facecolor = 'green')
ax.set_title('Основы анатомии matplotlib')
ax.title.set_color('white')
ax.title.set_size(20)

ax.set(xlim = [-10, 10], ylim = [-2, 2], xlabel = 'ось абцис (X
Axis)',
        ylabel = 'ось ординат (Y Axis)')
plt.show()

```

2. Отображение данных на графике

Пример 7. Методы `plot` и `scatter`

```

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([0, 1, 2, 3, 4], [0, 6, 7, 15, 19])
ax.scatter([0, 1, 2, 3, 4], [1, 3, 8, 12, 27])

plt.show()

```

Пример 8

Изменим параметры внешнего вида графиков:

```

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([0, 1, 2, 3, 4], [0, 6, 7, 15, 19], color = 'black',
linewidth = 5)
ax.scatter([0, 1, 2, 3, 4], [1, 3, 8, 12, 27], color = 'blue',
marker = '*')
plt.show()

```

Пример 9. Оформим предыдущий код в более читаемый вид:

```
x = [0, 1, 2, 3, 4]
y_1 = [0, 6, 7, 15, 19]
y_2 = [1, 3, 8, 12, 27]

fig = plt.figure()
ax = fig.add_subplot(111)

ax.plot(x, y_1,
        color = 'black',
        linewidth = 5)

ax.scatter(x, y_2,
           color = 'blue',
           marker = '*')

plt.show()
```

3. Методы Axes и pyplot

Пример 10. Модуль `pyplot` создает *Figure* и *Axes* автоматически. Перепишем пример 9 следующим образом:

```
plt.plot([0, 1, 2, 3, 4], [0, 6, 7, 15, 19], linewidth = 3)
plt.scatter([0, 1, 2, 3, 4], [1, 3, 8, 12, 27], color =
'orange')

plt.show()
```

4. Несколько Axes на одной Figure

Пример 11. Разместим несколько областей *Axes* на одной *Figure*

```
fig = plt.figure()

ax_1 = fig.add_subplot(2, 2, 1)
ax_2 = fig.add_subplot(2, 2, 2)
ax_3 = fig.add_subplot(2, 2, 3)
ax_4 = fig.add_subplot(2, 2, 4)

ax_1.set(title = 'ax_1', xticks=[], yticks=[])
ax_2.set(title = 'ax_2', xticks=[], yticks=[])
ax_3.set(title = 'ax_3', xticks=[], yticks=[])
ax_4.set(title = 'ax_4', xticks=[], yticks=[])

plt.show()
```

Пример 12

Метод `add_subplot()` разбивает *Figure* на указанное количество строк и столбцов. Для этого `add_subplot()` необходимо всего три числа, которые мы и передаем ему в качестве параметров:

- первое - количество строк;
- второе - количество столбцов
- третье - индекс ячейки.

Индексирование полученных ячеек начинается с левого верхнего угла, выполняется построчно слева-направо и заканчивается в правом нижнем углу:

```
fig = plt.figure()

ax_1 = fig.add_subplot(3, 2, 1)
ax_2 = fig.add_subplot(3, 2, 2)
ax_3 = fig.add_subplot(3, 2, 3)
ax_4 = fig.add_subplot(3, 2, 4)
ax_5 = fig.add_subplot(3, 2, 5)
ax_6 = fig.add_subplot(3, 2, 6)

ax_1.set(title = 'ax_1', xticks=[], yticks=[])
ax_2.set(title = 'ax_2', xticks=[], yticks=[])
ax_3.set(title = 'ax_3', xticks=[], yticks=[])
ax_4.set(title = 'ax_4', xticks=[], yticks=[])
ax_5.set(title = 'ax_5', xticks=[], yticks=[])
ax_6.set(title = 'ax_6', xticks=[], yticks=[])

plt.show()
```

Пример 13. Пример заполнения областями Axes области *Figure*:

```
fig = plt.figure()

ax_1 = fig.add_subplot(3, 2, 1)
ax_2 = fig.add_subplot(3, 2, 4)
ax_3 = fig.add_subplot(3, 2, 5)

ax_1.set(title = 'ax_1', xticks=[], yticks=[])
ax_2.set(title = 'ax_2', xticks=[], yticks=[])
ax_3.set(title = 'ax_3', xticks=[], yticks=[])

plt.show()
```

Пример 14. Пример заполнения областями Axes области *Figure*::

```
fig = plt.figure()

ax_1 = fig.add_subplot(3, 1, 1)
ax_2 = fig.add_subplot(3, 2, 4)
ax_3 = fig.add_subplot(3, 3, 9)

ax_1.set(title = 'ax_1', xticks=[], yticks=[])
ax_2.set(title = 'ax_2', xticks=[], yticks=[])
ax_3.set(title = 'ax_3', xticks=[], yticks=[])

plt.show()
```

Пример 15. Области Axes могут перекрывать друг друга, быть разного размера, разделенными некоторым пространством и размещаться в произвольных местах:

```
fig = plt.figure()

ax_1 = fig.add_subplot(3, 1, 1)
ax_2 = fig.add_subplot(6, 3, 3)
ax_3 = fig.add_subplot(3, 3, 4)
ax_4 = fig.add_subplot(3, 3, 6)
ax_5 = fig.add_subplot(3, 4, 10)
ax_6 = fig.add_subplot(5, 5, 25)

ax_1.set(title = 'ax_1', xticks=[], yticks=[])
ax_2.set(title = 'ax_2', xticks=[], yticks=[])
ax_3.set(title = 'ax_3', xticks=[], yticks=[])
ax_4.set(title = 'ax_4', xticks=[], yticks=[])
ax_5.set(title = 'ax_5', xticks=[], yticks=[])
ax_6.set(title = 'ax_6', xticks=[], yticks=[])

plt.show()
```

5. Subplots()

Пример 16. Использование функции `plt.subplots(nrows, ncols)`

```
fig, axes = plt.subplots(nrows = 2, ncols = 2 )

axes[0,0].set(title='axes[0,0]')
axes[0,1].set(title='axes[0,1]')
axes[1,0].set(title='axes[1,0]')
axes[1,1].set(title='axes[1,1]')

for ax in axes.flat:
    ax.set(xticks=[], yticks=[])

plt.show()
```

Пример 17. Нанесение сетки на области Axes

```
print(plt.subplots(nrows = 2, ncols = 2))
```

При выводе `plt.subplots(nrows, ncols)` создает кортеж из двух элементов:

- Область *Figure*;
- Массив объектов NumPy, состоящий из двух строк и двух столбцов. Каждый элемент этого массива представляет собой отдельную область Axes, к которой можно обратиться по ее индексу в данном массиве.

Пример 18. Для дальнейшей работы с данными областями нам необходимо распаковать данный кортеж:

```
fig, axes = plt.subplots(nrows = 2, ncols = 2)
```

Теперь `fig` – это *Figure*, а `axes` – это массив NumPy, элементами которого являются объекты Axes. Далее, мы решили

Пример 19. Установим каждой области Axes свой заголовок:

```
axes[0, 0].set(title='axes[0, 0]')
axes[0, 1].set(title='axes[0, 1]')
axes[1, 0].set(title='axes[1, 0]')
axes[1, 1].set(title='axes[1, 1]')
```

Пример 20. Установим для каждой области одинаковые параметры в цикле:

```
for ax in axes.flat:
    ax.set(xticks=[], yticks=[])
```

Пример 21. Установим заголовки в том же самом цикле:

```
fig, axes = plt.subplots(nrows=3, ncols=3)

n = 1

for ax in axes.flat:
    ax.set(title='axes_' + str(n), xticks=[], yticks=[])
    n += 1
plt.show()
```

Пример 22. По умолчанию количество строк и столбцов в методе `subplots` равно 1, что удобно для быстрого создания *Figure* с одной областью Axes:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()      # одна строка вместо двух:
                              # fig = plt.figure()
                              # ax = fig.add_subplot(111)
ax.set(title='Axes')

plt.show()
```

Заключение. Мы рассмотрели базовые компоненты matplotlib.

Figure – это контейнер самого верхнего уровня, та область на которой все нарисовано. Таких областей может быть несколько, каждая из которых может содержать несколько контейнеров Axes.

Axes – это та область на которой чаще всего и отражаются графики (данные в виде графиков), а так же все вспомогательные атрибуты (линии сетки, метки, указатели и т.д.). Часто, установка этой области сопровождается с вызовом *subplot*, который и помещает Axes на регулярную сетку. Поэтому Axes и Subplot можно считать синонимами.

Каждая область Axes содержит XAxis и YAxis. Они содержат деления, метки и другие вспомогательные атрибуты.