

Отчет по лабораторной работе №2

Задание (Rare)

1. Создать в каталоге для данной ЛР в своем репозитории виртуальное окружение и установить в него `matplotlib` и `numpy`. Создать файл `requirements.txt`.
 2. Открыть книгу [1] и выполнить уроки 1-3.
 3. Выбрать одну из неразрывных функции своего варианта, построить график этой функции и касательную к ней. Добавить на график заголовок, подписи осей, легенду, сетку, а также аннотацию к точке касания.
 4. Добавить в корень своего репозитория файл `.gitignore` отсюда, перед тем как делать очередной коммит.
 5. Оформить отчёт в `README.md`.
 6. Склонировать этот репозиторий НЕ в свой репозиторий, а рядом. Изучить использование этого инструмента и создать pdf-версию своего отчёта из `README.md`. Добавить её в репозиторий.
-

Описание проделанной работы

1. Подготовка окружения

Для начала работы было создано виртуальное окружение и установлены необходимые библиотеки, а именно, `matplotlib` и `numpy`.

Команды:

```
#  
python -m venv venv  
#  
venv\Scripts\activate  
#  
pip install matplotlib (numpy)  
#  
pip freeze > requirements.txt
```

2. Выполнение уроков 1-3 (из книги [1])

В процессе выполнения уроков были построены базовые графики для освоения функционала `matplotlib`.

Урок 1: Базовые графики

В первом уроке были изучены основы построения графиков с использованием библиотеки `matplotlib`. Целью урока было освоение базового синтаксиса, построение различных типов графиков и знакомство с основными элементами визуализации.

1.1. Построение простого линейного графика

Первым шагом был построен простейший график путем прямой передачи списков координат точек.

Figure 1

— □ ×

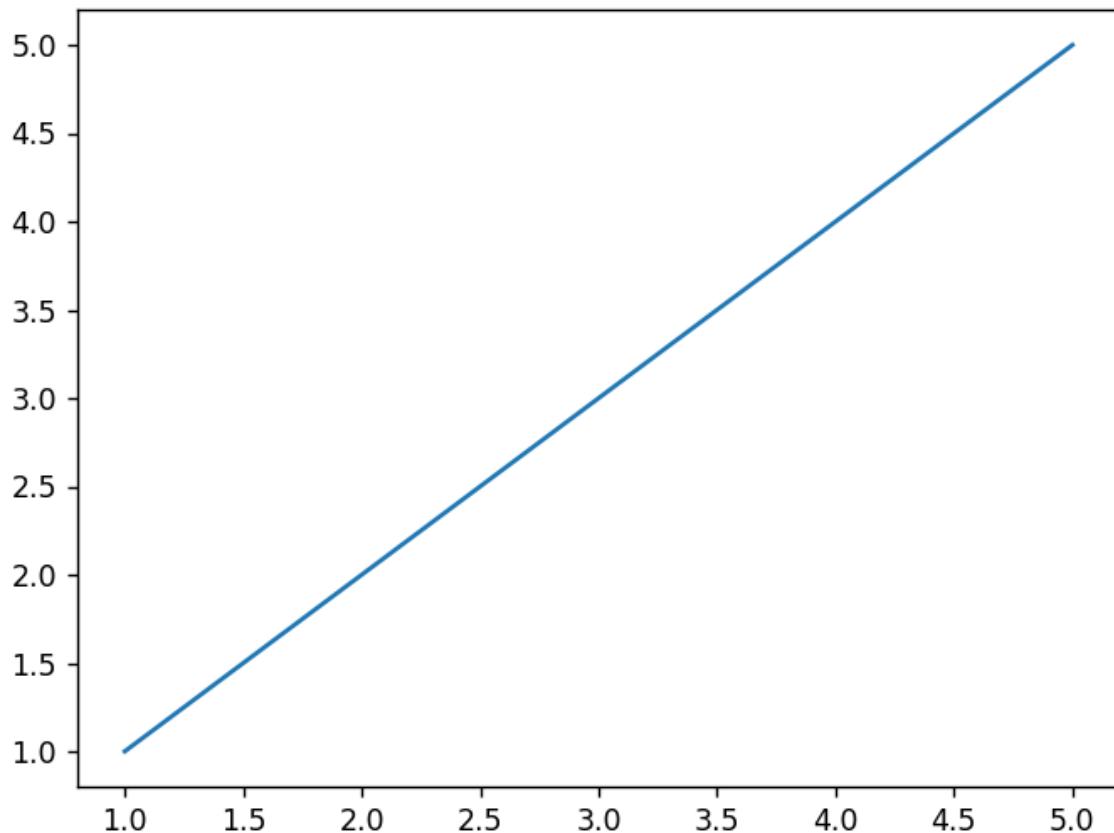


Figure 1: img.png

1.2. Построение линейного графика с настройками

Далее был построен более информативный график линейной зависимости $y = x$ с использованием библиотеки `matplotlib` для генерации данных. Были применены базовые настройки: добавлен заголовок, подписи осей и координатная сетка.

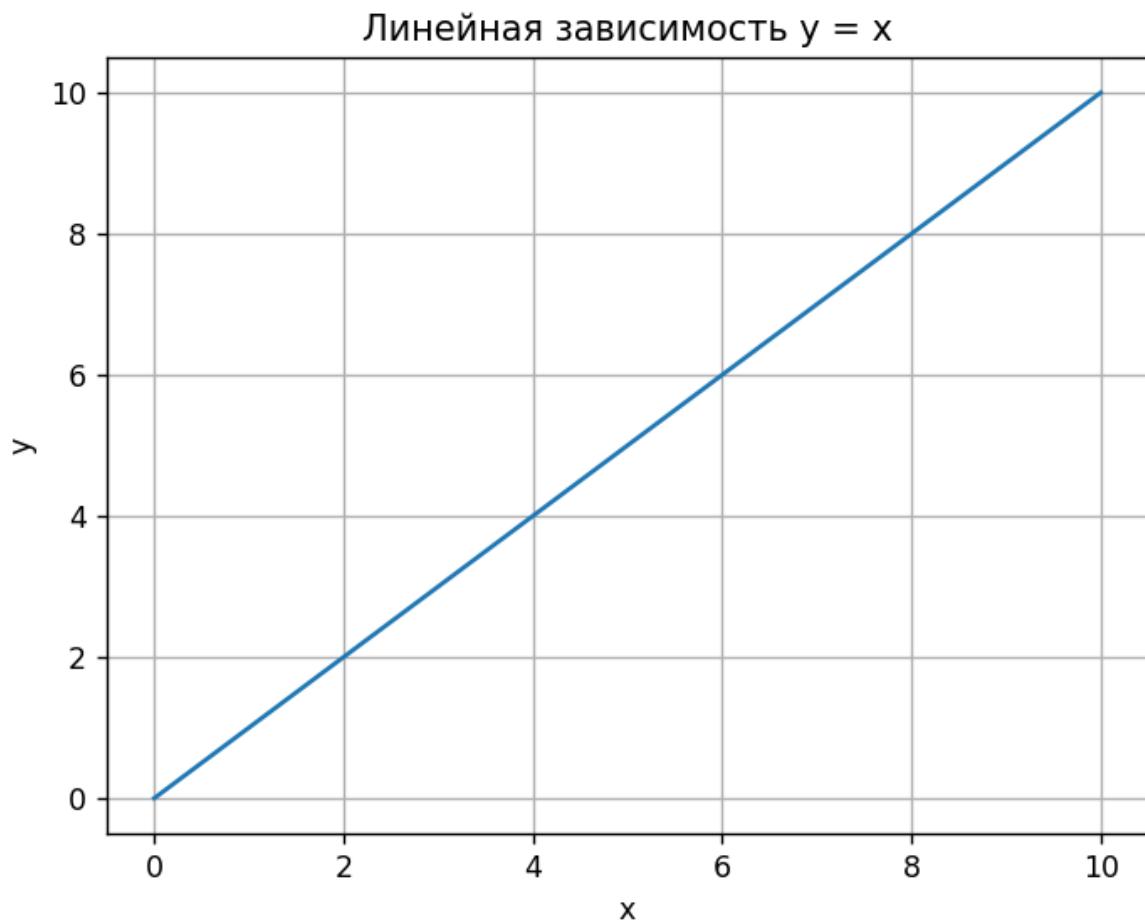


Figure 2: img_1.png

После этого был изменен стиль линии на красную пунктирную ('r-').

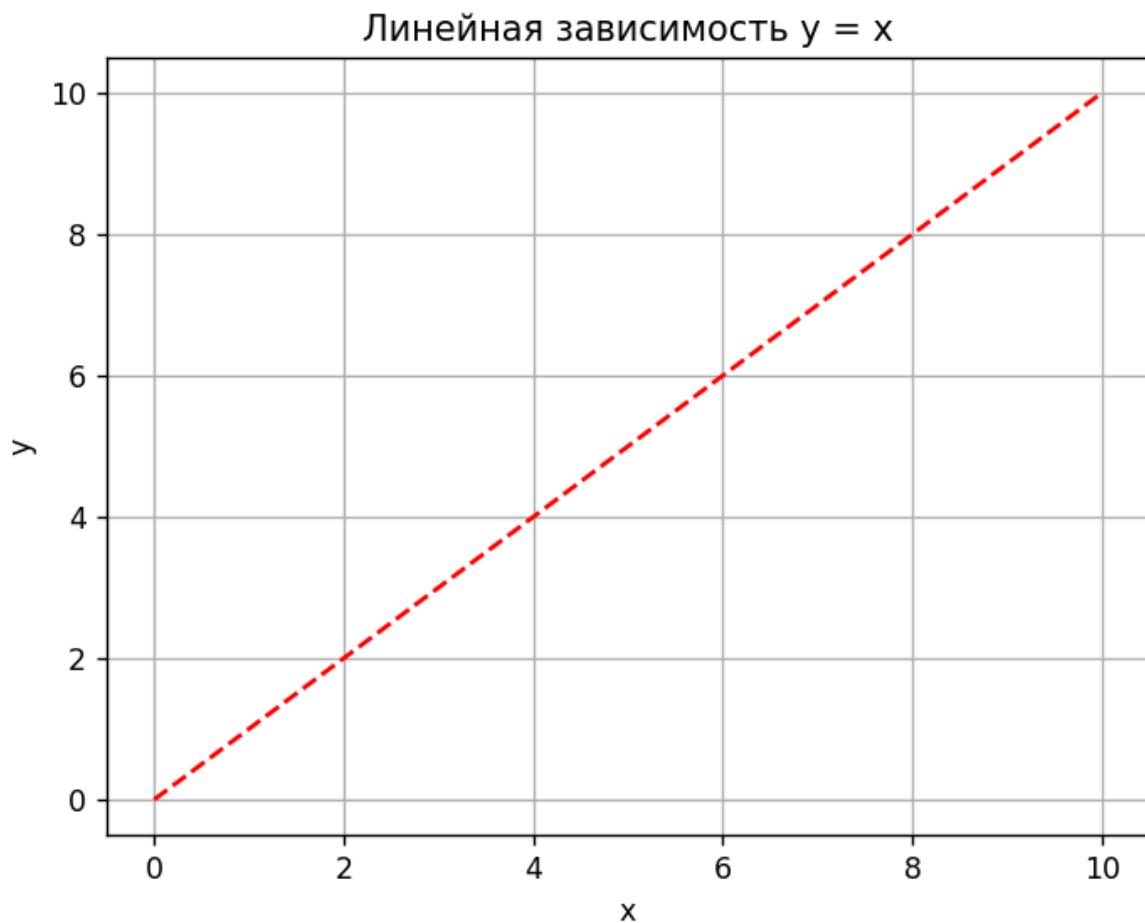


Figure 3: img_2.png

1.3. Построение нескольких графиков на одном поле

В этом разделе было показано, как отобразить две зависимости ($y_1 = x$ и $y_2 = x^2$) на одной координатной плоскости для их сравнения.

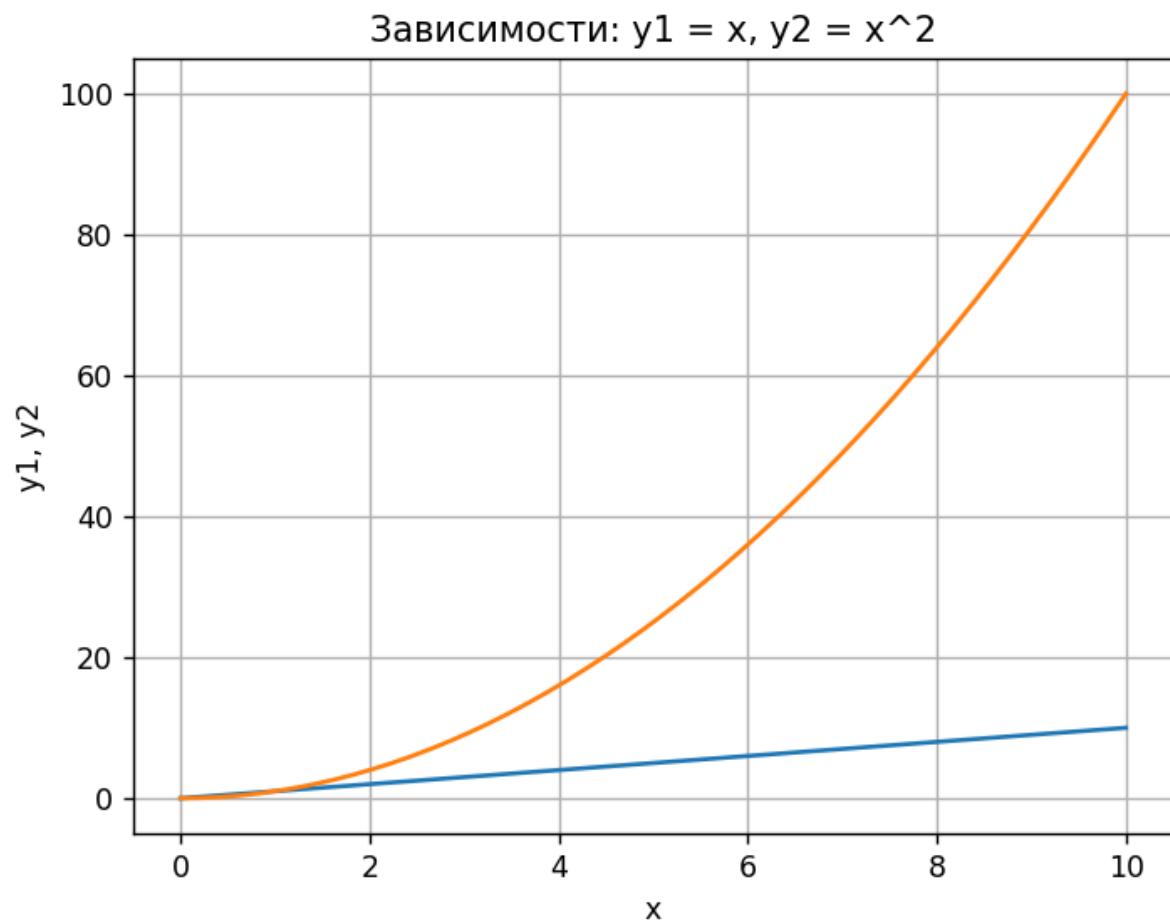
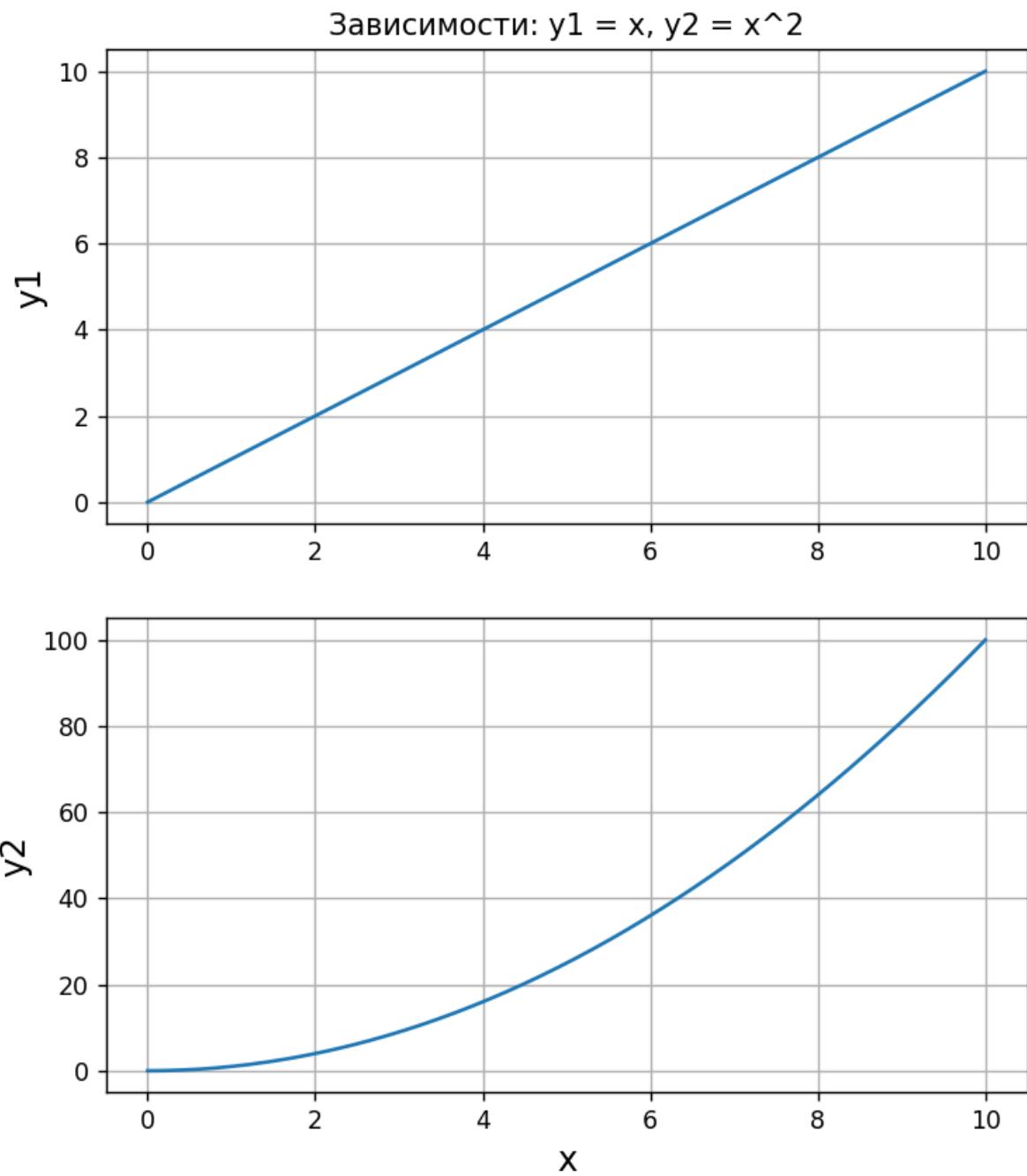


Figure 4: img_3.png

1.4. Представление графиков на разных полях (subplots)

Четвертая часть урока была посвящена размещению графиков на отдельных областях в пределах одного окна.



1.5. Построение столбчатой диаграммы

В завершение урока была рассмотрена работа с данными на примере столбчатой диаграммы, отображающей количество фруктов.

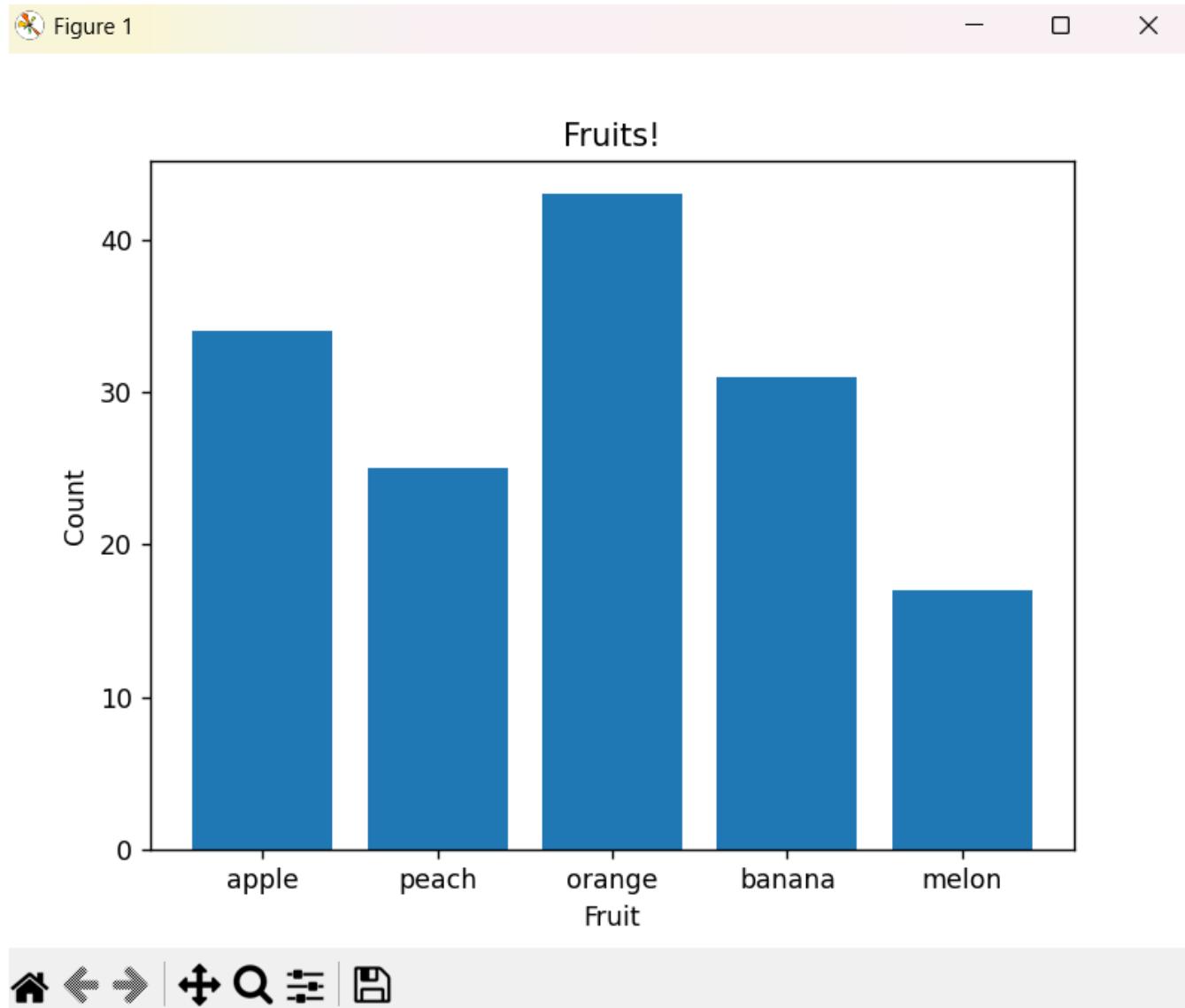


Figure 6: img_5.png

Урок 2. Основы работы модулем pyplot

Второй урок был посвящен изучению возможностей модуля pyplot. Были рассмотрены различные способы построения графиков, настройка текстовых элементов, работа со стилями линий и цветом, а также различные подходы к размещению нескольких графиков.

2.1. Построение графиков с помощью plot()

Были рассмотрены различные варианты вызова функции `plot()`:

Пустой график:

Figure 1

— □ ×

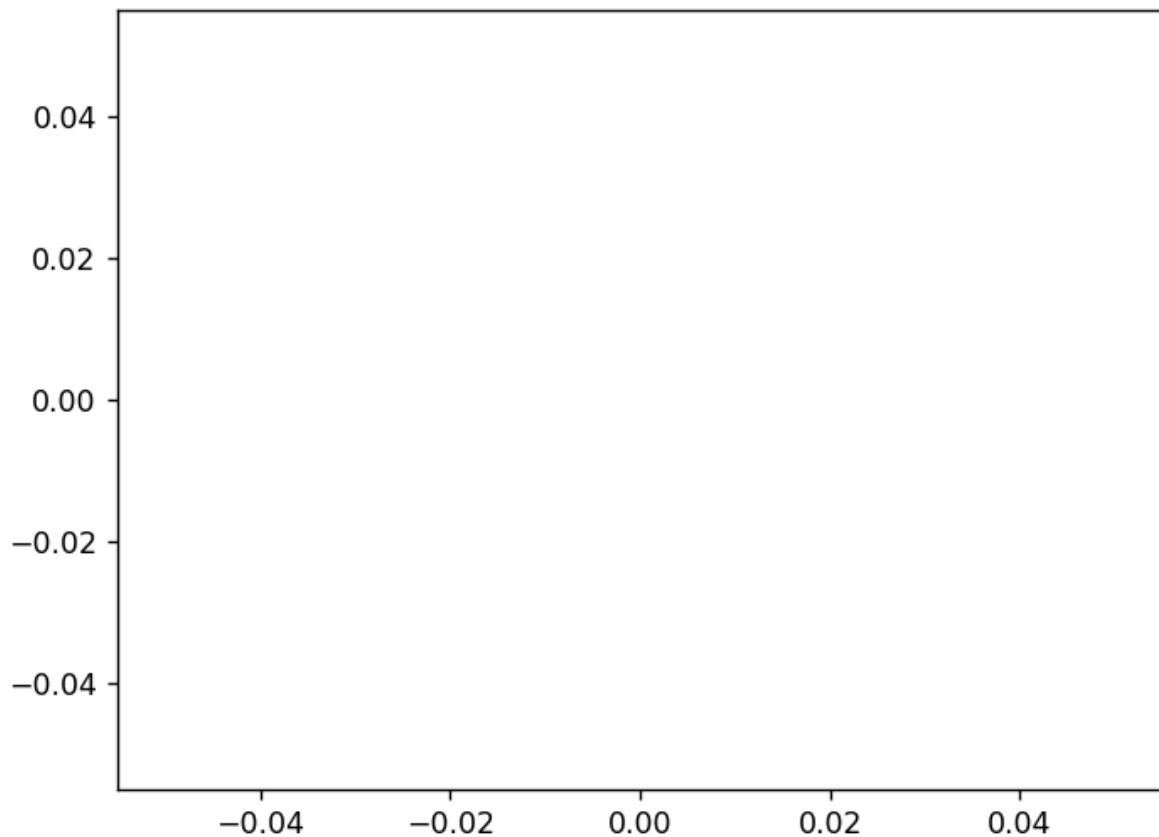


Figure 7: img_7.png

График только по значениям у:

Figure 1

— □ ×

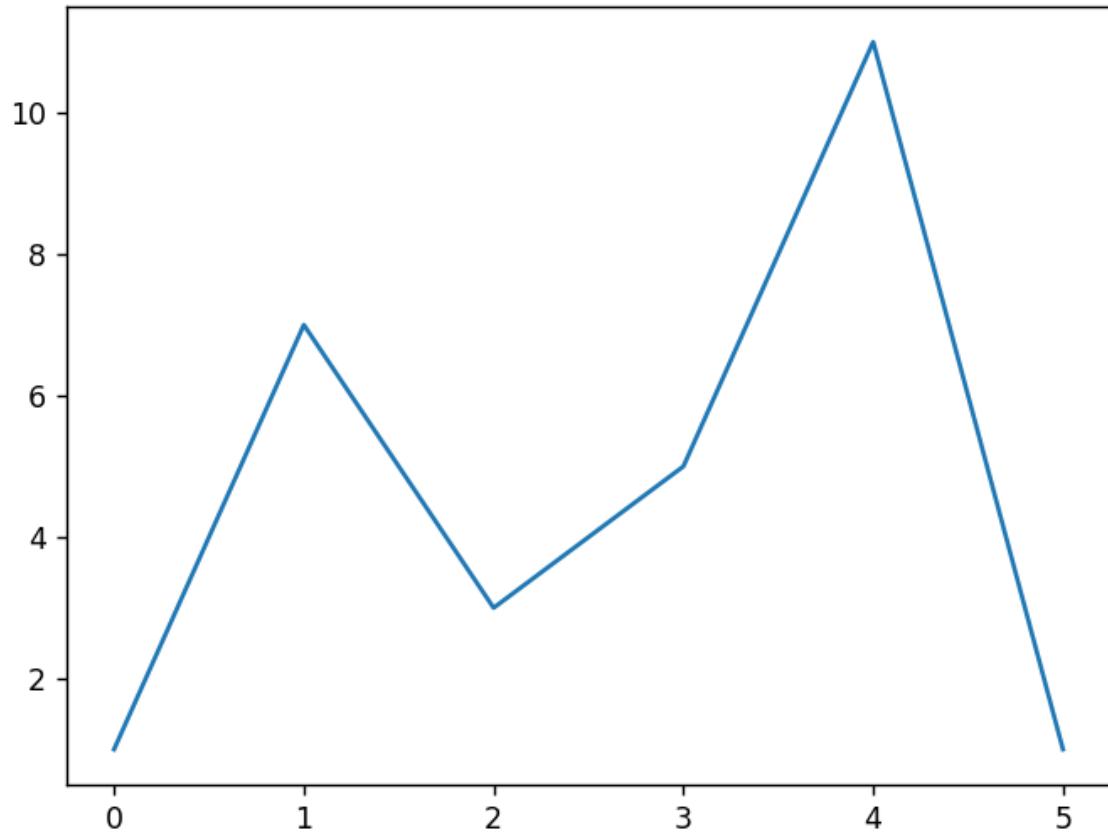


Figure 8: img_8.png

График с заданными x и у:

Figure 1

— □ ×

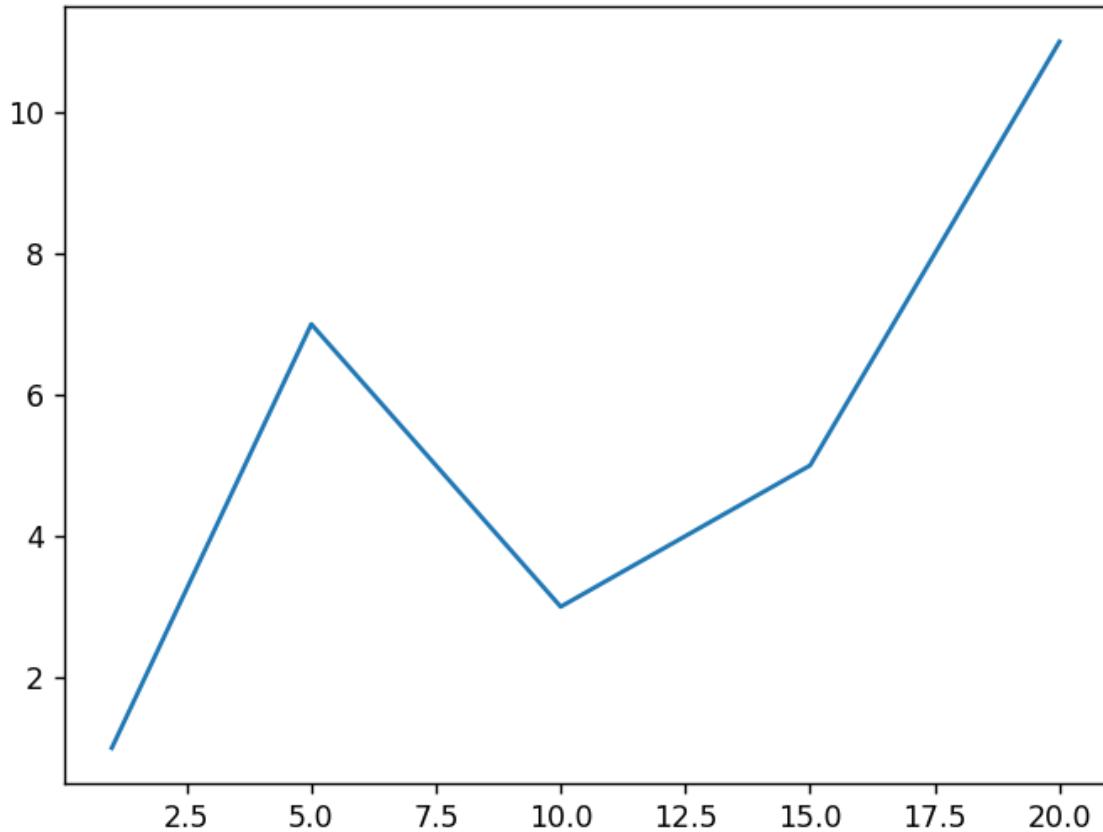


Figure 9: img_9.png

2.2. Текстовые надписи на графике

Были изучены функции для добавления текстовых элементов:

- xlabel() / ylabel() – подписи осей
- title() – заголовок графика
- text() – текстовое примечание в произвольном месте
- legend() – легенда
- grid() – включение сетки

Пример использования:

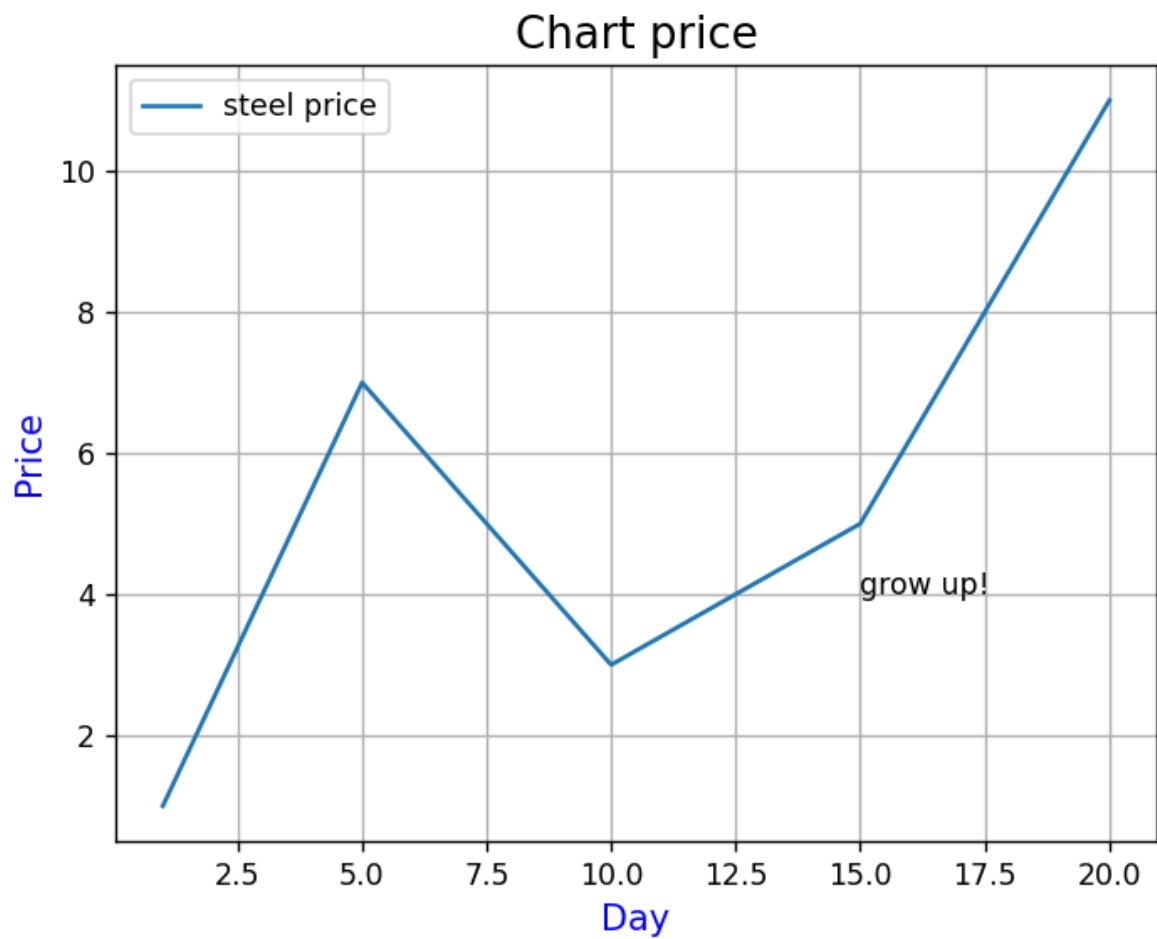


Figure 10: img_10.png

2.3. Работа с линейным графиком и цветом линий

Были изучены параметры настройки внешнего вида линий.

Пример с штриховой линией:

Figure 1

— □ ×

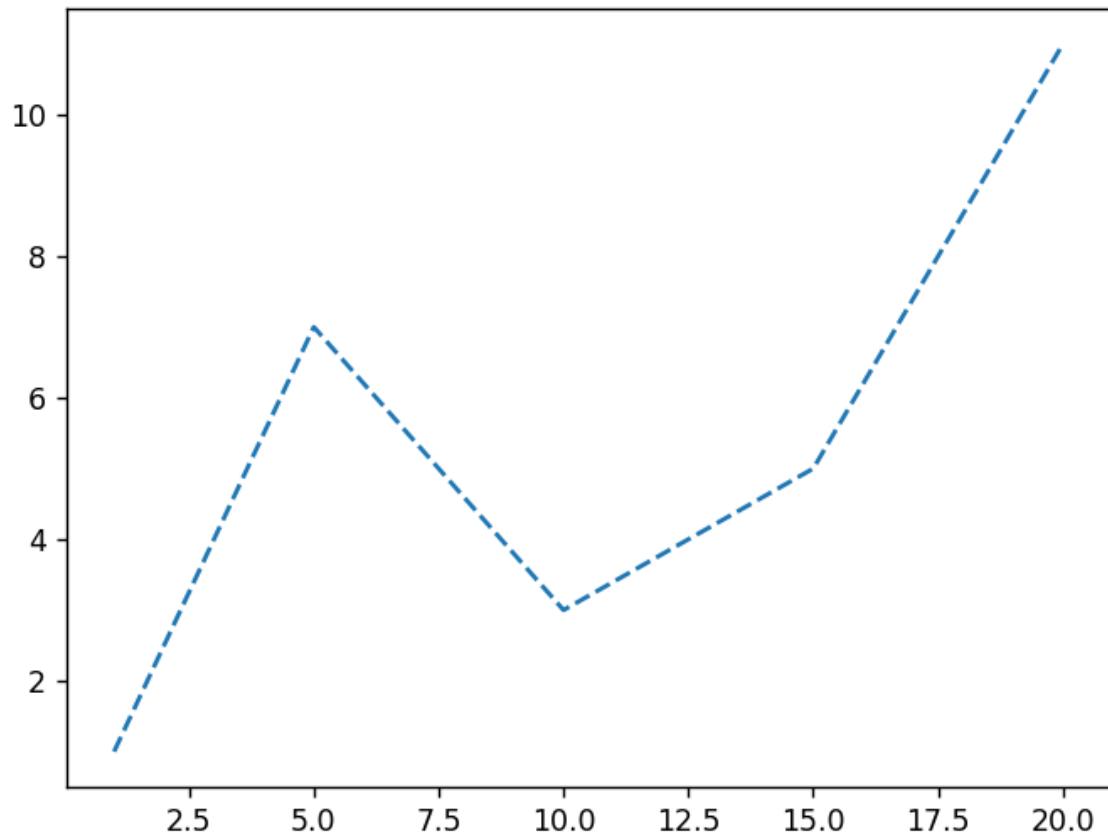


Figure 11: img_11.png

Несколько графиков с разными стилями:

Figure 1

— □ ×

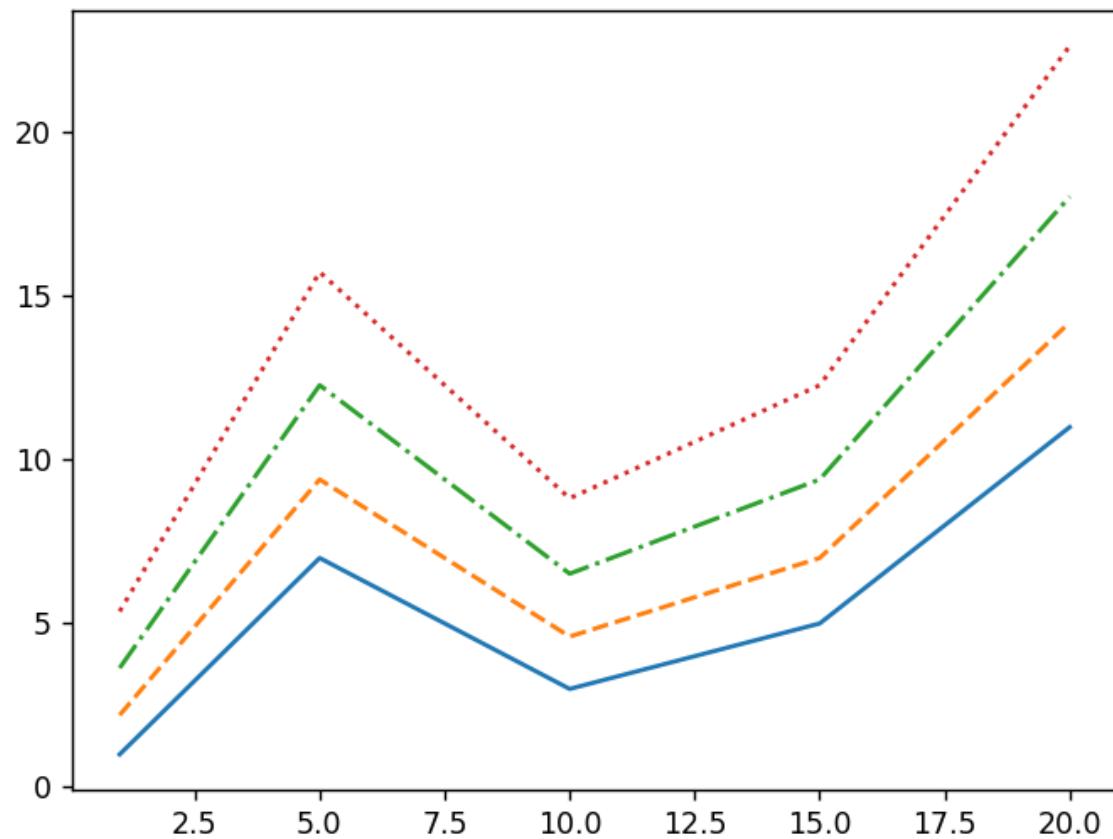


Figure 12: img_12.png

Пример с красной штриховой линией:

Figure 1

— □ ×

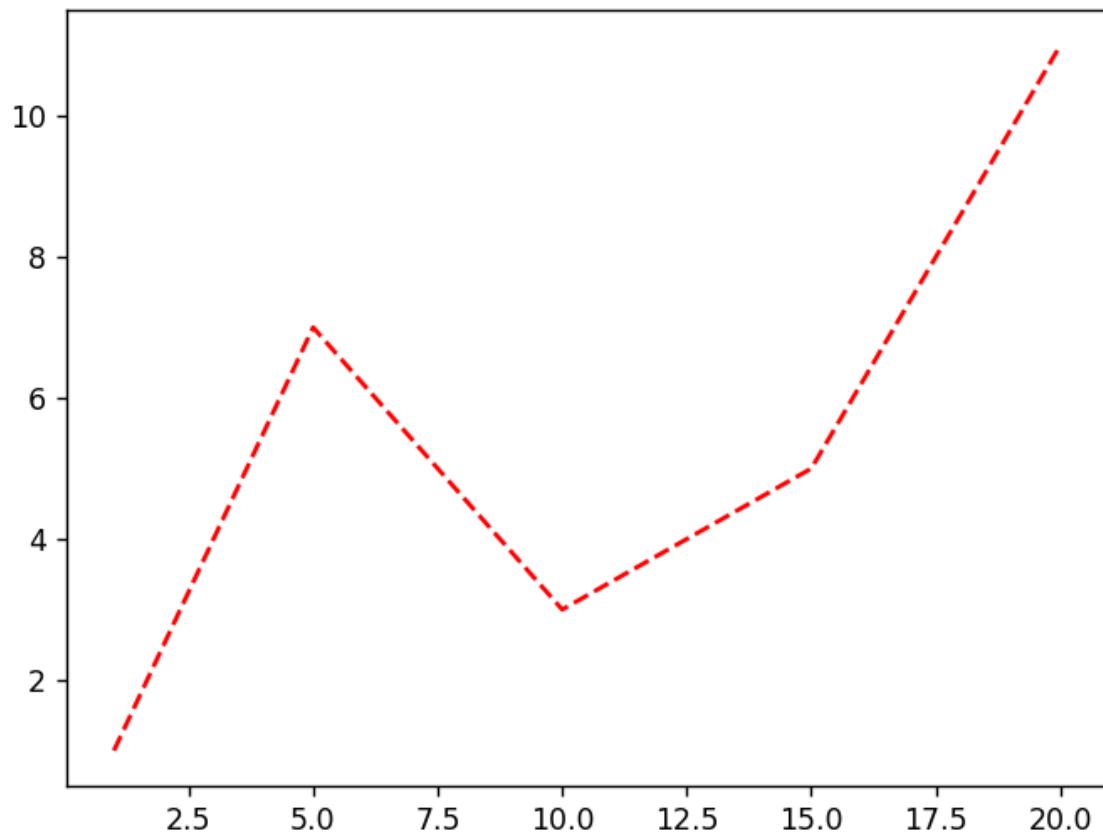


Figure 13: img_13.png

Красные кружки:

Figure 1

— □ ×

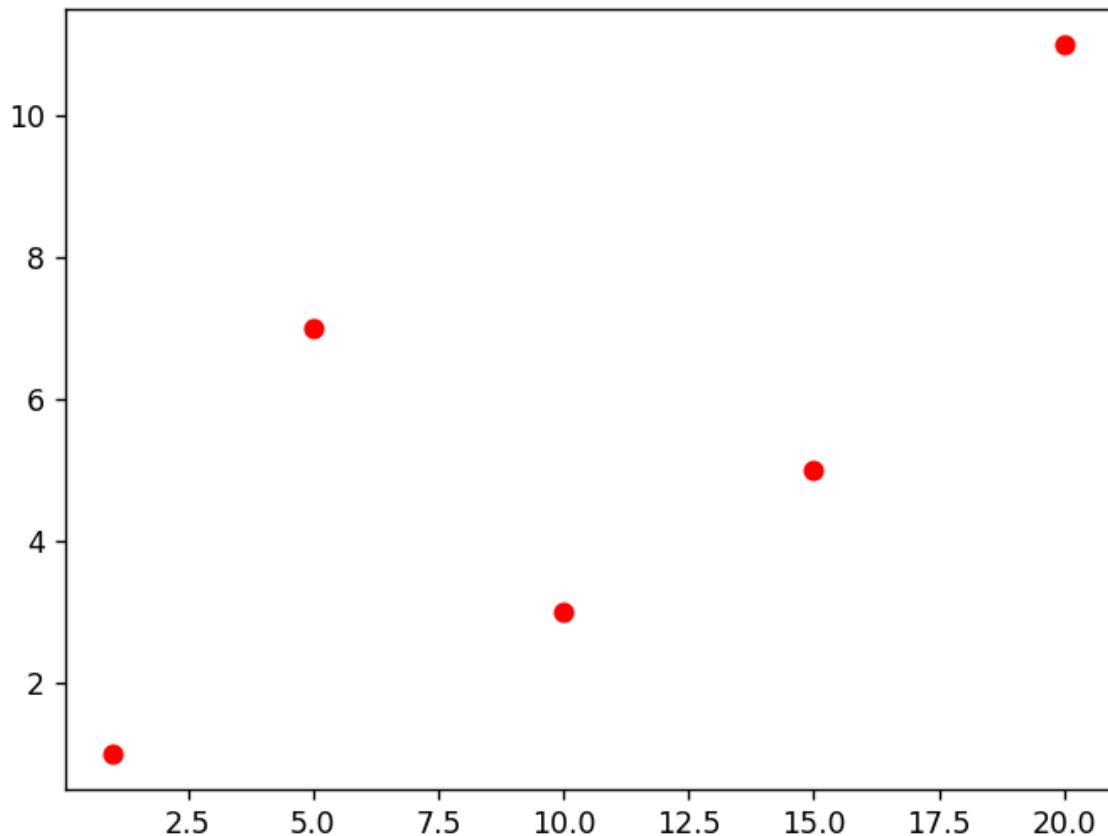


Figure 14: img_14.png

Синие крестики:

Figure 1

— □ ×

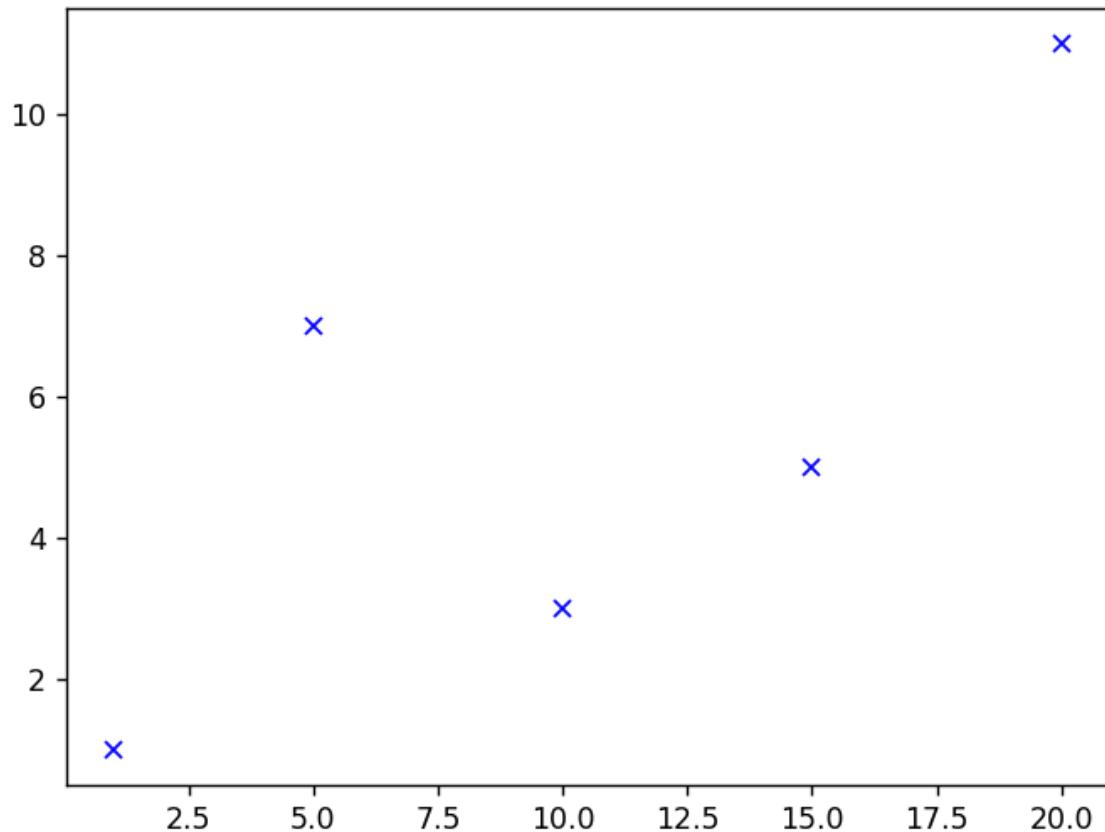


Figure 15: img_15.png

2.4. Размещение графиков на разных полях

Был изучен способ размещения нескольких графиков на разных полях с помощью функций `subplot()` и `subplots()`:

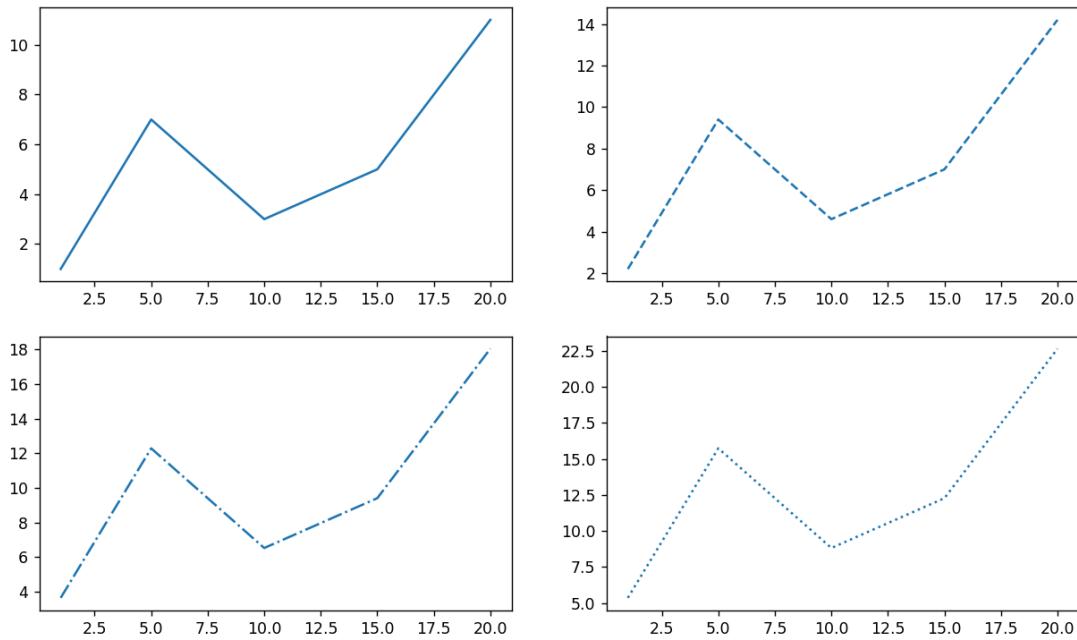


Figure 16: img_16.png

Урок 3. Настройка элементов графика

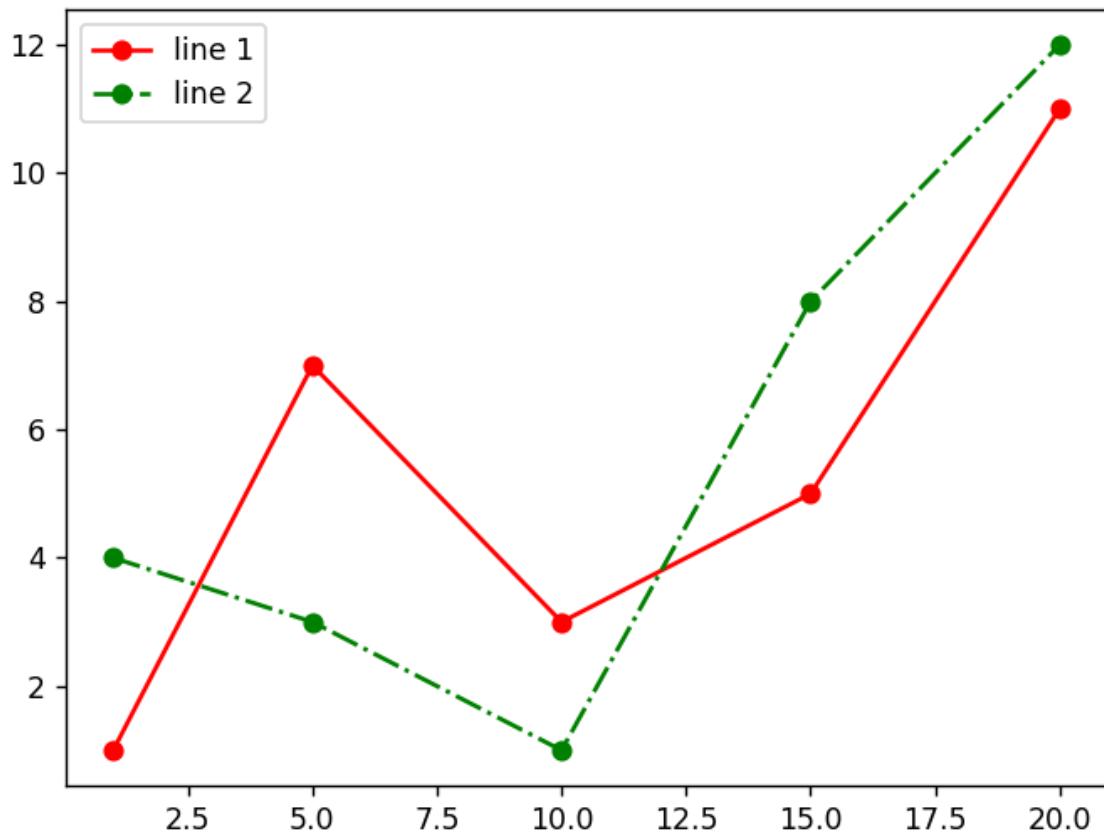
Третий урок посвящен настройке элементов графика: легенды, компоновке нескольких графиков, текстовым элементам и цветовой шкале.

3.1. Работа с легендой

Для отображения легенды используется функция `legend()`:

Figure 1

— □ ×



⌂ ← → | ⌂ Q ⌂ | ⌂

Figure 17: img_17.png

Можно вручную указать соответствие линий и меток:

Figure 1

— □ ×

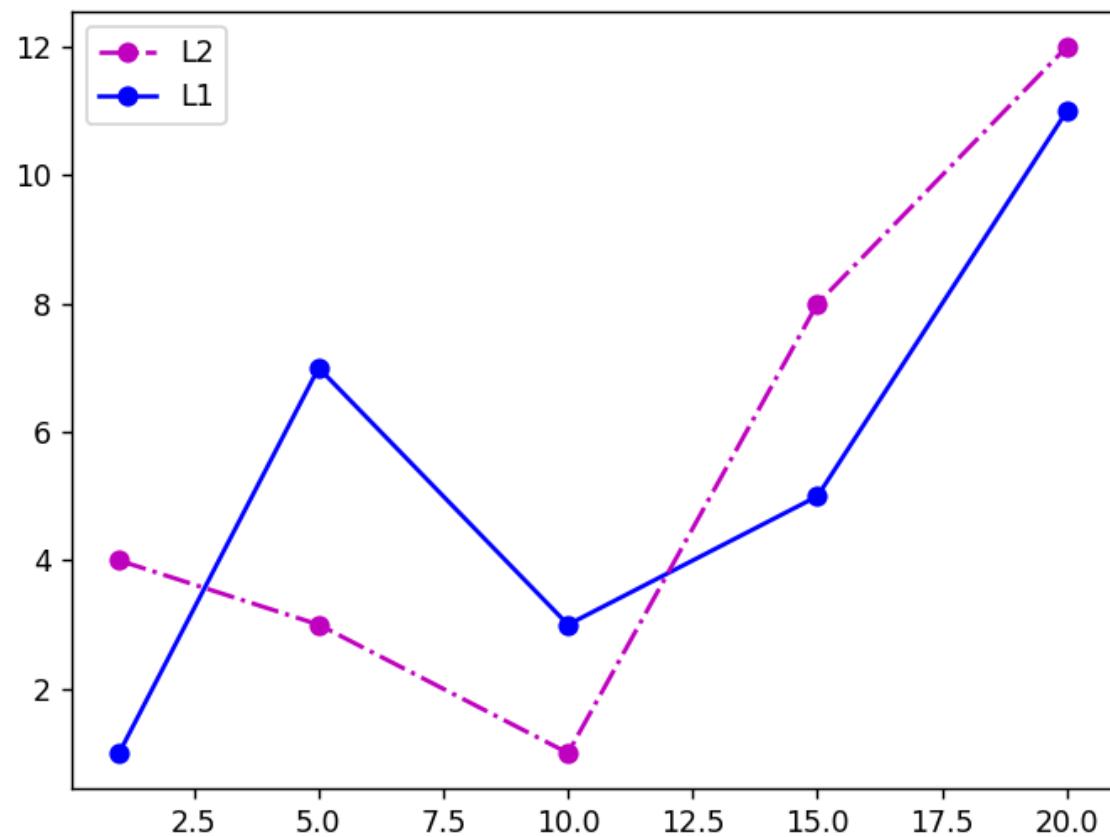
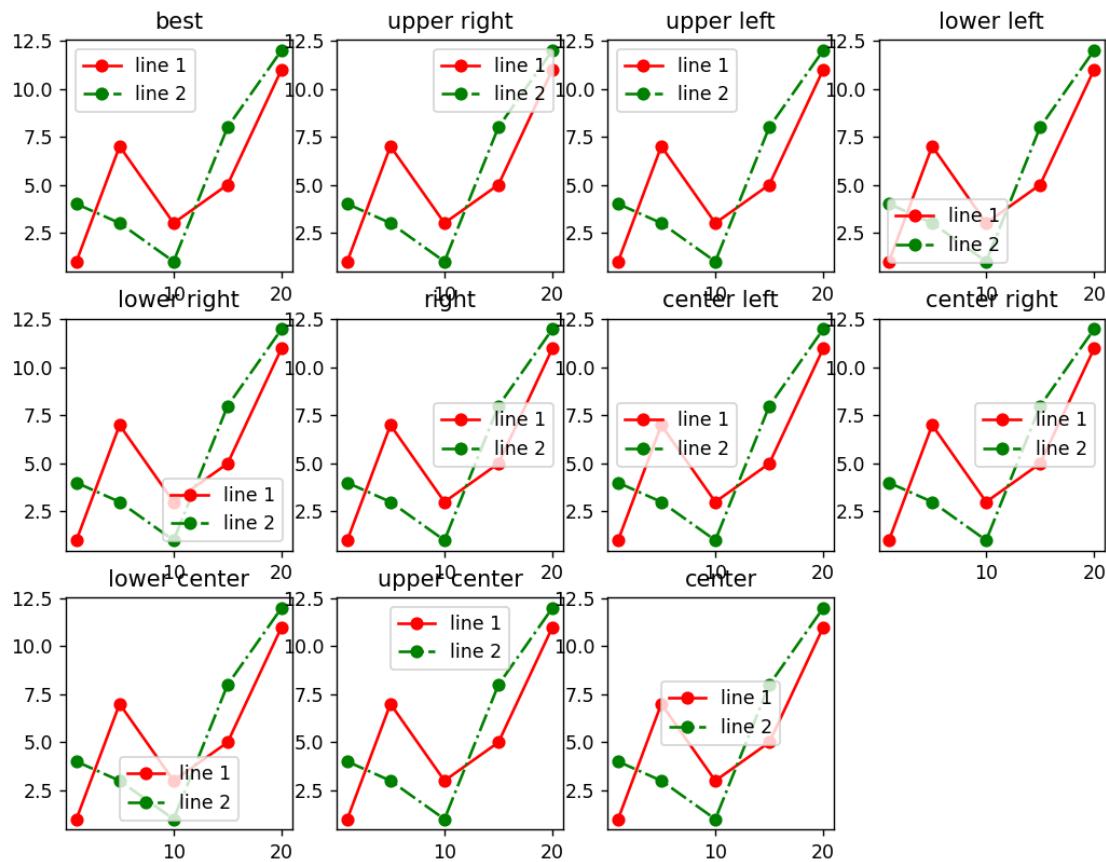


Figure 18: img_18.png

Параметр `loc` позволяет задать положение легенды:

Figure 1

- □ ×



⌂ ← → ⌂ Q ⌂ |

Figure 19: img_19.png

3.2. Компоновка графиков с GridSpec

Класс GridSpec позволяет создавать сложные компоновки с графиками различных размеров:

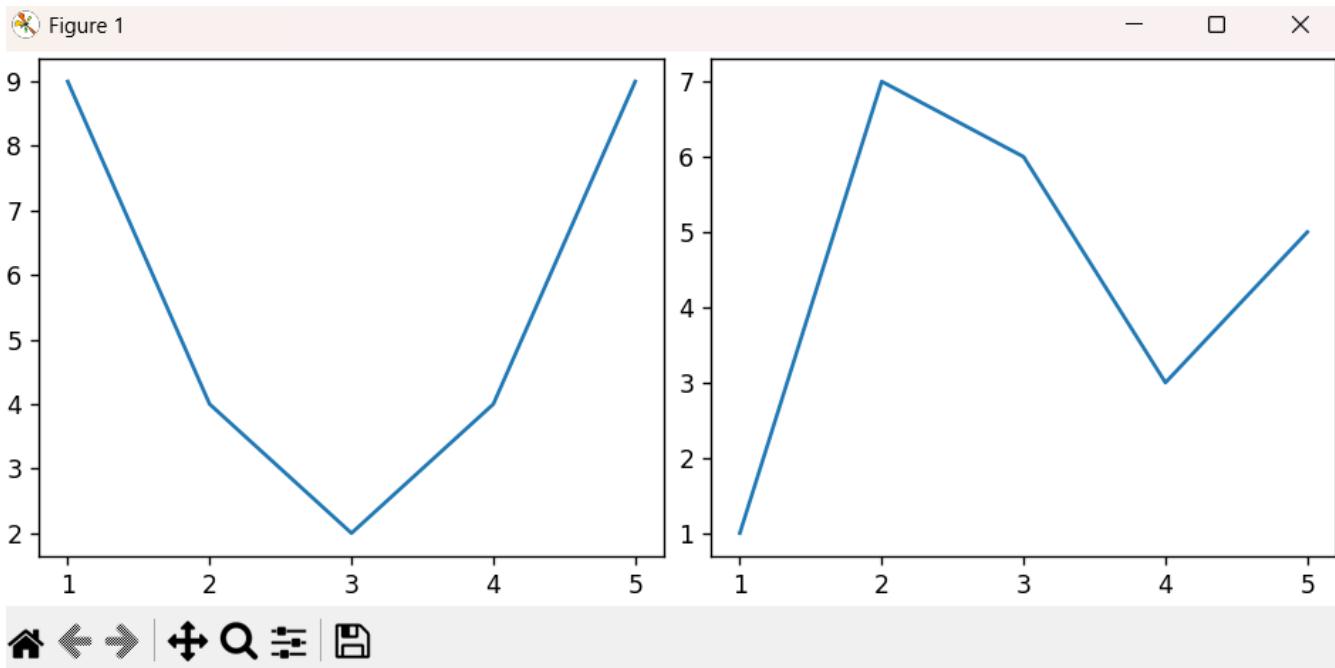


Figure 20: img_20.png

GridSpec позволяет создавать графики, занимающие несколько ячеек сетки:

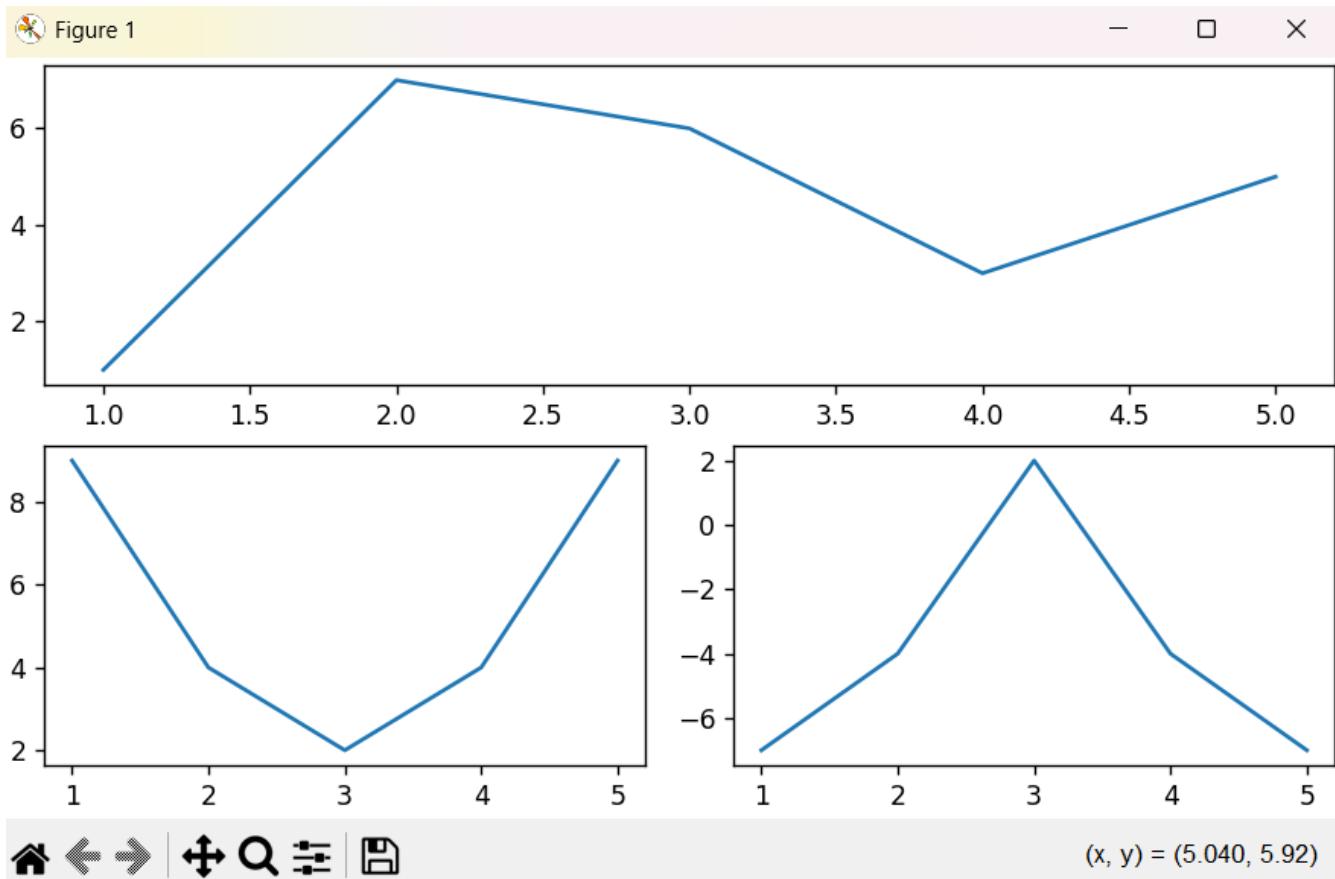


Figure 21: img_21.png

Можно задавать соотношения ширины и высоты областей:

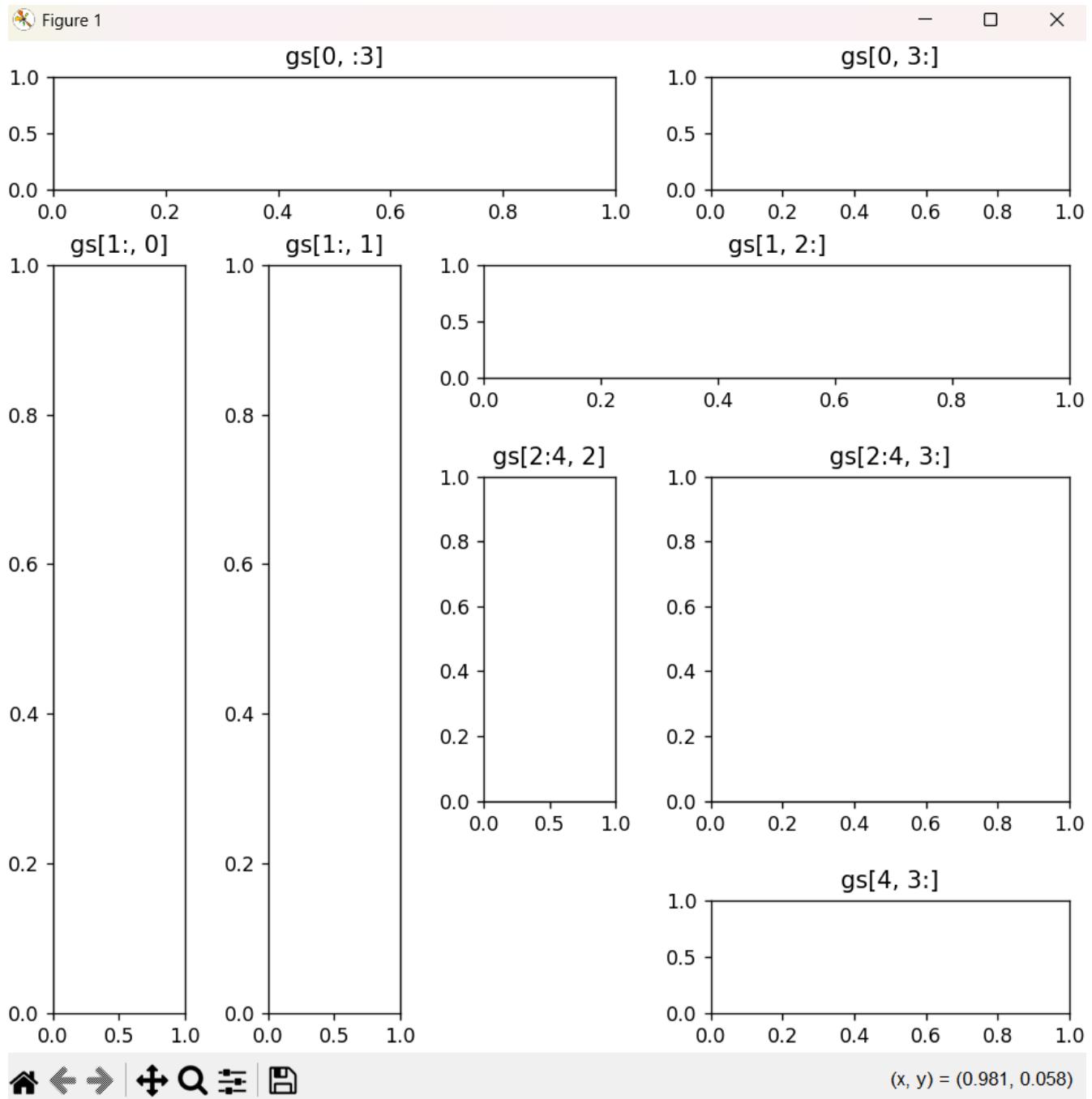


Figure 22: img_22.png

3.3. Текстовые элементы графика

В Matplotlib доступны различные текстовые элементы: заголовки, подписи осей, текстовые блоки и аннотации:

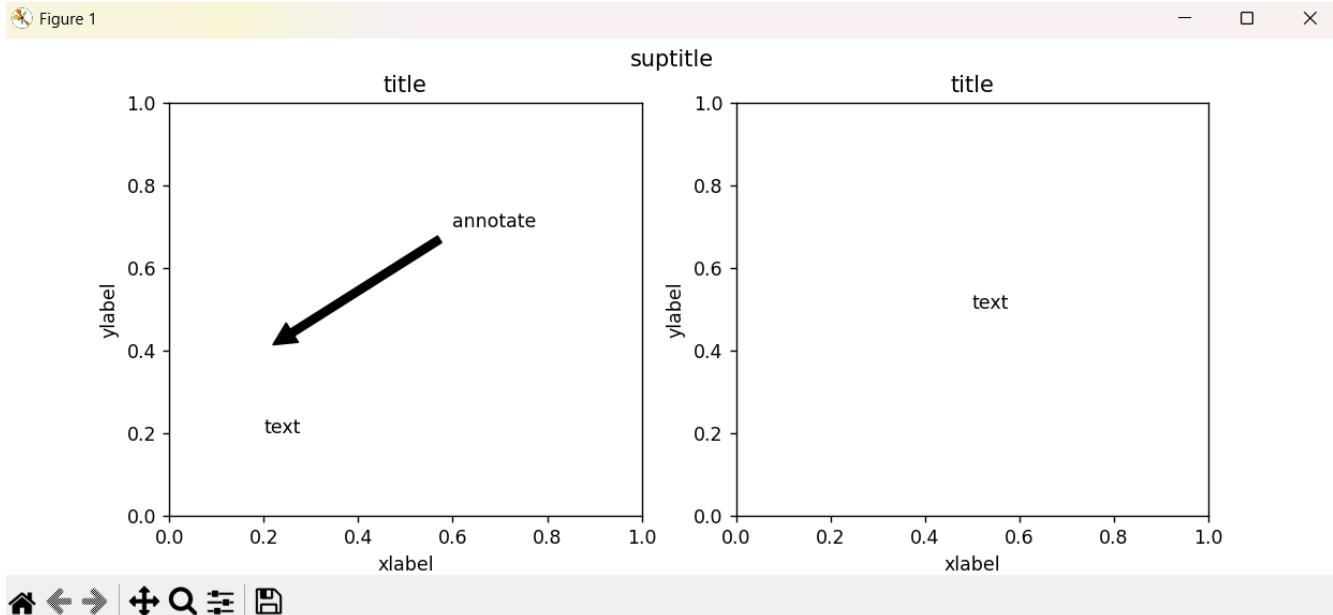


Figure 23: img_23.png

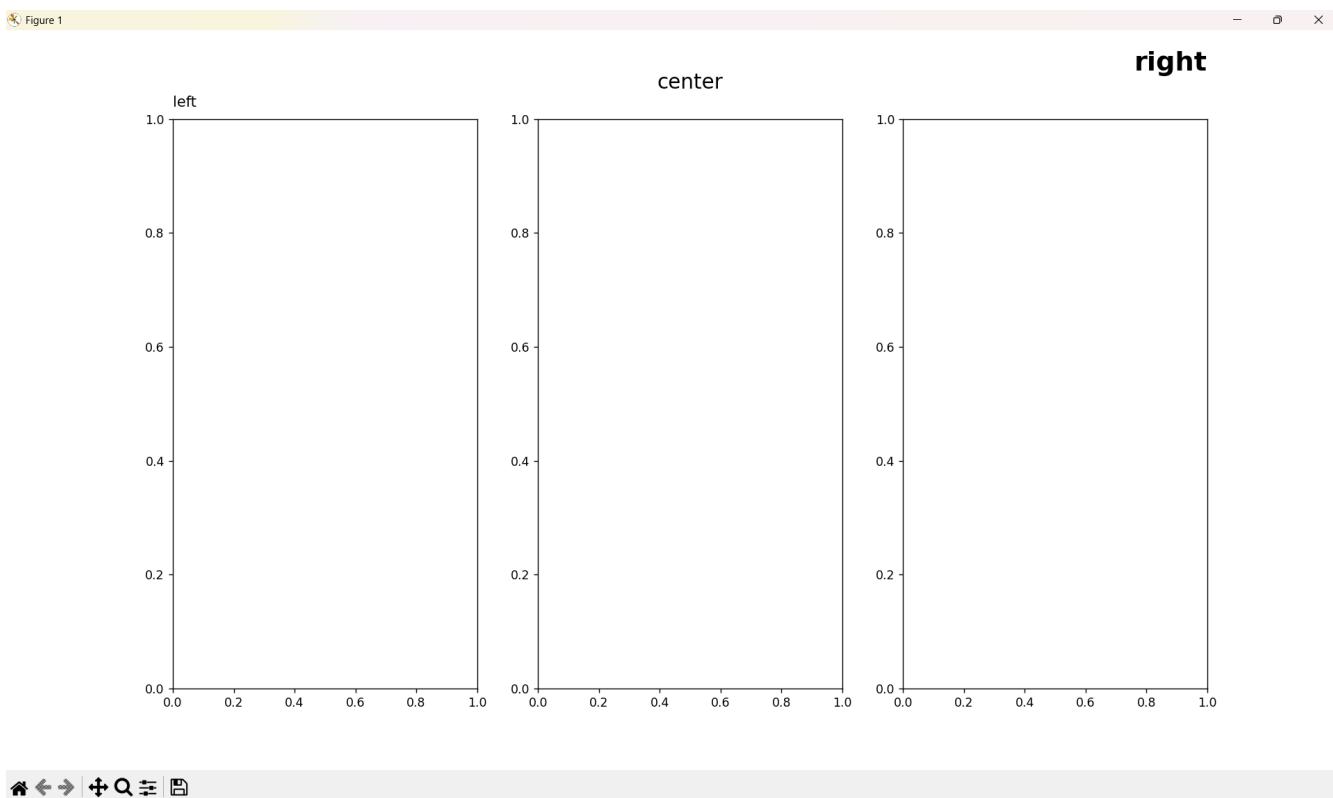


Figure 24: img_24.png

Figure 1

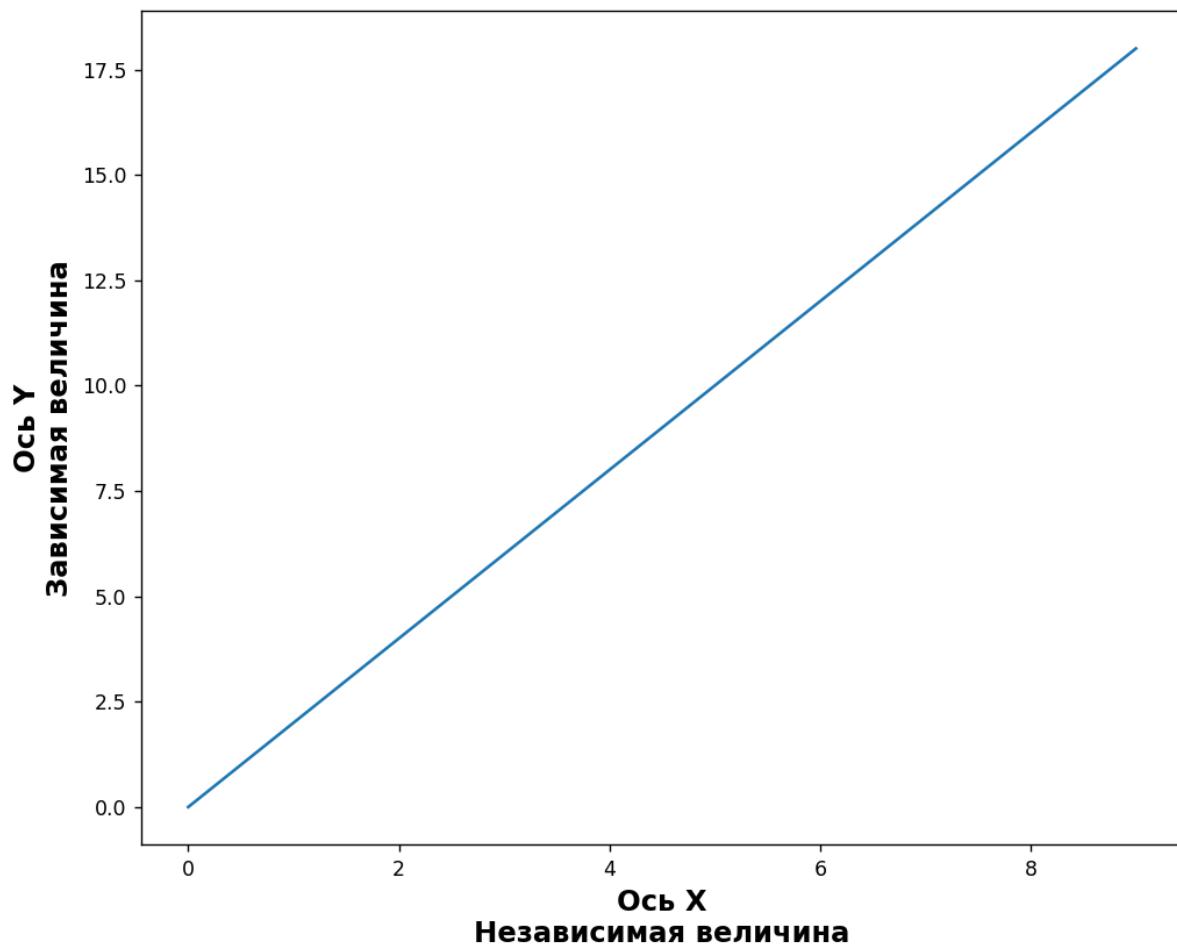


Figure 25: img_25.png

Figure 1

— □ ×

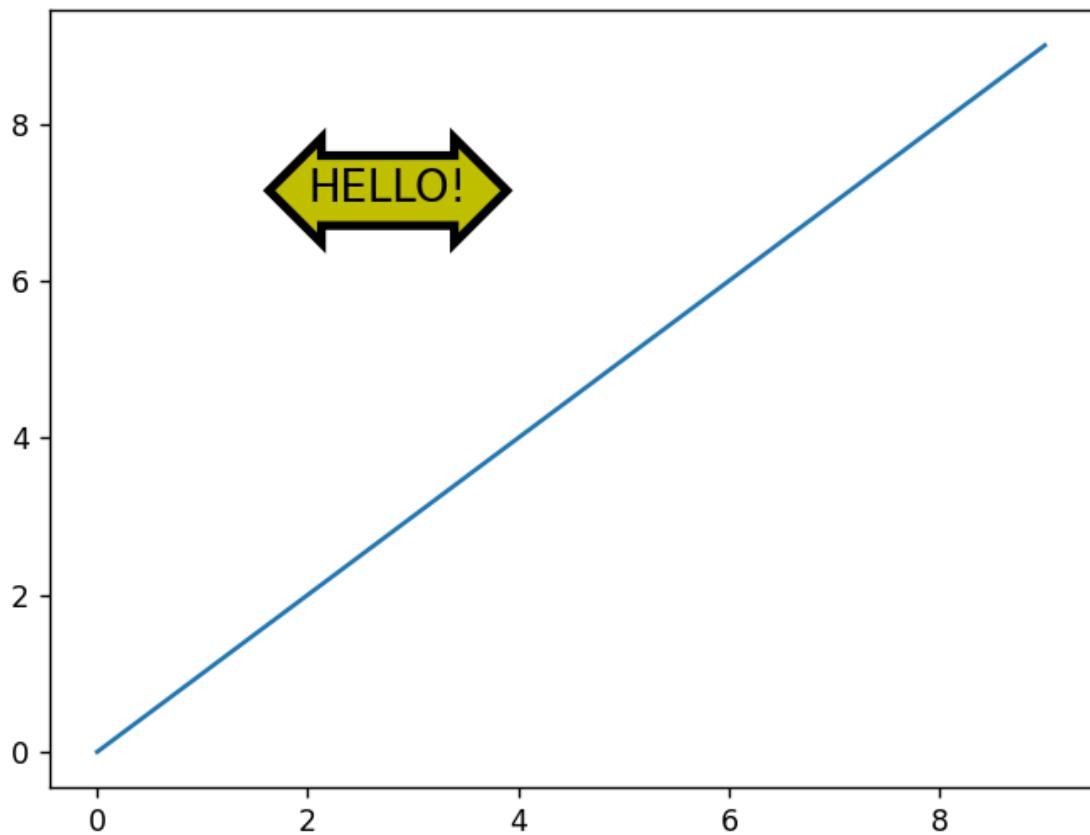


Figure 26: img_26.png

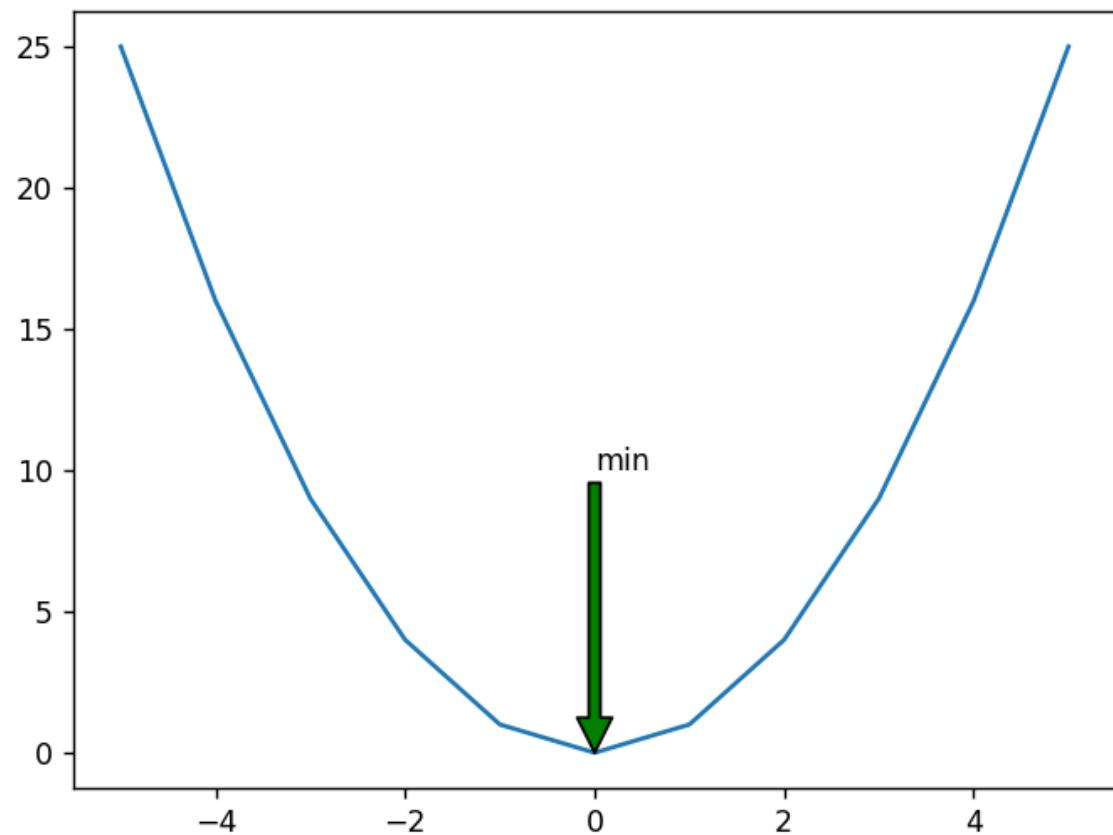


Figure 27: img_27.png

Figure 1

- □ ×

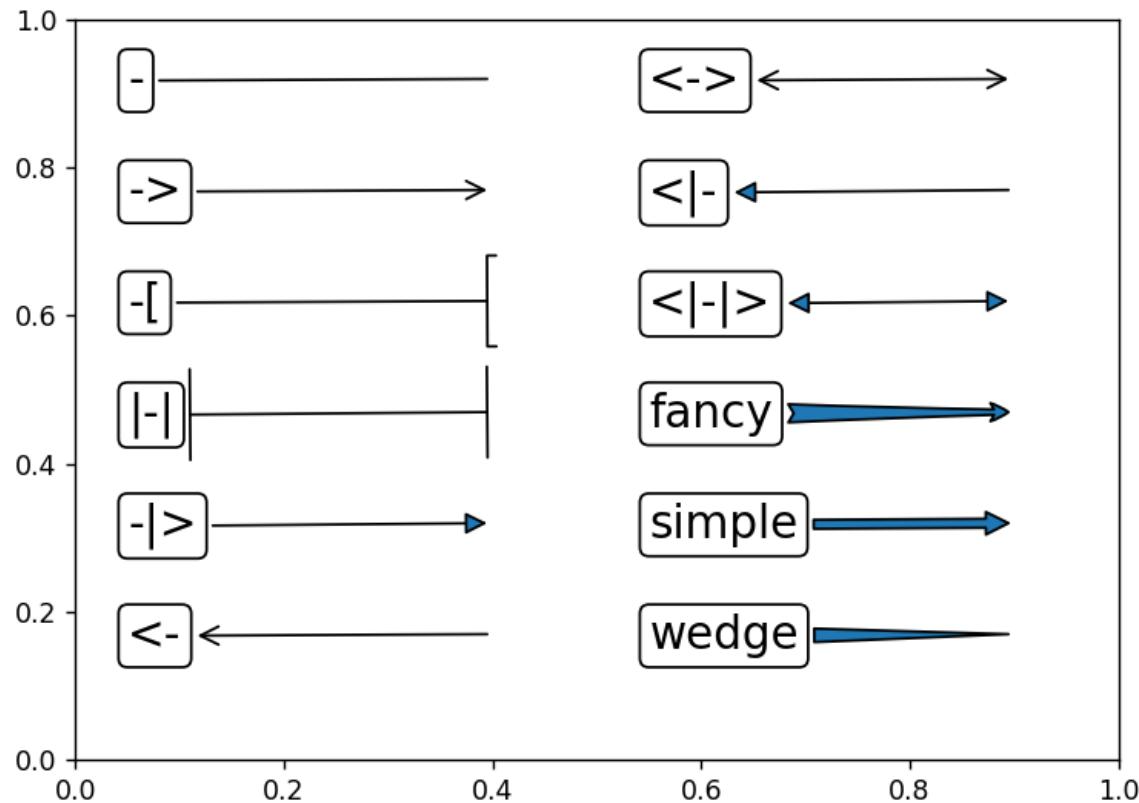


Figure 28: img_28.png

3.4. Цветовая полоса (Colorbar)

Colorbar используется для отображения соответствия между цветом и числовым значением на графиках типа `pcolor()`, `imshow()` и др:

Figure 1

— □ ×

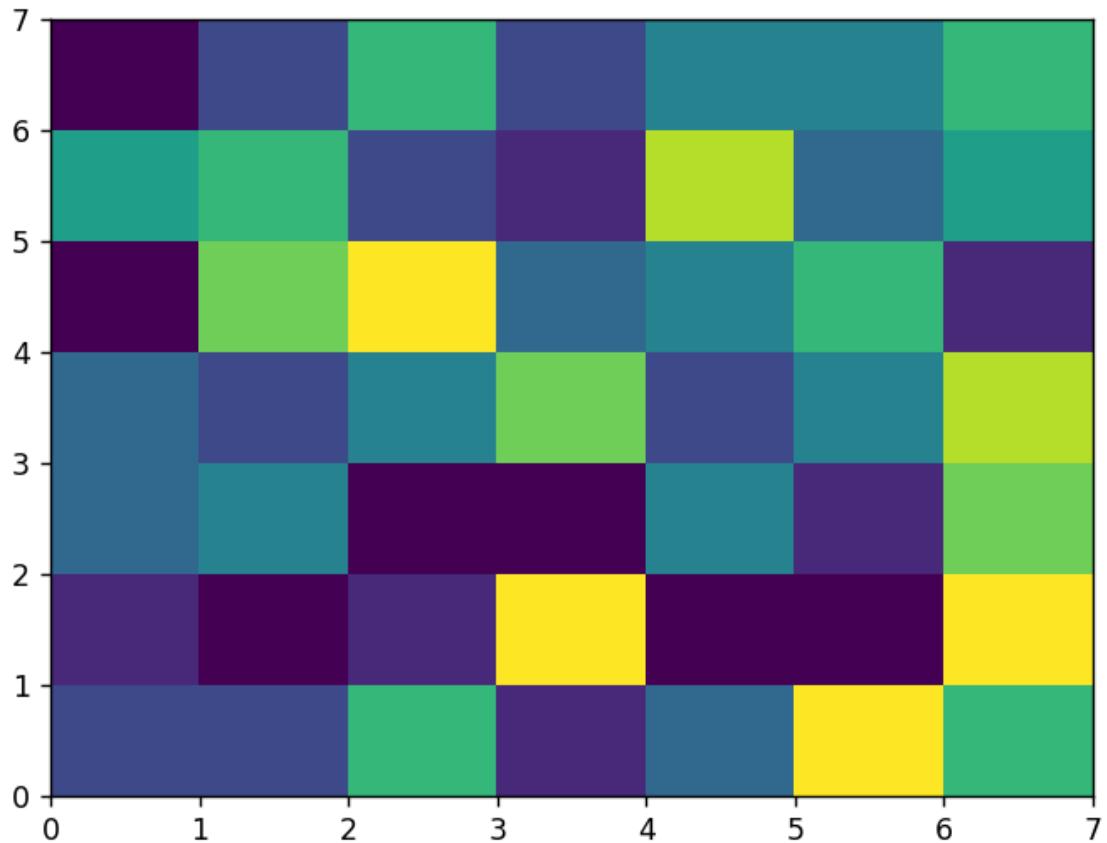


Figure 29: img_30.png

Figure 1

— □ ×

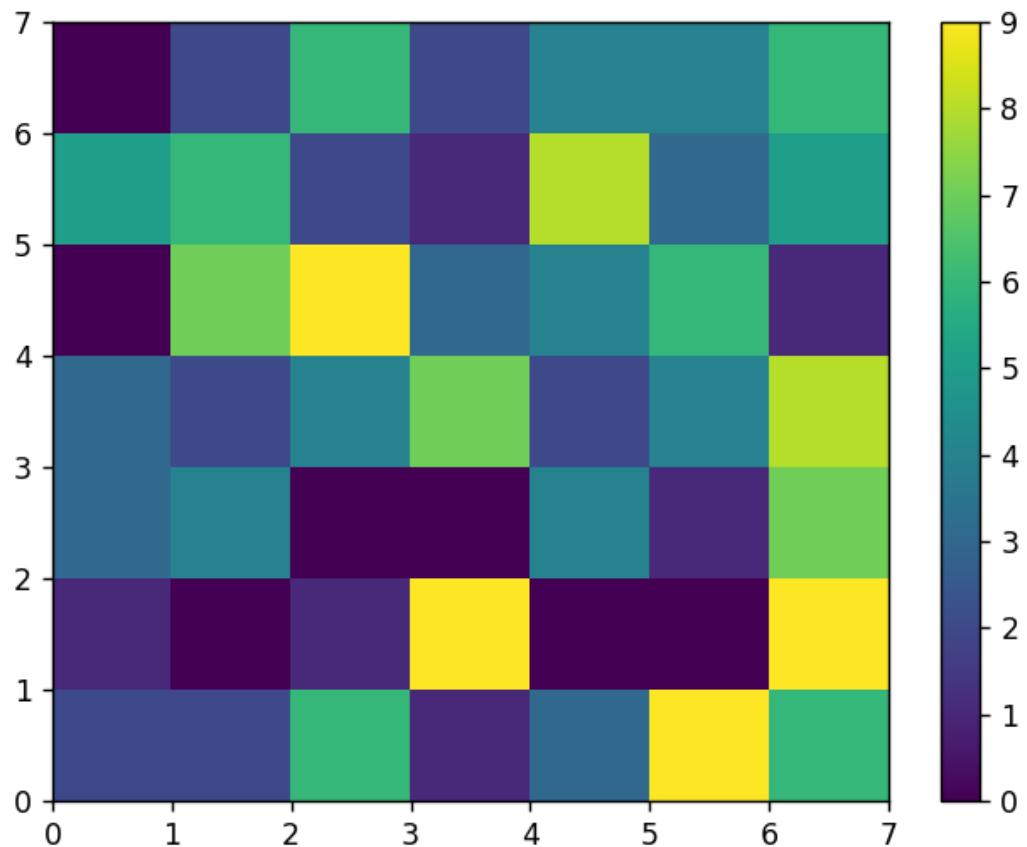


Figure 30: img_29.png

Figure 1

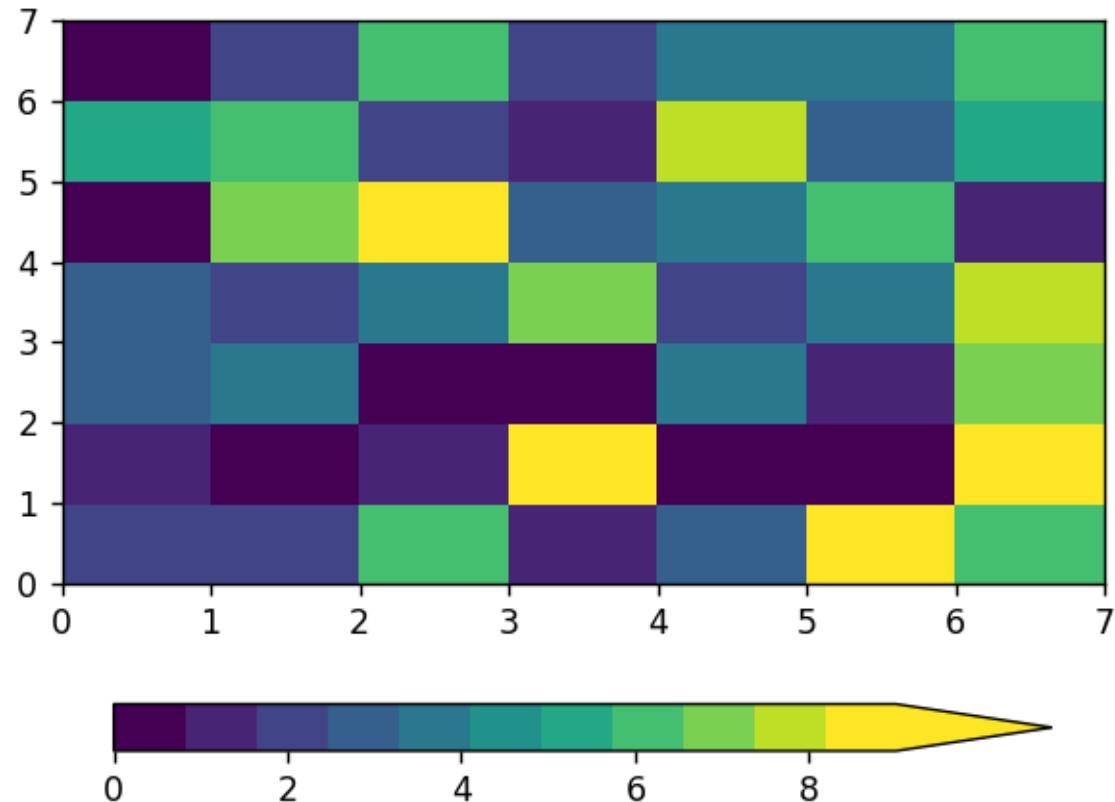


Figure 31: img_31.png

3. Индивидуальное задание (Вариант 10)

3.1. Постановка задачи

Необходимо построить график неразрывной функции и касательную к ней в заданной точке.

$$10. \quad f(x) = \begin{cases} -\cos e^x, & 0 \leq x \leq 1; \\ \ln(2x + \sin x^2), & 1 < x \leq 2. \end{cases}$$

Figure 32: img_32.png

3.2. Теоретическая часть

Касательная к графику функции $f(x)$ в точке x_0 описывается уравнением:

$$y = f(x_0) + f'(x_0) * (x - x_0)$$

где $f(x_0)$ - значение функции в точке касания, $f'(x_0)$ - значение ее производной в этой точке.

3.3. Аналитическое решение

Дана кусочная функция:

$$f(x) = \begin{cases} -\cos e^x, & 0 \leq x \leq 1; \\ \ln(2x + \sin x^2), & 1 < x \leq 2. \end{cases}$$

Для построения касательной выберем первую функцию $f(x) = -\cos(e^x)$ на отрезке $[0, 1]$. Возьмем точку касания x_0 - середина интервала.

Вычислим значение функции в этой точке:

$$f(0.5) = -\cos(e^{0.5})$$

Так как $e^{0.5} \approx 1.64872$, а $\cos(1.64872) \approx -0.078$, то

$$f(0.5) = -(-0.078) = 0.078$$

Таким образом, точка касания имеет координаты $(0.5; 0.078)$.

Далее, найдем производную функции $f(x) = -\cos(e^x)$:

$$f'(x) = \sin(e^x) * e^x$$

Подставим $x_0 = 0.5$:

$$f'(0.5) = \sin(e^{0.5}) * e^{0.5} \approx 0.996 * 1.64872 \approx 1.642$$

Подставляем значения в уравнение касательной. Получим:

$$y = 0.078 + 1.642(x - 0.5) = 0.078 + 1.642x - 0.821 = 1.642x - 0.743$$

Итак, уравнение касательной к графику функции $f(x) = -\cos(e^x)$ в точке $x_0 = 0.5$ есть

$$y = 1.642x - 0.743$$

Это прямая линия, которая касается графика в точке $(0.5; 0.078)$.

3.4. Программа

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 1, 100) # X
y = -np.cos(np.exp(x)) # Y

x0 = 0.5 # x_0

y0 = -np.cos(np.exp(x0)) # f(x_0)
k = np.sin(np.exp(x0)) * np.exp(x0) # f'(x_0)
y_kas = y0 + k * (x - x0) # y_kas(x)

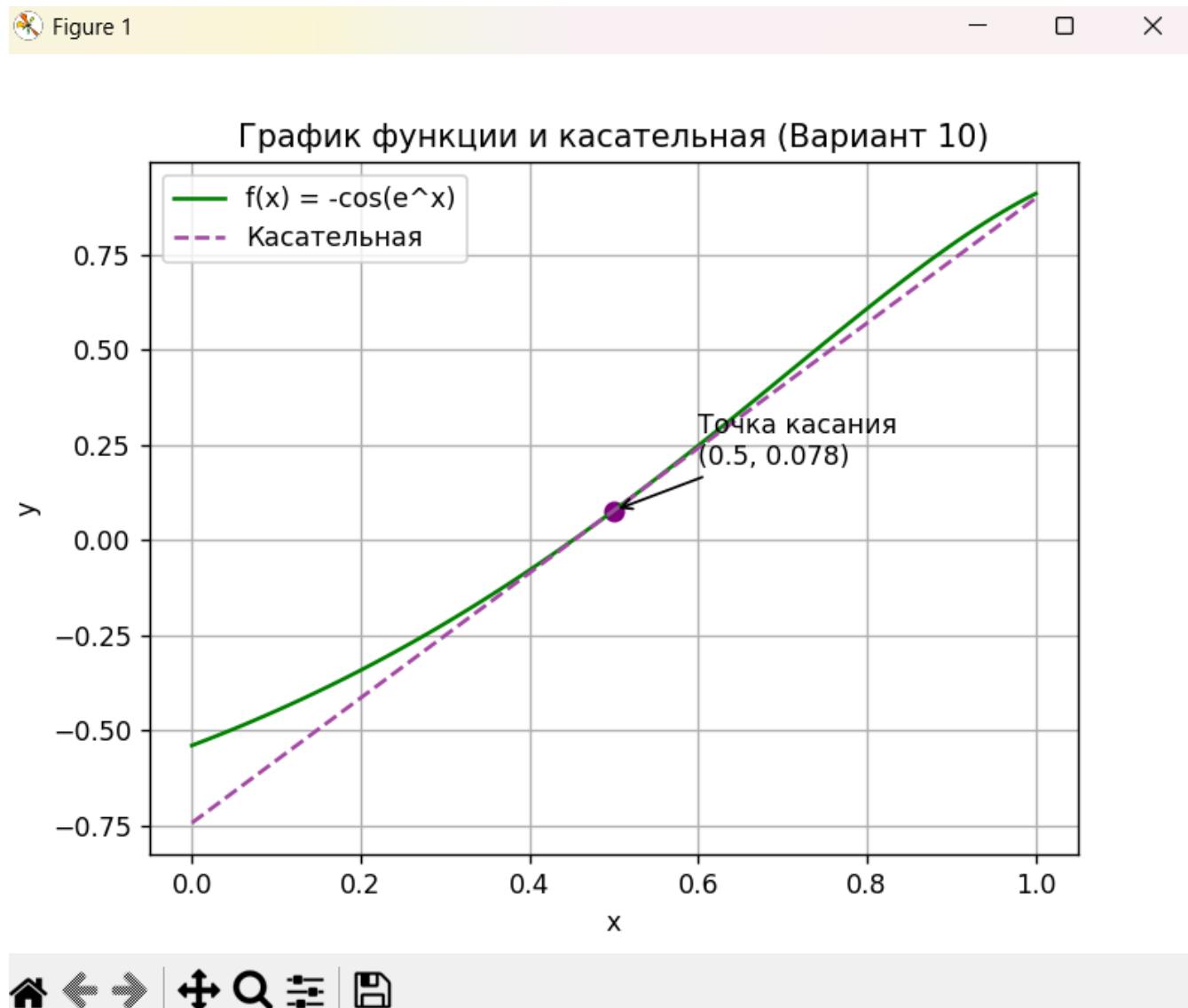
plt.plot(x, y, 'g-', label='f(x) = -cos(e^x)') # Y
plt.plot(x, y_kas, '--', color="#A349A4", label='kas(x)') # Y
plt.scatter(x0, y0, color='purple', s = 50) # x_0

plt.annotate(f'      \n{x0}, {y0:.3f})', #
             xy=(x0, y0),
             xytext=(0.6, 0.2),
             arrowprops=dict(arrowstyle='->', color='black'))

plt.title('      ( 10)') #
plt.xlabel('x') # X
plt.ylabel('y') # Y
plt.grid(True) #
plt.legend() #

plt.show()
```

3.5. Результат работы программы



Список использованных источников:

1. Devpractice Team. Библиотека Matplotlib
2. Matplotlib
3. Лабораторная работа №2