

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт перспективной инженерии  
Департамент цифровых, робототехнических систем и электроники**

**Отчет по лабораторной работе № 3  
«Работа с JupyterNotebook, JupyterLab, GoogleColab»  
по дисциплине «Искусственный интеллект и машинное обучение»**

Выполнил:  
Евдокимов Станислав Алексеевич,  
студент 2 курса группа ИВТ-б-о-23-  
1, 09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Автоматизированные системы  
обработки информации и  
управления», очная форма обучения

Руководитель практики:  
Воронкин Роман Александрович

Ставрополь 2025

**Тема:** Основы работы с библиотекой matplotlib.

**Цель работы:** исследовать базовые возможности библиотеки matplotlib языка программирования Python.

**Порядок выполнения работы:**

[StanislavEvdokimov/Lab3](#)

**Задание 1.** Построение простого графика.

▼ Напишите код, который строит график функции  $y = x^2$  на интервале  $[-10, 10]$ . Добавьте заголовок, подписи осей и сетку

```
[29]: x = np.linspace(-10, 10, 50)
      y = x**2
      plt.plot(x,y)
      plt.title("График функции y = x^2")
      plt.xlabel("x")
      plt.ylabel("y")
      plt.grid(True)
```



Рисунок 1. Простой график

**Задание 2.** Настройка стилей и цветов.

Постройте три линии на одном графике:

$y = x$  (синяя, пунктирная линия),  
 $y = x^2$  (зеленая, штрих-пунктирная линия),  
 $y = x^3$  (красная, сплошная линия).

Добавьте легенду и сделайте оси одинакового масштаба.

```
[61]: x = np.linspace(-1, 1, 50)
      y1 = x
      y2 = x**2
      y3 = x**3
      plt.plot(x, y1, c='b', linestyle='--', label='y = x')
      plt.plot(x, y2, c='g', linestyle='-.', label='y = x^2')
      plt.plot(x, y3, c='r', linestyle='-', label='y = x^3')
      plt.grid(True)
      plt.legend()
      plt.axis('equal')
```

```
[61]: (-1.1, 1.1, -1.1, 1.1)
```

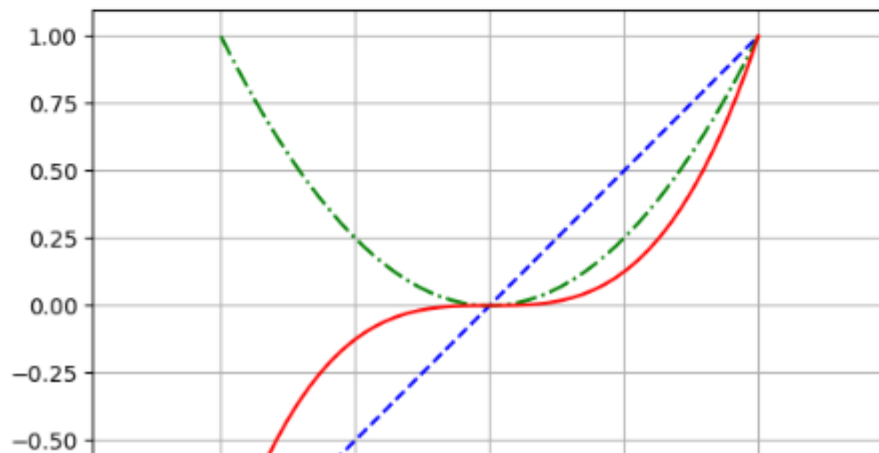


Рисунок 2. Стили и цвета

**Задание 3.** Использование различных типов графиков.

Сгенерируйте 50 случайных точек и постройте диаграмму рассеяния (scatter plot), где цвет точек зависит от их координаты по оси  $x$ , а размер точек зависит от координаты по оси  $y$ .

```
[84]: x = np.random.rand(50)
y = np.random.rand(50)
plt.scatter(x, y, c = x, s = y*100)
plt.title("Диаграмма рассеяния")
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True)
```

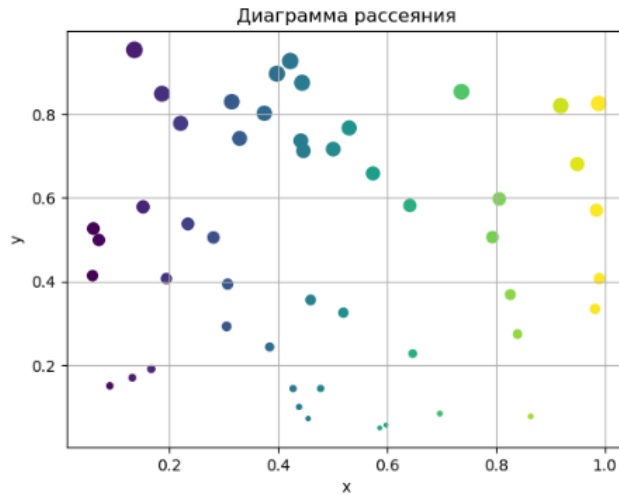


Рисунок 3. Типы графиков

#### Задание 4. Гистограмма распределения.

- Сгенерируйте 1000 случайных чисел из нормального распределения с параметрами  $\mu = 0$ ,  $\sigma = 1$  и постройте их гистограмму с 30 бинами. Добавьте вертикальную линию в среднем значении.

```
[125]: xy = np.random.normal(loc = 0, scale = 1, size = 1000)
plt.hist(xy, bins = 30, edgecolor = "black")
sredn = np.mean(xy)
plt.axvline(sredn, color="red")
plt.title("Гистограмма распределения")
plt.grid(True)
```



Рисунок 4. Гистограмма распределения

#### Задание 5. Столбчатая диаграмма.

Создайте столбчатую диаграмму, которая показывает количество студентов, получивших оценки:

"Отлично" — 20 человек,  
"Хорошо" — 35 человек,  
"Удовлетворительно" — 30 человек,  
"Неудовлетворительно" — 15 человек.

Добавьте подписи к осям и заголовок.

```
[144]: res = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
stud = [20, 35, 30, 15]
plt.bar(res, stud)
plt.title("Столбчатая диаграмма")
plt.xlabel("Оценка")
plt.ylabel("Количество студентов")
```

```
[144]: Text(0, 0.5, 'Количество студентов')
```

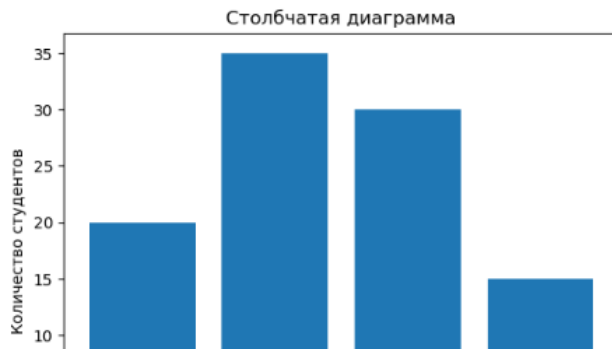


Рисунок 5. Столбчатая диаграмма

### Задание 6. Круговая диаграмма.

Используя данные предыдущей задачи, постройте круговую диаграмму с процентными подписями секторов.

```
[180]: plt.pie(stud, labels = res, autopct = "%1.1f%%")
plt.title("Круговая диаграмма")
```

```
[180]: Text(0.5, 1.0, 'Круговая диаграмма')
```



Рисунок 6. Круговая диаграмма

### Задание 7. Трехмерный график поверхности.

Используя `mpl_toolkits.mplot3d`, постройте 3D-график функции  $z = \sin(\sqrt{x^2 + y^2})$  на сетке значений  $x, y$  в диапазоне  $[-5, 5]$ .

```
[220]: import mpl_toolkits.mplot3d
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))
siz = plt.figure(figsize = (5, 5))
tred = siz.add_subplot(111, projection = "3d")
graf = tred.plot_surface(x, y, z, cmap = "viridis")
```

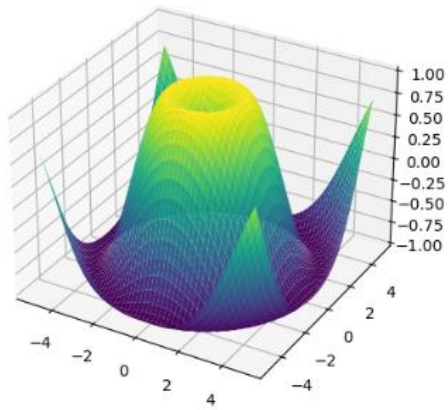


Рисунок 7. Трехмерный график поверхности

**Задание 8.** Множественные подграфики.

Постройте четыре графика в одной фигуре (2 × 2 сетка):

1. Линейный график  $y = x$
2. Парабола  $y = x^2$
3. Синус  $y = \sin(x)$
4. Косинус  $y = \cos(x)$

Добавьте заголовки к каждому подграфику.

```
In [5]: x = np.linspace(-10, 10, 50)
size, axes = plt.subplots(2, 2, figsize = (5, 5))
axes[0, 0].plot(x, x)
axes[0, 0].set_title("y = x")
axes[0, 0].grid(True)
axes[0, 1].plot(x, x**2)
axes[0, 1].set_title("$y = x^2$")
axes[0, 1].grid(True)
axes[1, 0].plot(x, np.sin(x))
axes[1, 0].set_title("$y = \sin(x)$")
axes[1, 0].grid(True)
axes[1, 1].plot(x, np.cos(x))
axes[1, 1].set_title("$y = \cos(x)$")
axes[1, 1].grid(True)
```

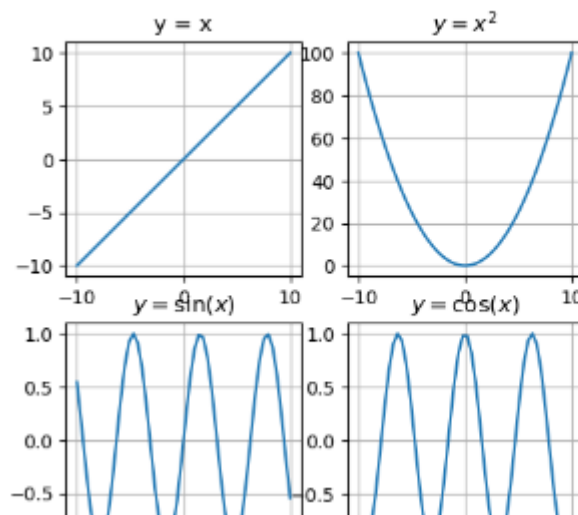


Рисунок 8. Множественные подграфики

**Задание 9.** Тепловая карта.

- ▼ Создайте случайную матрицу  $10 \times 10$  с элементами от 0 до 1 и визуализируйте её как тепловую карту с цветовой шкалой. ¶

```
[241]: tepl = np.random.rand(10, 10)
plt.figure(figsize=(5, 5))
tmap = plt.imshow(tepl)
plt.colorbar(tmap)
plt.title("Тепловая карта")
```

```
[241]: Text(0.5, 1.0, 'Тепловая карта')
```

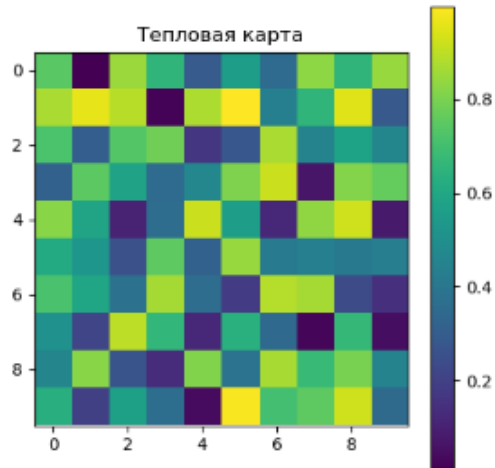


Рисунок 9. Тепловая карта

**Задание 10.** Индивидуальное задание 1. Температурные изменения за сутки



Измерялась температура воздуха в течение дня:

Время суток (часы): [0, 3, 6, 9, 12, 15, 18, 21, 24]  
Температура (°C): [-5, -6, -4, 2, 8, 12, 10, 3, -2]

Используйте разные цвета для температуры ниже и выше 0°C. ¶

```
2]: import matplotlib.pyplot as plt
import numpy as np
time = [0, 3, 6, 9, 12, 15, 18, 21, 24]
temp = [-5, -6, -4, 2, 8, 12, 10, 3, -2]
below_zero = [temp if temp < 0 else None for temp in temperatures]
above_zero = [temp if temp >= 0 else None for temp in temperatures]
plt.figure(figsize=(10, 6))
plt.plot(hours, below_zero, 'b-o', label='Температура < 0°C', linewidth=2, markersize=8)
plt.plot(hours, above_zero, 'r-o', label='Температура ≥ 0°C', linewidth=2, markersize=8)
plt.title("Температурные изменения")
plt.xlabel("Время суток")
plt.ylabel("Параметры температуры")
plt.xticks(hours)
plt.grid(True, linestyle='--', alpha=0.6)
plt.axhline(0, color='black', linestyle='-', linewidth=1)
plt.tight_layout()
plt.show()
```

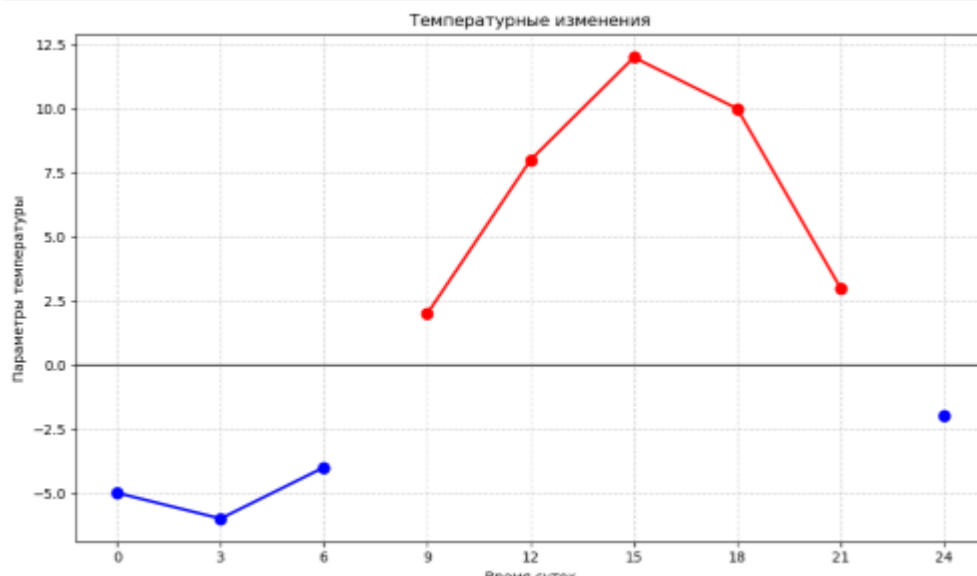


Рисунок 10. Температурные изменения

**Задание 11.** Индивидуальное задание 2. Задача на построение столбчатых диаграмм Matplotlib

## Количество научных публикаций по областям знаний

Анализ количества публикаций за год:

Области: ['Математика', 'Физика', 'Биология', 'Химия', 'Компьютерные науки']  
Публикации: [120, 90, 150, 110, 180] **↑**

Используйте диаграмму с горизонтальными столбцами и отметьте область с наибольшим количеством публикаций.

```
111: city = ["Математика", "Физика", "Биология", "Химия", "Компьютерные науки"]  
    tra = [120, 90, 150, 110, 180]  
    max_r = np.max(tra)  
    plt.figure(figsize=(8, 5))  
    bars = plt.bar(city, tra)  
    plt.axhline(y = max_r, color = 'r')  
    plt.title("Анализ кол-ва публикаций")  
    plt.xlabel("Области")  
    plt.ylabel("Публикации")  
    plt.show()
```

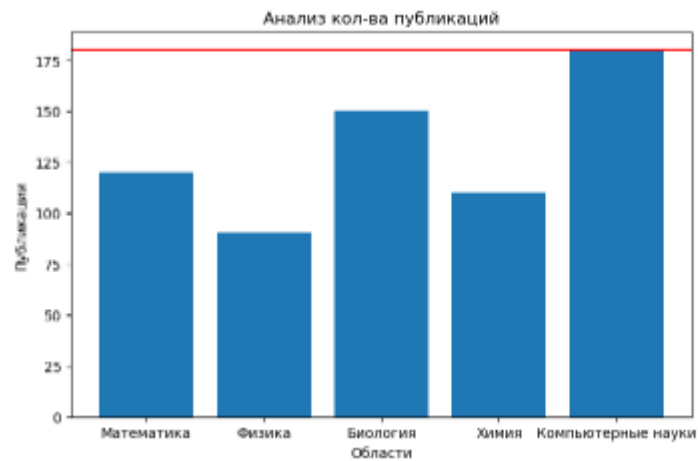


Рисунок 11. Задача на построение столбчатых диаграмм Matplotlib

**Задание 12.** Индивидуальное задание 3. Задача на вычисление определенного интеграла с помощью Matplotlib

В каждой задаче требуется:

1. Построить график подинтегральной функции.
2. Вычислить площадь под кривой на заданном отрезке как значение определенного интеграла.

Рассчитайте площадь под функцией:  
на интервале  $[-2, 2]$ .

$$f(x) = 2x + 3$$

```
2]: def f(x): return (2*x + 3)
x = np.linspace(-2, 2)
y = f(x)
dx = x[1] - x[0]
integ = np.trapz(y, x, dx)
plt.figure(figsize = (8, 5))
plt.plot(x, y)
plt.fill_between(x, y)
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```

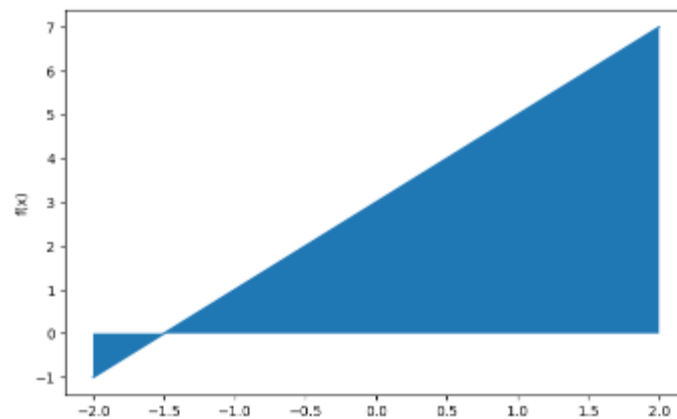


Рисунок 12. Задача на вычисление определенного интеграла с помощью  
Matplotlib

**Задание 13.** Индивидуальное задание 4. Задачи на построение 3D-  
графиков с помощью Matplotlib

### Экспоненциальный гребень

Постройте график:

$$f(x, y) = e^{-\sqrt{x^2+y^2}} \cdot \cos(2\pi\sqrt{x^2+y^2})$$

на  $x, y \in [-3, 3]$

```
In [5]: from mpl_toolkits.mplot3d import Axes3D
x = np.linspace(-3, 3, 300)
y = np.linspace(-3, 3, 300)
X, Y = np.meshgrid(x, y)
Z = np.exp(-np.sqrt(X**2 + Y**2)) * np.cos(2 * np.pi * np.sqrt(X**2 + Y**2))
fig = plt.figure(figsize = (10, 7))
ax = fig.add_subplot(111, projection = '3d')
surf = ax.plot_surface(X, Y, Z, cmap = 'inferno')
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("f(x, y)")
fig.colorbar(surf)
plt.show()
```

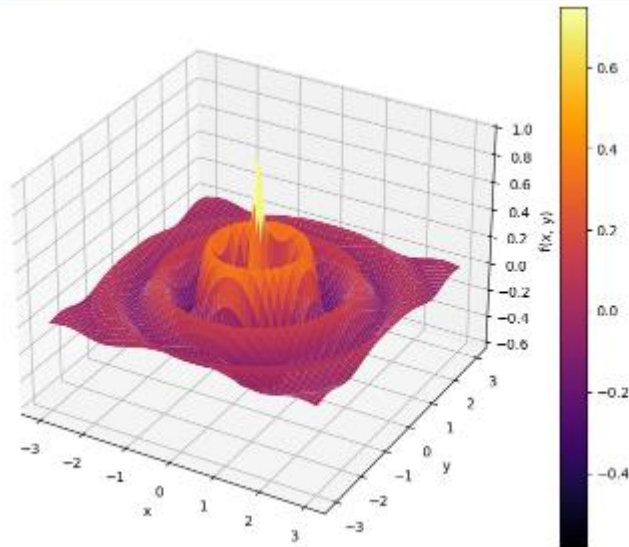


Рисунок 13. Задачи на построение 3D-графиков с помощью Matplotlib

### Ответы на контрольные вопросы:

#### 1. Как осуществляется установка пакета matplotlib?

Bash: `pip install matplotlib`

Или через Anaconda: `conda install matplotlib`

#### 2. Какая "магическая" команда должна присутствовать в ноутбуках

Jupyter для корректного отображения графиков?

Python: `%matplotlib inline`

(Добавляется в начале ноутбука для встраивания графиков в вывод ячейки.)

#### 3. Как отобразить график с помощью функции plot?

Python: `import matplotlib.pyplot as plt`

`plt.plot([1, 2, 3], [4, 5, 1]) # X, Y`

```
plt.show()
```

#### **4. Как отобразить несколько графиков на одном поле?**

Используйте `plt.plot` несколько раз:

```
Python: plt.plot(x1, y1, label='График 1')
```

```
plt.plot(x2, y2, label='График 2')
```

```
plt.legend()
```

```
plt.show()
```

#### **5. Какой метод использовать для построения диаграмм категориальных данных?**

- Для столбчатых диаграмм: `plt.bar()` или `plt.barh()` (горизонтальные).

- Для круговых: `plt.pie()`.

#### **6. Основные элементы графика:**

- Оси (X, Y), заголовок (`plt.title()`), подписи осей (`plt.xlabel()`, `plt.ylabel()`), легенда (`plt.legend()`), сетка (`plt.grid()`).

#### **7. Управление текстовыми надписями:**

```
Python: plt.title("Заголовок")
```

```
plt.xlabel("Ось X")
```

```
plt.ylabel("Ось Y")
```

```
plt.text(x, y, "Текст") # Добавить текст в координаты (x, y)
```

#### **8. Управление легендой:**

```
Python: plt.plot(x, y, label='Линия 1')
```

```
plt.legend(loc='upper right') # Положение: 'best', 'upper left', ит.д.
```

#### **9. Цвет и стиль линий:**

```
Python: plt.plot(x, y, color='red', linestyle='--', linewidth=2)
```

- Цвета: `'red'`, `'#FF0000'`, `'r'`.

- Стили: `'-'` (сплошная), `'--'` (пунктир), `'.'` (точечная).

#### **10. Размещение графиков в разных полях:**

Используйте `subplot()`:

```
Python: plt.subplot(2, 1, 1) # 2 строки, 1 столбец, позиция 1
```

```
plt.plot(x1, y1)
```

```
plt.subplot(2, 1, 2) # Позиция 2  
plt.plot(x2, y2)
```

### **11. Построение линейного графика:**

```
Python: x = [1, 2, 3]  
y = [4, 5, 1]  
plt.plot(x, y)  
plt.show()
```

### **12. Заливка между графиком и осью / двумя графиками:**

- Между графиком и осью:

```
Python: plt.fill_between(x, y, color='blue', alpha=0.3)
```

- Между двумя графиками:

```
python  
plt.fill_between(x, y1, y2, color='green')
```

### **13. Выборочная заливка по условию:**

```
Python: plt.fill_between(x, y, where=(y > 0), color='red')
```

### **14. Двухцветная заливка:**

Разделите заливку на два условия:

```
Python: plt.fill_between(x, y, where=(y > 0), color='red')  
plt.fill_between(x, y, where=(y <= 0), color='blue')
```

### **15. Маркировка графиков:**

Используйте параметр `marker`:

```
Python: plt.plot(x, y, marker='o', markersize=5, markeredgecolor='black')
```

- Маркеры: `'o'`, `'s'`, `'^'`, `'x'`.

### **16. Обрезка графиков:**

Задайте пределы осей:

```
Python: plt.xlim(0, 10)  
plt.ylim(-5, 5)
```

### **17. Ступенчатый график:**

```
Python: plt.step(x, y, where='post') # 'pre', 'mid', 'post'
```

Особенность: Значения меняются на границах интервалов.

### **18. Стековый график:**

Python: `plt.stackplot(x, y1, y2, y3, labels=['A', 'B', 'C'])`

Особенность: Отображает накопление величин.

### **19. Stem-график:**

Python: `plt.stem(x, y, linefmt='--', markerfmt='o')`

Особенность: Отображает отдельные точки с вертикальными линиями.

### **20. Точечный график:**

Python: `plt.scatter(x, y, s=50, c='red', edgecolor='black')`

Особенность: Показывает распределение точек.

### **21. Столбчатые диаграммы:**

Python: `plt.bar(categories, values)`

`plt.barh(categories, values)` # Горизонтальная

### **22. Групповая столбчатая диаграмма и errorbar:**

- Групповая:

Python: `plt.bar(x - width/2, values1, width, label='Группа 1')`

`plt.bar(x + width/2, values2, width, label='Группа 2')`

- С errorbar:

Python: `plt.errorbar(x, y, yerr=errors, fmt='o')`

### **23. Круговая диаграмма:**

Python: `plt.pie(sizes, labels=labels, autopct='%1.1f%%')`

`plt.axis('equal')`

### **24. Цветовая карта (colormap):**

- Пример:

Python: `plt.imshow(data, cmap='viridis')`

`plt.colorbar()`

- Доступные карты: `'plasma'`, `'coolwarm'`, `'jet'`.

### **25. Отображение изображения:**

Python: `img = plt.imread('image.png')`

`plt.imshow(img)`

`plt.axis('off')`

## **26. Тепловая карта:**

```
Python: plt.imshow(data, cmap='hot')  
plt.colorbar()
```

## **27-30. 3D-графики:**

Импортируйте 3D-модуль:

```
Python: from mpl_toolkits.mplot3d import Axes3D
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111, projection='3d')
```

- Линейный 3D-график:

```
ax.plot3D(x, y, z)
```

- Точечный 3D-график:

```
ax.scatter3D(x, y, z, c=z, cmap='viridis')
```

- Каркасная поверхность:

```
ax.plot_wireframe(X, Y, Z)
```

- Трехмерная поверхность:

```
ax.plot_surface(X, Y, Z, cmap='viridis')
```

**Вывод:** в ходе лабораторной работы мы исследовали базовые возможности библиотеки matplotlib языка программирования Python