

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
дисциплины
«Основы теории электромагнитных полей и волн»
Вариант 13

Выполнил:
Милькевич Александр Александрович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи,
направленность (профиль)
«Интеллектуальная обработка данных
в инфокоммуникационных системах и
сетях», очная форма обучения

(подпись)

Проверил:
Ассистент департамента цифровых,
робототехнических систем и
электроники Хацукова А.И

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Основы работы с библиотекой matplotlib

Цель: исследовать базовые возможности библиотеки matplotlib языка программирования Python

Ход выполнения работы:

Ссылка на гит хаб: https://github.com/lolndo/AI_Lab3.git

1.Задание 1. Напишите код, который строит график функции $y = x^2$ на интервале $[-10, 10]$. Добавьте заголовок, подписи осей и сетку.

```
x = np.linspace(-10,10,50)
y = x**2

plt.title('задание 1 линейная зависимость y=x^2')
plt.xlabel("x")
plt.ylabel("y")
plt.grid()

plt.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x24f4f145a30>]

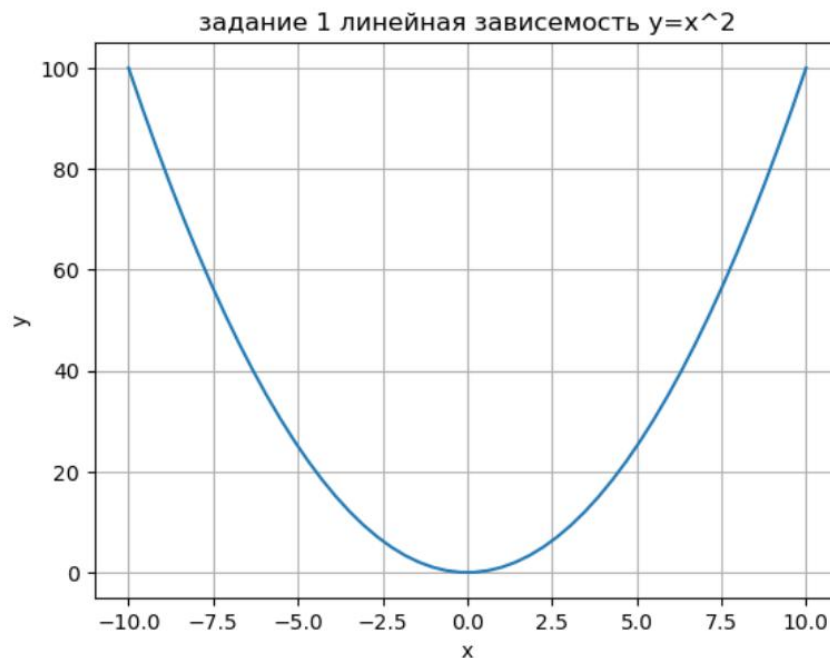


Рисунок 1. Выполнение задания №1

2.Задание №2. Постройте три линии на одном графике. Добавьте легенду и сделайте оси одинакового масштаба.

```

y1=x
y2 = x**2
y3=x**3
plt.figure(figsize=(8, 6))

plt.plot(x , y1, "--b", label="y = x")
plt.plot(x ,y2, "g-.", label="y = x^2")
plt.plot(x , y3, "r", label="y = x^3")

plt.title('задание 2. Линии разных цветов')
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
plt.legend()

plt.xlim(-10, 10) # Пределы для оси x
plt.ylim(-10, 10) # Пределы для оси y
plt.gca().set_aspect('equal')

```

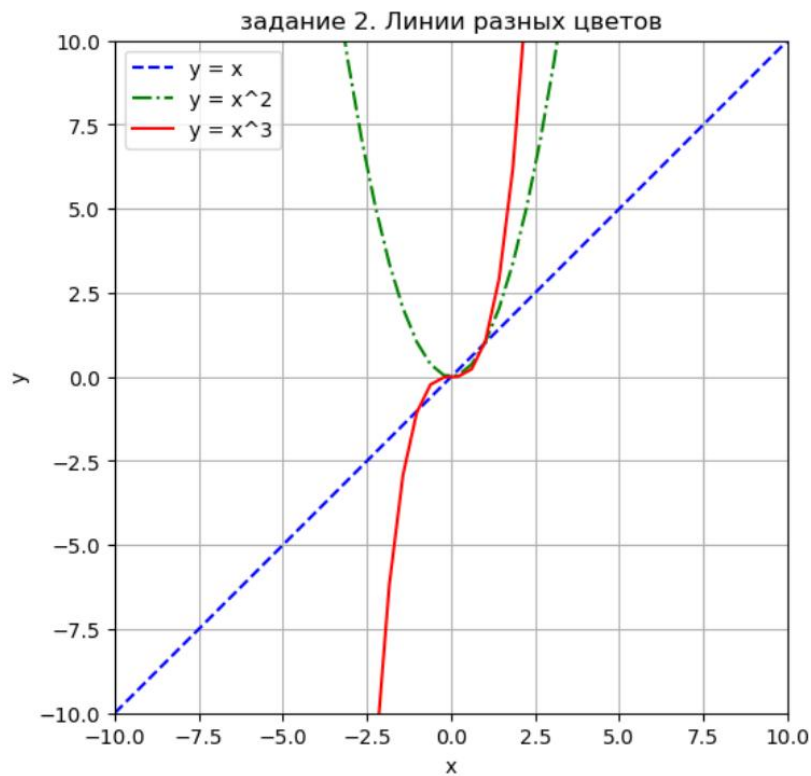


Рисунок 2. Программа для выполнения задания №2

3.Задание №3. 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.

```

: x = np.random.rand(50)
  y = np.random.rand(50)

colors = x
sizes = 200*y

plt.figure(figsize=(8, 6))
plt.scatter(x, y, c=colors, s=sizes, alpha=0.7)

plt.xlabel("Координата X")
plt.ylabel("Координата Y")
plt.title("Диаграмма рассеяния: Цвет - X, Размер - Y")

plt.show()

```

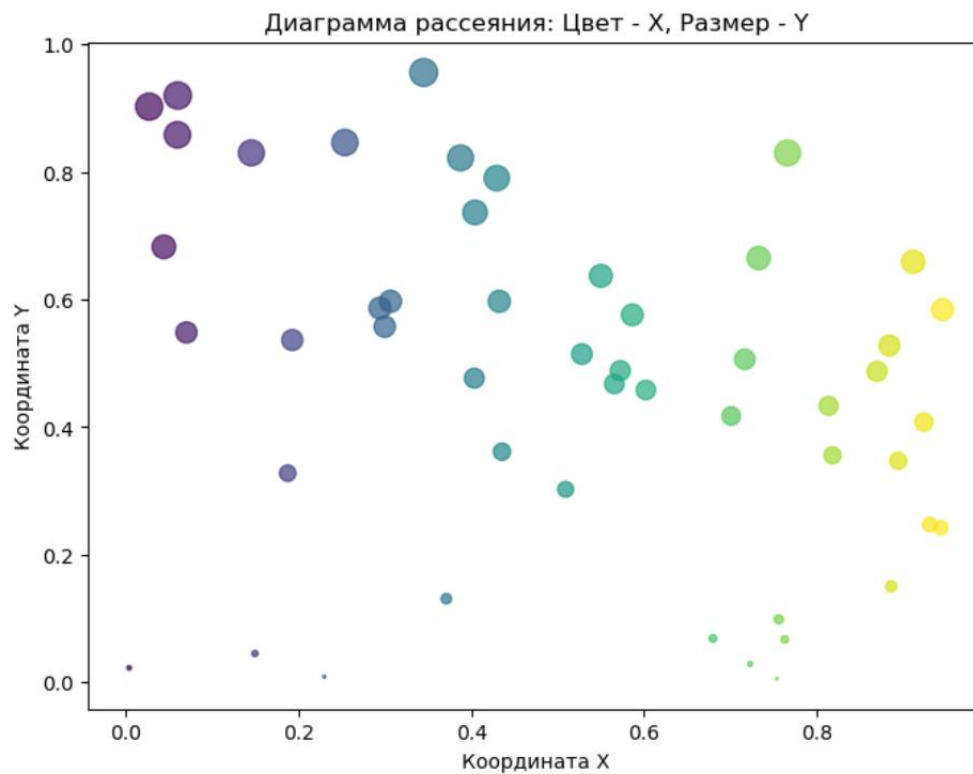


Рисунок 3. Программа для выполнения задания №3

4.Задание №4. 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.

```

mu = 0
sigma = 1
num_samples = 1000

data = np.random.normal(mu, sigma, num_samples)
mean_value = np.mean(data)

plt.figure(figsize=(10, 6))
plt.hist(data, bins=30, density=True, alpha=0.7, color='skyblue', edgecolor='black')

plt.axvline(mean_value, color='red', linestyle='dashed', linewidth=2, label=f'Среднее = {mean_value:.2f}')

plt.xlabel("Значение")
plt.ylabel("Частота")
plt.title("Гистограмма случайных чисел из нормального распределения")

plt.legend()
plt.grid(True)
plt.show()

```

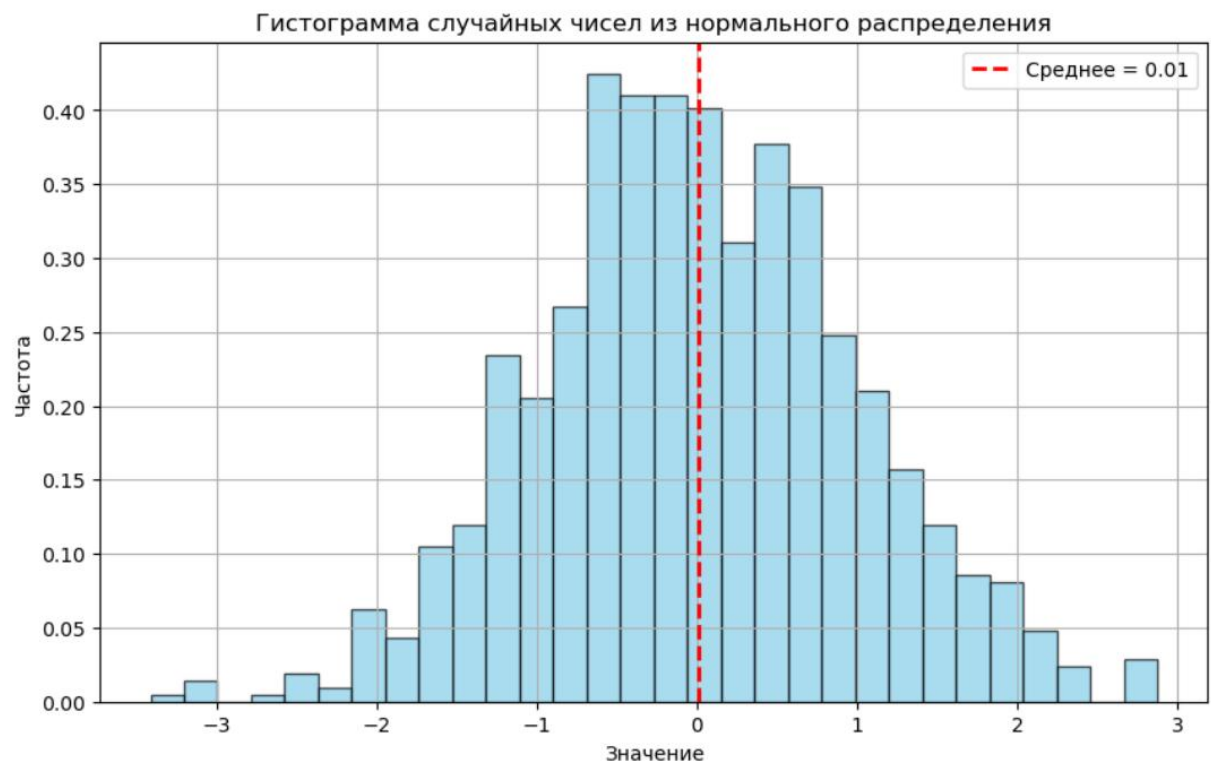


Рисунок 4. Программа для выполнения задания №4

5.Задание №5. Создайте столбчатую диаграмму, которая показывает количество студентов, получивших оценки.

```

cool = 20
good = 35
ud = 30
neud = 15
groups = ["cool", "good", "ud", "neud"]
counts = [cool, good, ud, neod]
colors=["ForestGreen", "YellowGreen", "Gold", "red"]
plt.bar(groups,counts, color=colors)
plt.title("успеваемость учеников")
plt.ylabel("количество учеников")
plt.xlabel("оценка")

```

Text(0.5, 0, 'оценка')

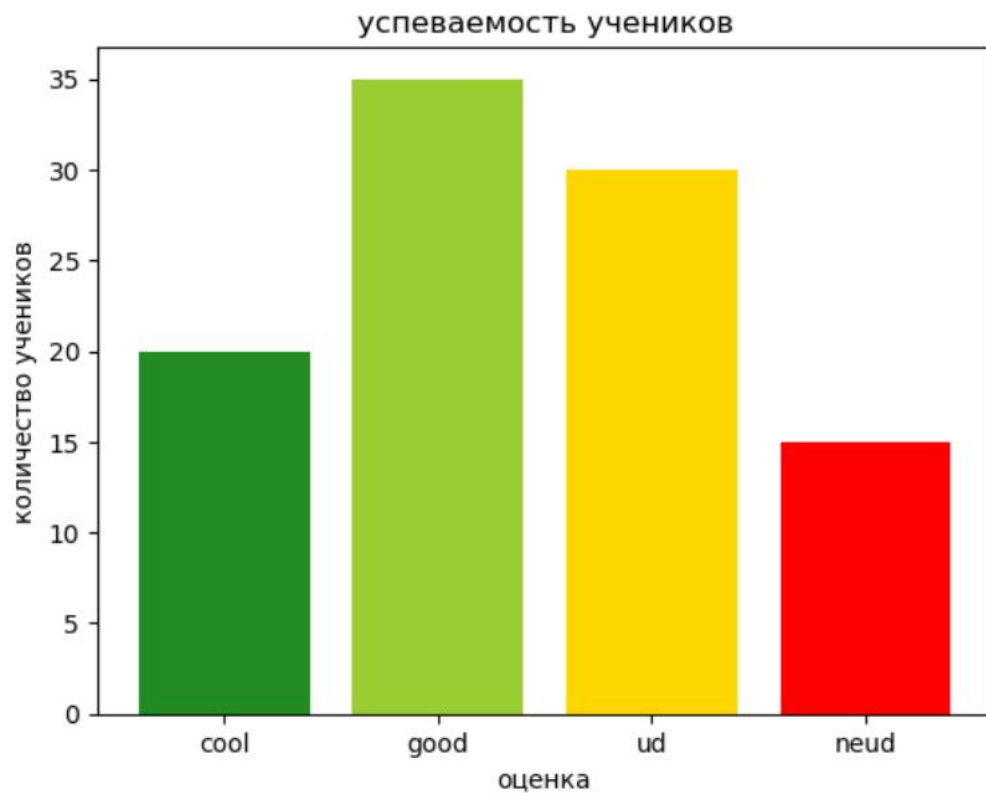


Рисунок 5. Программа для выполнения задания №5

6.Задание 6. Используя данные предыдущей задачи, постройте круговую диаграмму с процентными подписями секторов.

```
fig, ax = plt.subplots()
ax.pie(counts, labels=groups, autopct='%1.1f%%', colors=colors,
       rotatelabels=True)
ax.axis("equal")
```

```
(-1.0999966430892643,
 1.0999998401471078,
 -1.0999988490952763,
 1.0999965958963198)
```

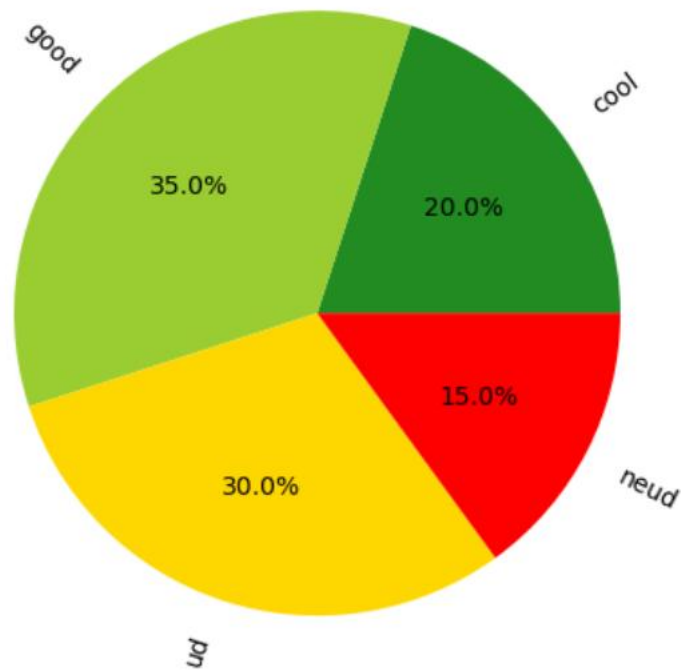


Рисунок 6. Программа для выполнения задания №6

7.Задание 7. Используя `mpl_toolkits.mplot3d` , постройте 3D-график функции $z = \sin(\sqrt{x^2 + y^2})$ на сетке значений x, y в диапазоне $[-5, 5]$.

```
from mpl_toolkits.mplot3d import Axes3D
x = np.linspace(-np.pi, np.pi, 50)
y = x
z = np.sin(np.sqrt(x**2+y**2))
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(x, y, z, label='parametric curve')
```

[<mpl_toolkits.mplot3d.art3d.Line3D at 0x2176c20eea0>]

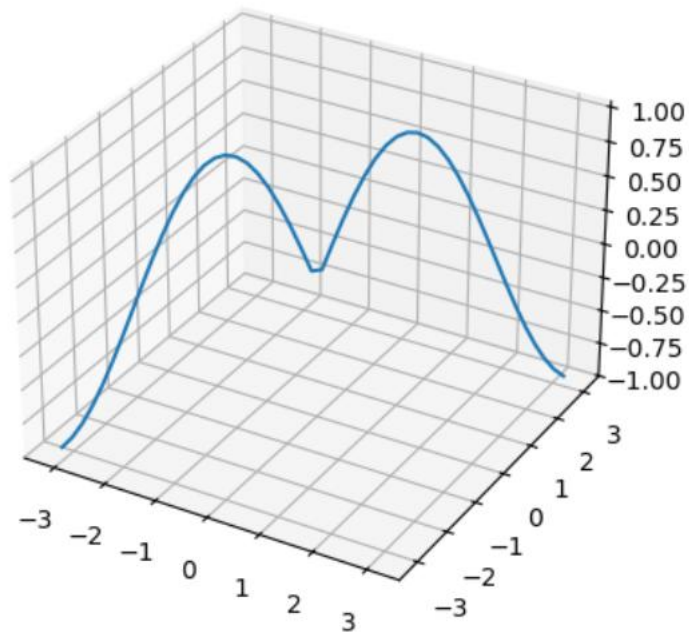


Рисунок 7. Программа для выполнения задания №7

8.Задание 8. Постройте четыре графика в одной фигуре (2×2 сетка).

```
x = np.linspace(-10, 10, 100)
y1 = x
y2 = x**2
y3 = np.sin(x)
y4 = np.cos(x)

fig, axes = plt.subplots(2, 2, figsize=(12, 8))

axes[0, 0].plot(x, y1)
axes[0, 0].set_title('y = x (Линейный)')

axes[0, 1].plot(x, y2)
axes[0, 1].set_title('y = x^2 (Парабола)')

axes[1, 0].plot(x, y3)
axes[1, 0].set_title('y = sin(x) (Синус)')

axes[1, 1].plot(x, y4)
axes[1, 1].set_title('y = cos(x) (Косинус)')

fig.suptitle('Четыре графика в одной фигуре', fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])

plt.show()
```

Рисунок 8.1. Листинг для выполненная задания №8

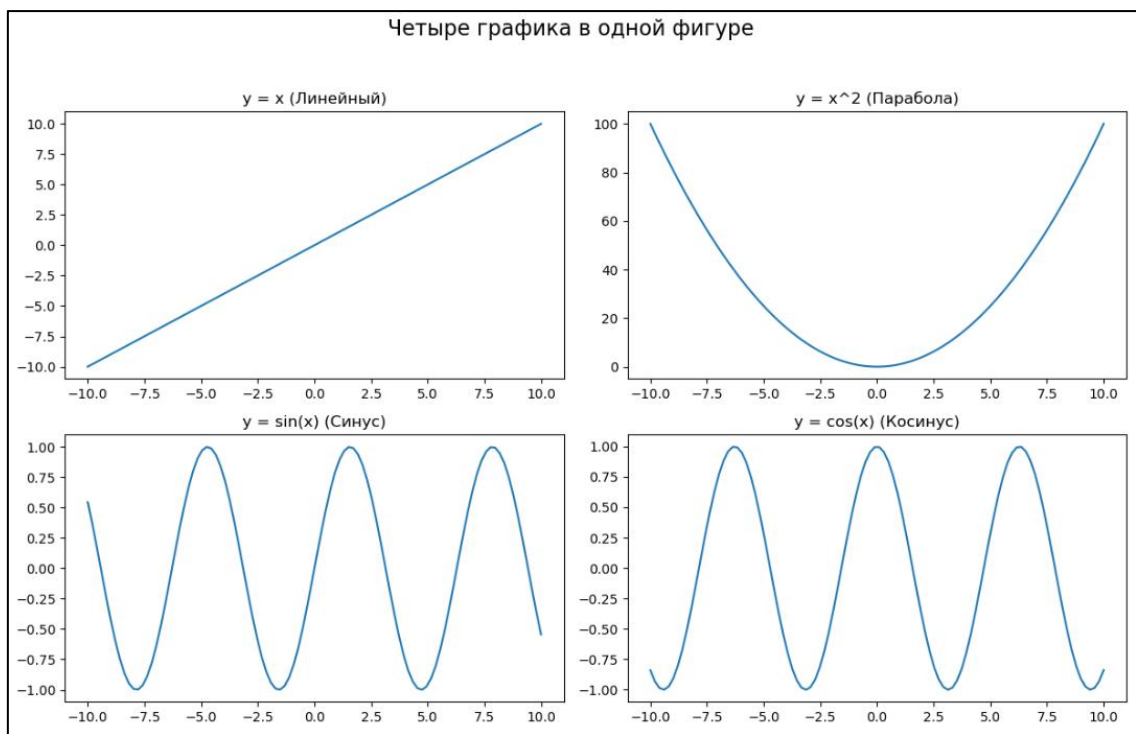


Рисунок 8.2. Графики задания №8

9.Задание 9. Создайте случайную матрицу 10×10 с элементами от 0 до 1 и визуализируйте её как тепловую карту с цветовой шкалой.

```
A = np.random.rand(10, 10)
print(A)
```

```
plt.pcolormesh(A, cmap='plasma', edgecolors="k", shading='flat')
plt.title("Тепловая карта: Цвет - X, Размер - Y")
plt.colorbar(label="Значение X (цвет)")
plt.show()
```

```
[[0.08368867 0.21796114 0.94620185 0.49940405 0.65218839 0.89237875
 0.37623505 0.86197709 0.22416463 0.16342838]
 [0.79179072 0.69574514 0.01230582 0.75189567 0.53400066 0.10180406
 0.56632252 0.7907908 0.86515281 0.1454105 ]
 [0.94709482 0.67795087 0.27764641 0.32642111 0.95239453 0.23111835
 0.00714036 0.56900181 0.11379776 0.98169638]
 [0.17906985 0.0481501 0.77791183 0.52827391 0.54176717 0.0538797
 0.38520156 0.58807426 0.8963889 0.86928814]
 [0.71347333 0.85660407 0.52718576 0.70292556 0.38316084 0.05266356
 0.65409987 0.40805806 0.95401035 0.80895504]
 [0.520685 0.03942288 0.56055457 0.85855342 0.89293304 0.10879684
 0.59767521 0.538515 0.84685655 0.26639809]
 [0.63072659 0.28959661 0.70145169 0.75547033 0.20417179 0.89036596
 0.70303174 0.99718669 0.71149405 0.07603687]
 [0.56770634 0.69252611 0.60983371 0.03788413 0.61994198 0.75908213
 0.75394721 0.17995164 0.67655374 0.78524452]
 [0.84045937 0.27613103 0.0423256 0.21907747 0.29276821 0.11371711
 0.47120454 0.02926155 0.65444364 0.47385124]
 [0.99649142 0.29482961 0.33026204 0.20606959 0.92910473 0.60878403
 0.3879633 0.77337898 0.07235881 0.37658315]]
```

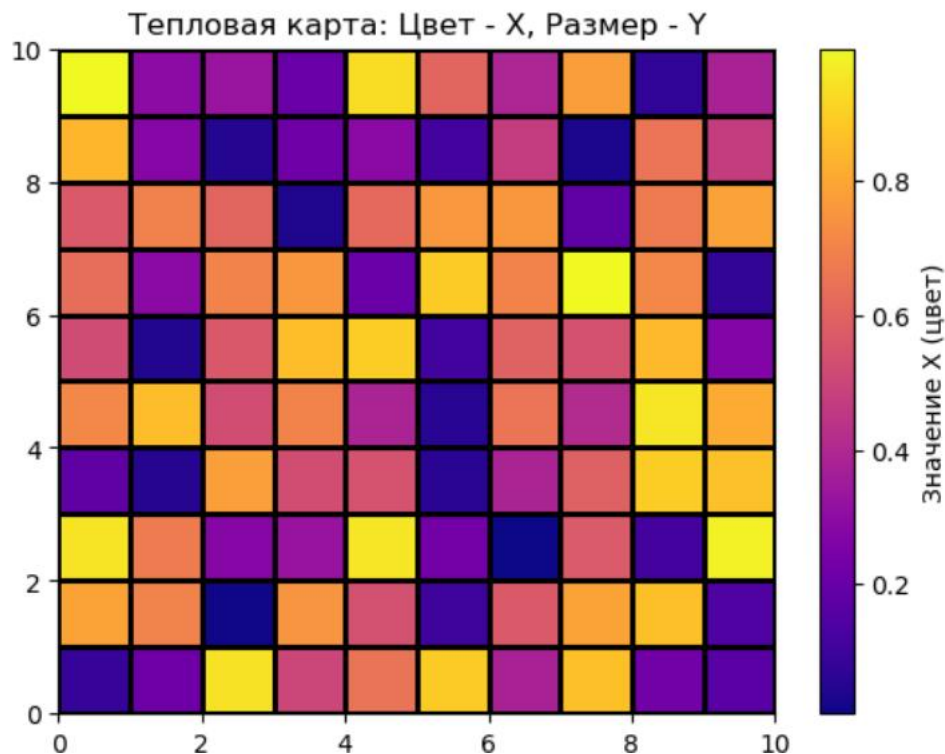


Рисунок 9. Программа для выполнения задания №9

10. Индивидуальное задание 1. построить линейный график телефонов за полгода. Добавить горизонтальную линию, показывающую среднюю оценку

```
month = ['Янв', 'Фев', 'Март', 'Апр', 'Май', 'Июнь']  
score = [4.2, 4.3, 4.1, 3.8, 3.5, 3.7]  
mean = np.mean(score)
```

```
plt.title('Рейтинг популярности смартфонов')  
plt.xlabel("x")  
plt.ylabel("y")  
plt.grid()  
plt.xlim(0, 5) # Пределы для оси x  
plt.plot(month, score)  
plt.plot([0,10], [mean, mean], '--r', lw=2)
```

[<matplotlib.lines.Line2D at 0x29f589786b0>]

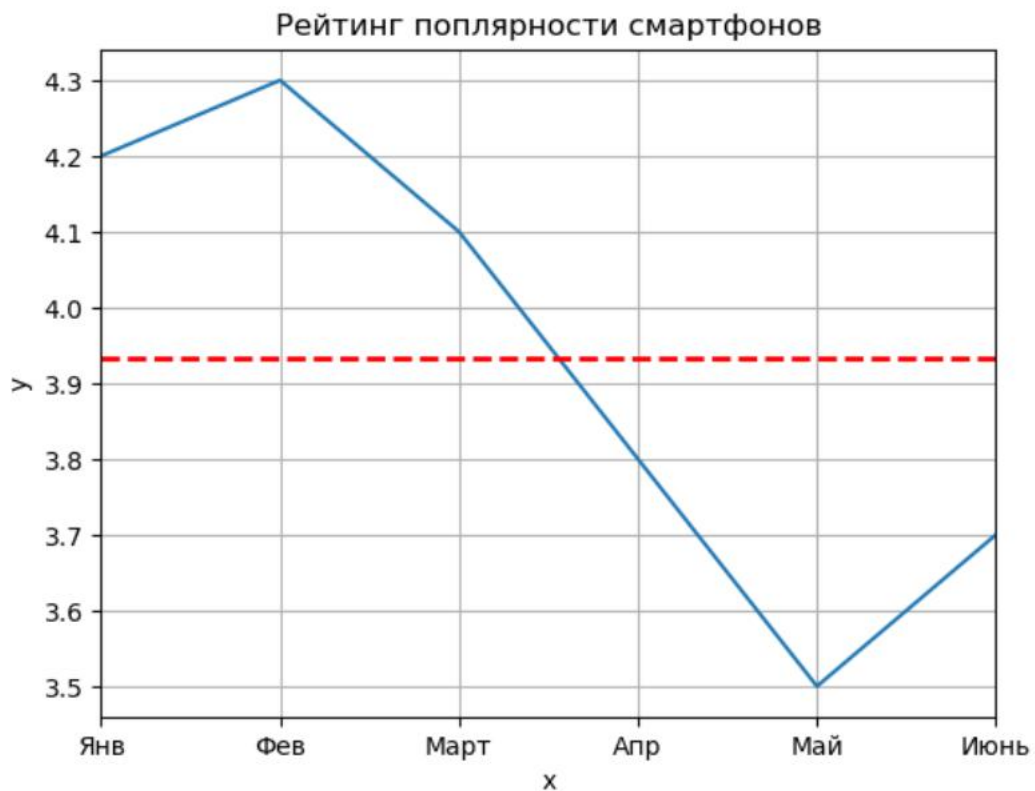


Рисунок 10. Программа для выполнения задания

11. Индивидуальное задание 2. Построить столбчатую диаграмму по количеству медалей

```
groups = ['США', 'Китай', 'Россия', 'Германия', 'Япония']  
counts = [38, 32, 28, 22, 20]  
colors=["ForestGreen", "YellowGreen", "Gold", "red"]  
plt.bar(groups,counts, color='gold')  
plt.title("количество медалей у стран на Олимпиаде")  
plt.ylabel("количество медалей")
```

```
Text(0, 0.5, 'количество медалей')
```

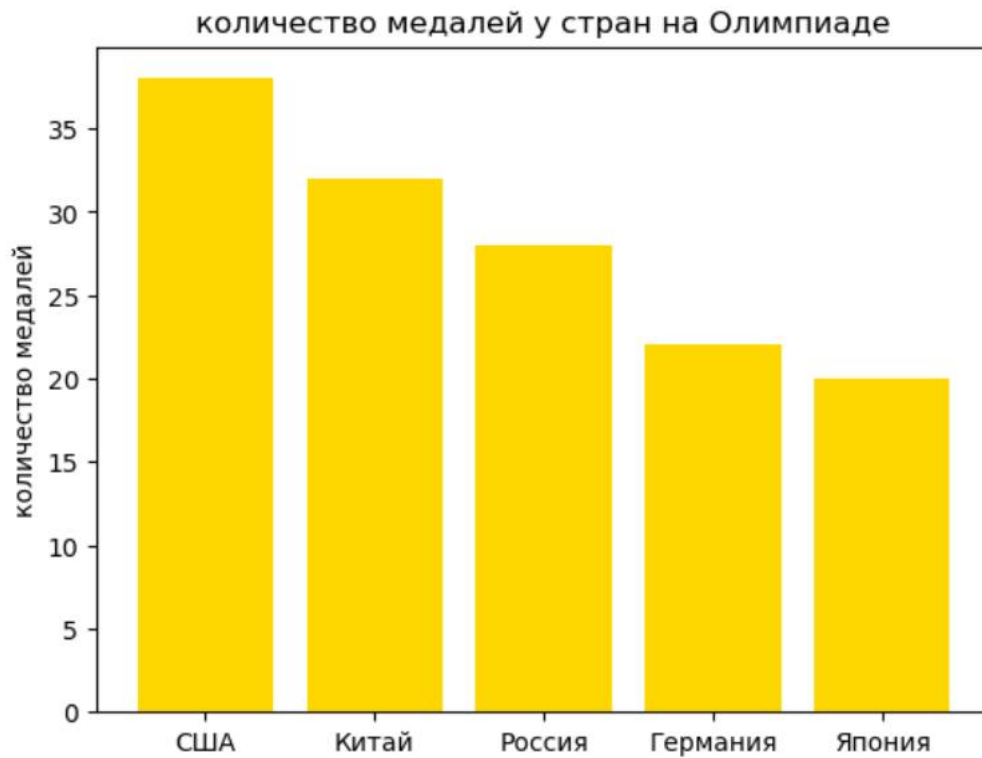


Рисунок 11. Программ для решения задания

12. Индивидуальное задание 3.

```
from scipy.integrate import quad

def f(x):
    return np.exp(2*x)
a = 0
b = 1
x = np.linspace(a-0.5, b+0.5, 100)
y = f(x)

plt.figure(figsize=(8, 6))
plt.plot(x, y, 'r', linewidth=2, label='$f(x) = e^{2x}$')

integral, error = quad(f, a, b)
print(integral)

x_fill = np.linspace(a, b, 100)
y_fill = f(x_fill)
plt.fill_between(x_fill, y_fill, color='skyblue', alpha=0.5, label='Площадь под кривой')

plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('График $f(x) = e^{2x}$ и площадь под кривой на отрезке [0, 1]')
plt.grid(True)

plt.xlim(a-0.5, b+0.5)
plt.ylim(0, f(b)+1)
plt.legend()

plt.text(0.5*(a+b), f(b)+0.5, f'Площадь = {integral:.2f}', ha='center')
plt.show()
```

Рисунок 12. Листинг для построения площади под кривой

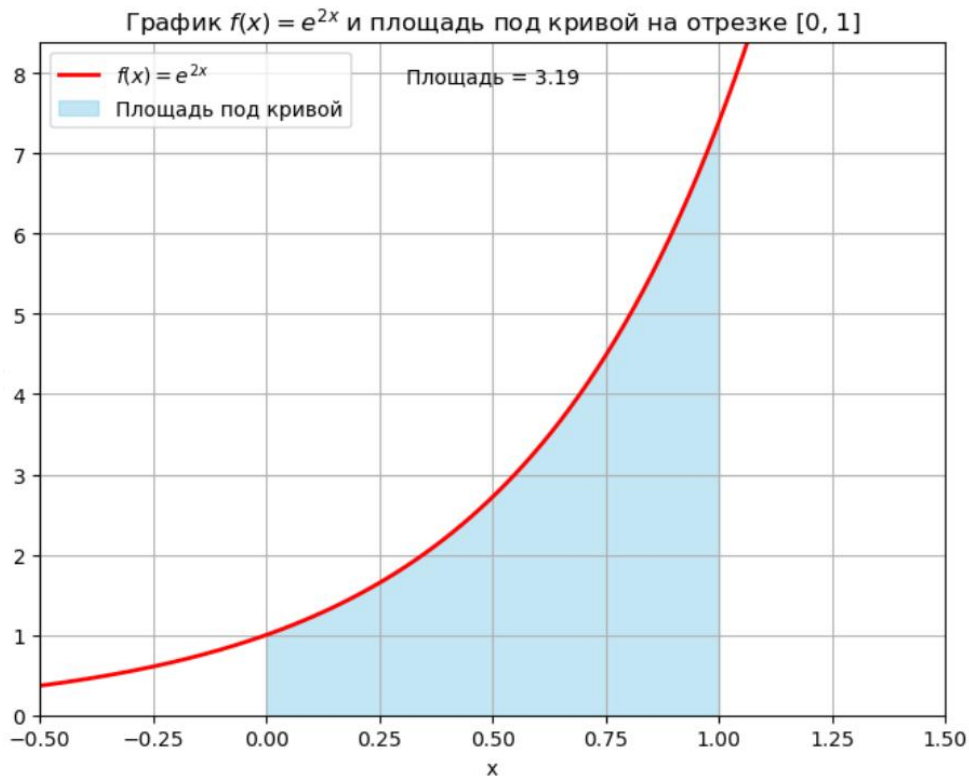


Рисунок 13. Площадь под кривой

Вывод: : в процессе выполнения лабораторной работы были освоены и применены на практике основные функции библиотеки Matplotlib, используемой для визуализации данных в Python. Изучены различные типы графиков: линейные, точечные, столбчатые, круговые, тепловые и 3D-визуализации. Также были рассмотрены методы настройки внешнего вида графиков, включая изменение цветов, стилей, добавление легенд, подписей, сетки, настройку масштаба и компоновку нескольких графиков на одном поле.

Контрольные вопросы:

1. Как осуществляется установка пакета matplotlib? Командой: `pip install matplotlib`
2. Какая "магическая" команда должна присутствовать в ноутбуках Jupyter для корректного отображения графиков? `%matplotlib inline`
3. Как отобразить график с помощью функции `plot()`? Сначала создаётся массив x , затем $y = f(x)$, и строится график: `plt.plot(x, y)` `plt.show()`
4. Как отобразить несколько графиков на одном поле? Вызвать `plot()` несколько раз до `show()`: `plt.plot(x, y1, label='график 1')` `plt.plot(x, y2, label='график 2')`
5. Какой метод известен для построения диаграмм категориальных данных? Метод `bar()` или `barh()` для столбчатых диаграмм.
6. Какие основные элементы графика вам известны? Оси, подписи осей, заголовок, легенда, сетка, линии, маркеры, цвет.
7. Как осуществляется управление текстовыми надписями на графике? Через `plt.title()`, `plt.xlabel()`, `plt.ylabel()`, `plt.text()`, `plt.annotate()`.
8. Как осуществляется управление легендой графика? Через `plt.legend()`. Опции: `loc`, `fontsize`, `title` и др.
9. Как задать цвет и стиль линий графика? Аргументы `plot()`: `plt.plot(x, y, color='red', linestyle='--', linewidth=2)`

10. Как выполнить размещение графика в разных полях? Через `plt.subplots()` или `plt.subplot()`. Например: `fig, axs = plt.subplots(2, 2)` `axs[0, 0].plot(x, y)`

11. Как выполнить построение линейного графика с помощью `matplotlib`? Через `plt.plot(x, y)`. Это по умолчанию линейный график.

12. Как выполнить заливку области между графиком и осью? Между двумя графиками? Используется `plt.fill_between()`: `plt.fill_between(x, y, color='lightblue')`

13. Как выполнить выборочную заливку, которая удовлетворяет некому условию? Через условие в `fill_between()`: `plt.fill_between(x, y, where=(y>0), color='green')`

14. Как выполнить двухцветную заливку? Несколько вызовов `fill_between()` с разными условиями.

15. Как выполнить маркировку графиков? Через параметр `label` в `plot()`, и `plt.legend()`.

16. Как выполнить обрезку графиков? Через `plt.xlim()` и `plt.ylim()`, например: `plt.xlim(-5, 5)`

17. Как построить ступенчатый график? В чем особенность? Используется `plt.step()`. Он отображает данные в виде "лестницы", полезен для дискретных изменений.

18. Как построить стекированный график? Используется `plt.stackplot()`: `plt.stackplot(x, y1, y2, labels=['y1', 'y2'])`

19. Как построить `stem`-график? Через `plt.stem()`. Он показывает точки и вертикальные линии от оси.

20. Как построить точечный график? Через `plt.scatter(x, y)`. Особенность: визуализирует распределение точек по координатам.

21. Как осуществляется построение столбчатых диаграмм? С помощью `plt.bar(x, height)` или `plt.barh()`.

22. Что такое групповая столбчатая диаграмма? Что такое диаграмма с errorbar? Групповая: несколько столбцов рядом для одной категории. errorbar: диаграмма с отображением погрешности (plt.errorbar).

23. Как выполнить построение круговой диаграммы? plt.pie(values, labels=..., autopct='%1.1f%%')

24. Что такое цветовая карта? Как работать с цветами в matplotlib? Цветовая карта (colormap) — отображает числовые значения в цвет. Пример: plt.imshow(data, cmap='hot')

25. Как отобразить изображение средствами matplotlib? Через plt.imshow(img), где img — массив (например, из OpenCV или PIL).

26. Как отобразить тепловую карту средствами matplotlib? plt.imshow(matrix, cmap='hot') или через seaborn. Добавляют plt.colorbar() для шкалы.

27. Как выполнить построение линейного 3D-графика? Через ax.plot3D(x, y, z), где ax — subplot(111, projection='3d').

28. Как построить точечную 3D-диаграмму? ax.scatter3D(x, y, z)

29. Как выполнить построение каркасной поверхности? ax.plot_wireframe(X, Y, Z)

30. Как построить трёхмерную поверхность? ax.plot_surface(X, Y, Z, cmap='viridis')