

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
«Искусственный интеллект и машинное обучение»
Вариант 5

Выполнил:

Беков Шамиль Расулович
2 курс, группа ИВТ-б-о-23-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:

Доцент департамента цифровых,
робототехнических систем и электроники
института перспективной инженерии
Воронкин Роман Александрович

(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.

Github: <https://github.com/ShamilBekov/Laba-3/tree/main>

Порядок выполнения работы:

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

```
[49]: import matplotlib.pyplot as plt
      %matplotlib inline
      import numpy as np
      x=np.linspace(-10,10)
      y=x**2
      plt.plot(x,y)
      plt.title("График функции $y=x^2$")
      plt.xlabel("x")
      plt.ylabel("y")
      plt.grid(True)
      plt.show()
```

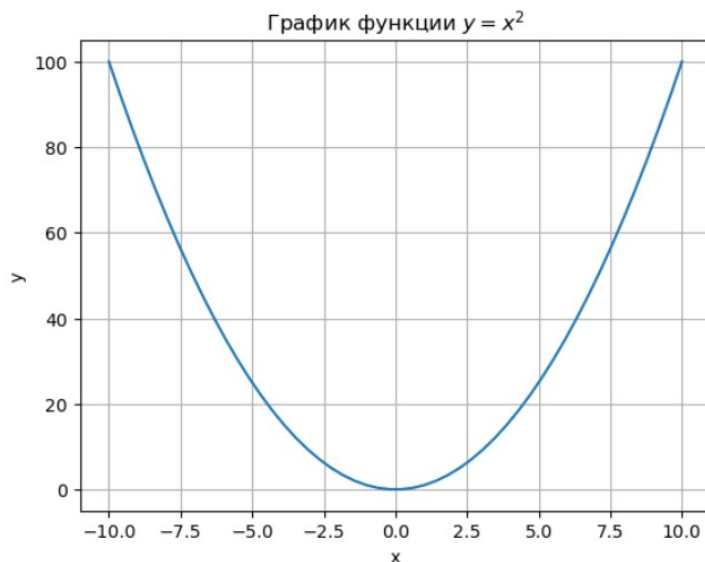


Рисунок 1. Первое задание

2. Построили три линии на одном графике:

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

Добавили легенду и сделали оси одинакового масштаба.

```
[53]: x=np.linspace(-1,1)
y=x
y1=x**2
y2=x**3
plt.plot(x, y, "b--", label="$y=x$")
plt.plot(x, y1, "g-.", label="$y=x^2$")
plt.plot(x, y2, "r-", label="$y=x^3$")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Графики функций $y=x, y=x^2, y=x^3$")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.show()
```

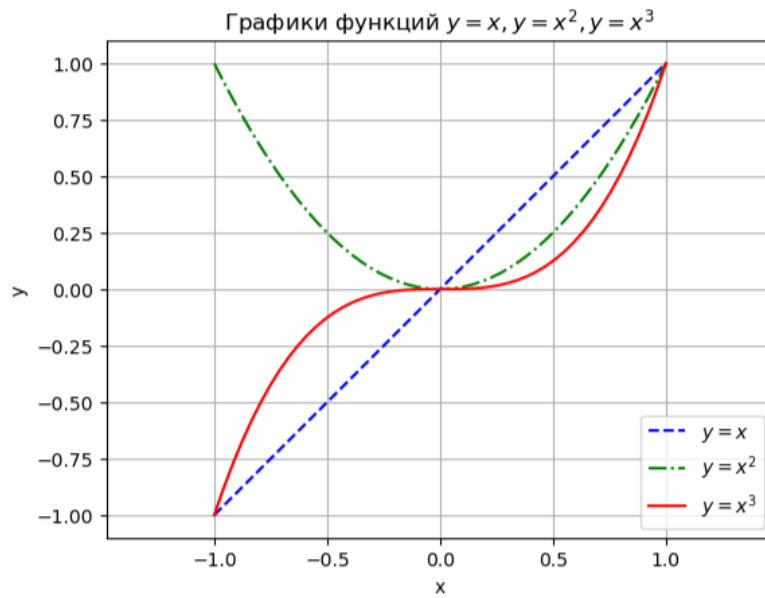


Рисунок 2. Второе задание

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

```
[245]: x = np.random.uniform(-10, 10, 50)
y = np.random.uniform(-10, 10, 50)
colors = x
sizes = (y - min(y)) / (max(y) - min(y)) * 200 + 50
plt.figure(figsize=(8, 6))
plt.scatter(x, y, c=colors, s=sizes, cmap='viridis', edgecolors='k')
plt.title('Диаграмма рассеяния (scatter plot)')
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.grid()
plt.colorbar(label='$x$')
plt.show()
```

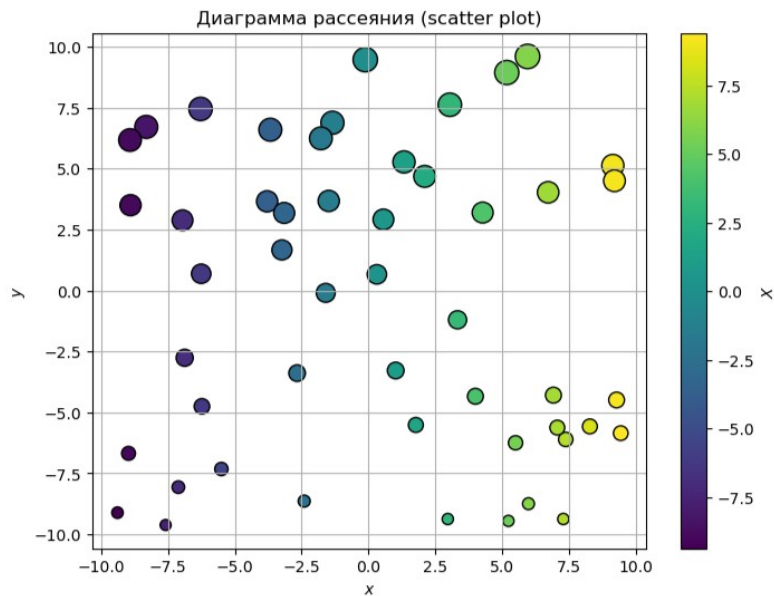


Рисунок 3. Третье задание

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.

```
[81]: norm = np.random.normal(0,1,1000)
plt.hist(norm,bins=30,color='blue',edgecolor='black',alpha=0.7)
plt.axvline(np.mean(norm), color='red')
plt.show()
```

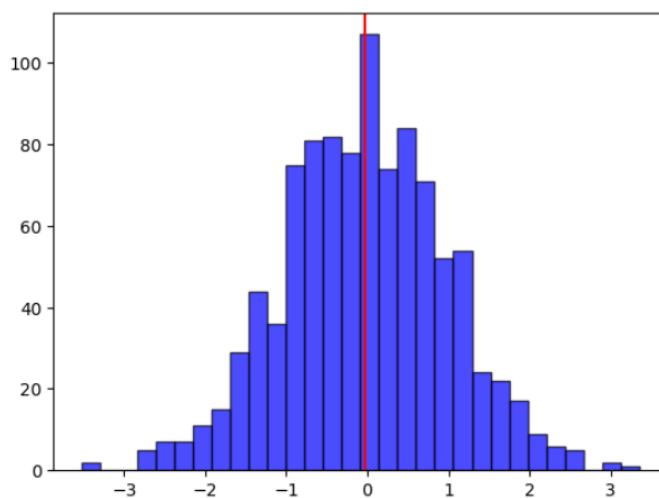


Рисунок 4. Задание 4

5. Построили столбчатую диаграмму, которая показывает количество студентов, получивших оценки:

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

Добавили подписи к осям и заголовок.

```
[87]: di = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]  
ch = [20, 35, 30, 15]  
plt.bar(di, ch, color=['green', 'yellow', 'orange', 'red'])  
plt.xlabel("Оценки")  
plt.ylabel("Количество студентов")  
plt.title("Успеваемость студентов")  
plt.show()
```

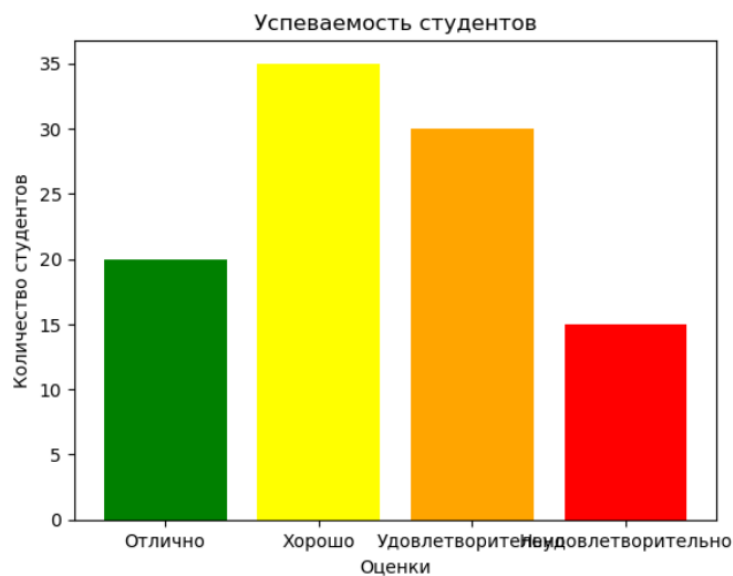


Рисунок 5. Пятое задание

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

```
[89]: di = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
ch = [20, 35, 30, 15]
plt.figure(figsize=(6,6))
plt.pie(ch, labels=di, autopct='%1.1f%%', colors=['green', 'yellow', 'orange', 'red'], startangle=140)
plt.title("Успеваемость студентов")
plt.show()
```

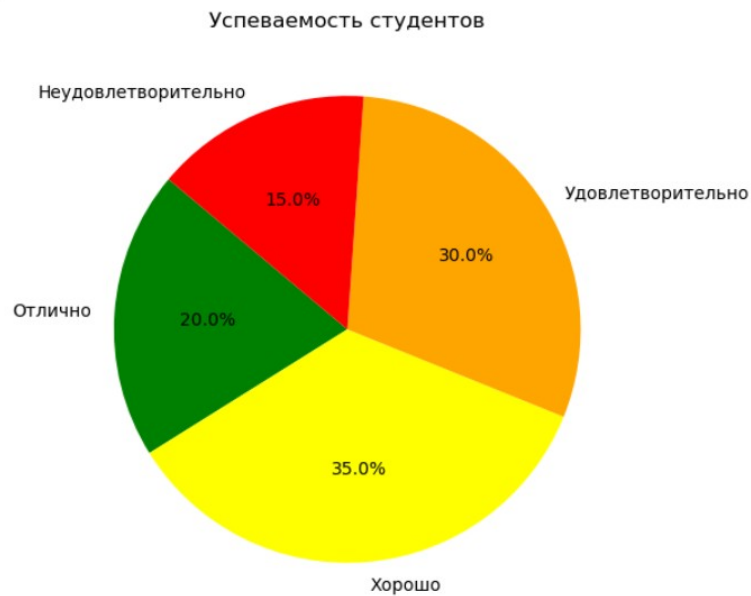


Рисунок 6. Шестое задание

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

```
[183]: from mpl_toolkits.mplot3d import Axes3D
x = np.linspace(-5,5,150)
y = np.linspace(-5,5,150)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))
fig = plt.figure(figsize=(12,9))
a = fig.add_subplot(111,projection='3d')
a.plot_surface(X,Y,Z,cmap='viridis')
a.set_xlabel('x')
a.set_ylabel('y')
a.set_zlabel('z')
a.set_title("3D-график  $z = \sin(\sqrt{x^2 + y^2})$ ")
plt.show()
```

3D-график $z = \sin(\sqrt{x^2 + y^2})$

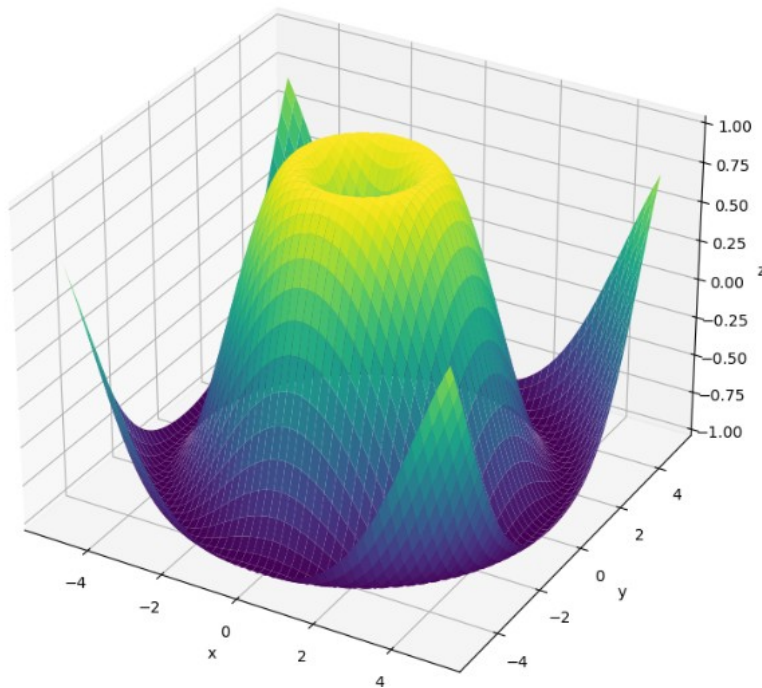


Рисунок 7. Седьмое задание

8. Построили четыре графика в одной фигуре (2 × 2 сетка):

- Линейный график $y = x$,
- Парабола $y = x^2$,
- Синус $y = \sin(x)$,
- Косинус $y = \cos(x)$.

Добавили подписи к осям и заголовков.

```
[185]: x = np.linspace(-10,10)
fig, axes = plt.subplots(2,2,figsize=(12,10))
axes[0,0].plot(x,x,color='red')
axes[0,0].set_title('y = x')
axes[0,1].plot(x,x**2,color='blue')
axes[0,1].set_title('y = x^2')
axes[1,0].plot(x,np.sin(x),color='yellow')
axes[1,0].set_title('y = sin(x)')
axes[1,1].plot(x,np.cos(x),color='orange')
axes[1,1].set_title('y = cos(x)')
plt.show()
```

Рисунок 8. Код восьмого задания

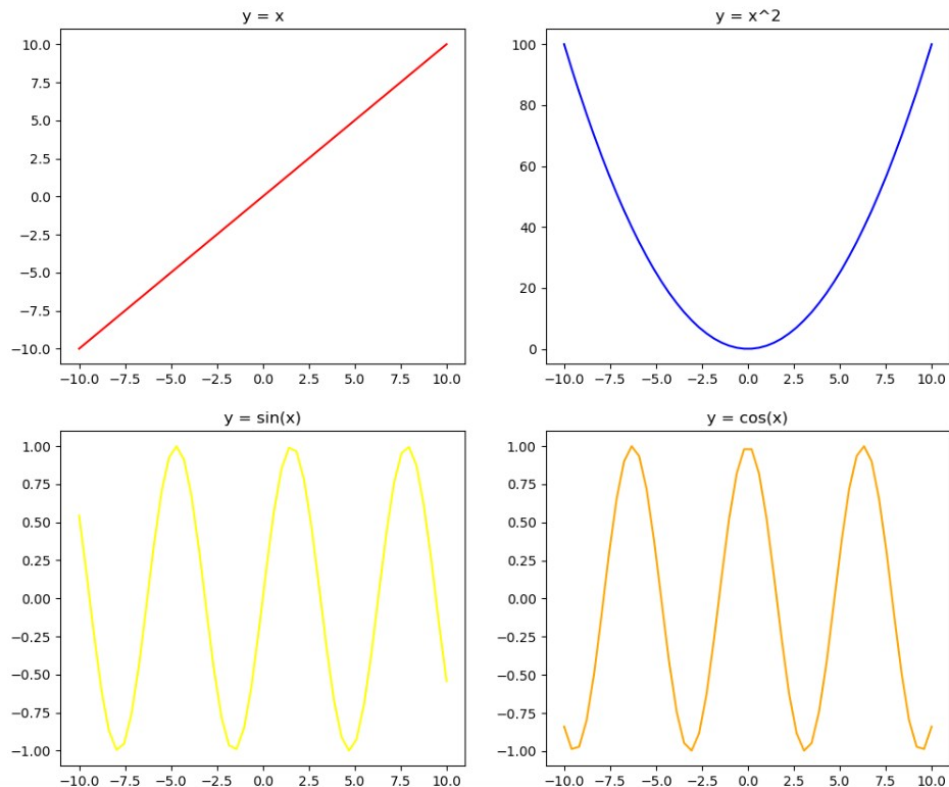


Рисунок 9. Графики восьмого задания

9. Создали случайную матрицу 10×10 с элементами от 0 до 1 и визуализировали её как тепловую карту с цветовой шкалой.

```
[195]: matrix = np.random.rand(10,10)
plt.figure(figsize=(10, 8))
plt.imshow(matrix, cmap='viridis', aspect='auto')
plt.colorbar()
plt.title("Тепловая карта (imshow)")
plt.show()
```

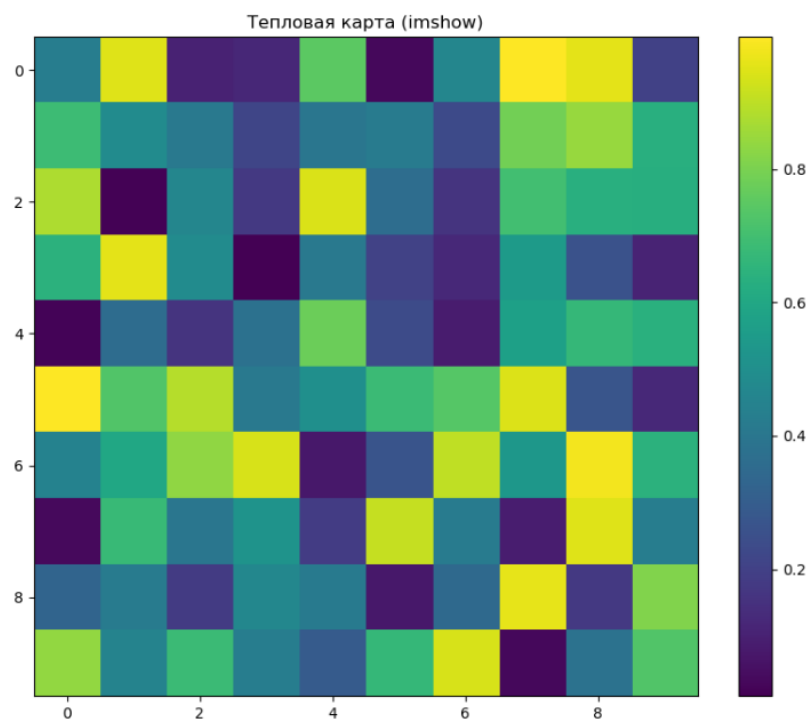


Рисунок 10. Девятое задание

10. На протяжении года измерялся уровень воды в озере:

- Месяцы: ['Янв', 'Фев', 'Март', 'Апр', 'Май', 'Июнь', 'Июль', 'Авг', 'Сен', 'Окт', 'Нояб', 'Дек'],
- Уровень воды (м): [2.3, 2.4, 2.6, 2.8, 3.0, 3.1, 3.2, 3.1, 3.0, 2.8, 2.6, 2.4].

Используйте синий цвет линии и добавьте точки с маркерами.

Построили график, обозначили оси и использовали маркеры на точках.

```
1]: import matplotlib.pyplot as plt

months = ['Янв', 'Фев', 'Март', 'Апр', 'Май', 'Июнь', 'Июль', 'Авг', 'Сен', 'Окт', 'Нояб', 'Дек']
water_level = [2.3, 2.4, 2.6, 2.8, 3.0, 3.1, 3.2, 3.1, 3.0, 2.8, 2.6, 2.4]
plt.figure(figsize=(10, 6))
plt.plot(months, water_level, color='blue', marker='o', markersize=8)

plt.title('Уровень воды в озере по месяцам')
plt.xlabel('Месяцы')
plt.ylabel('Уровень воды (м)')
plt.grid(True)
plt.ylim(2.2, 3.3)

for i, value in enumerate(water_level):
    plt.text(i, value + 0.05, f'{value} м', ha='center')

plt.show()
```

Рисунок 11. Выполнение первого индивидуального задания

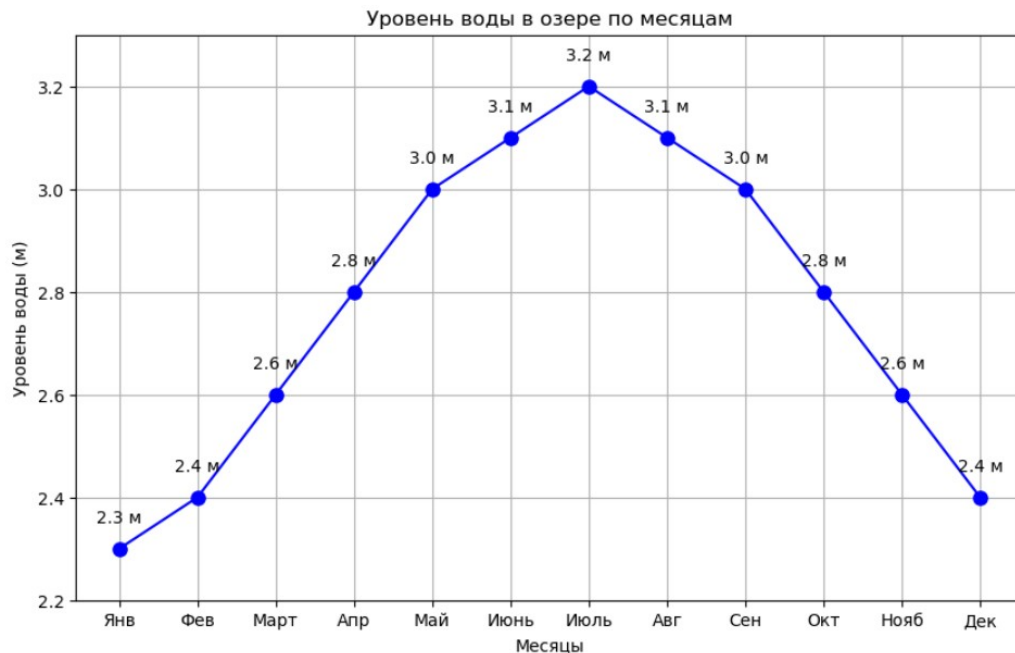


Рисунок 12. Первое индивидуальное задание

11. Средняя температура в разные сезоны::

- Сезоны: ['Зима', 'Весна', 'Лето', 'Осень'],
- Температура (C): [-5, 10, 25, 12].

Используйте разные цвета для каждого сезона.

Построили горизонтальную столбчатую диаграмму. Добавили подписи к столбцам.

```
[45]: seasons = ['Зима', 'Весна', 'Лето', 'Осень']
temperature = [-5, 10, 25, 12]
colors = ['blue', 'yellow', 'red', 'orange']
plt.figure(figsize=(8, 6))
bars = plt.bar(seasons, temperature, color=colors, edgecolor='black', linewidth=1.2)
plt.title('Среднемесячная температура по сезонам')
plt.xlabel('Сезоны')
plt.ylabel('Температура (°C)')
plt.grid(axis='y', linestyle='--')
plt.ylim(-10, 30)
plt.show()
```

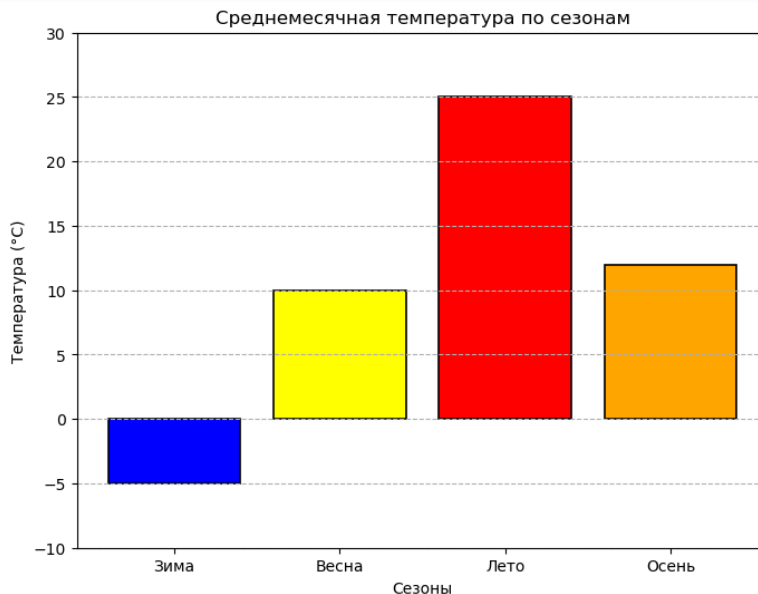


Рисунок 13. Второе индивидуальное задание

12. Вычислили площадь под функцией $f(x) = \frac{1}{x}$ на интервале $[1, 5]$.

Закрасили соответствующую область.

```
[251]: from scipy.integrate import quad
def f(x):
    return 1 / x

result, error = quad(f, 1, 5)
print(f"Площадь под кривой: {result:.4f}")
x = np.linspace(0.2, 6, 500)
y = f(x)
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'r-', linewidth=2, label=r'$f(x) = \frac{1}{x}$')
x_fill = np.linspace(1, 5, 100)
y_fill = f(x_fill)
plt.fill_between(x_fill, y_fill, color='skyblue', alpha=0.4, label=f'Площадь = {result:.4f}')

plt.title('Площадь под гиперболой на интервале [1, 5]', fontsize=14)
plt.xlabel('x')
plt.ylabel('f(x)')
plt.axhline(0, color='black')
plt.axvline(0, color='black')
plt.grid(linestyle='--')
plt.xlim(0, 6)
plt.ylim(0, 3)
plt.show()
```

Площадь под кривой: 1.6094

Рисунок 14. Выполнение индивидуального задания

Площадь под кривой: 1.6094

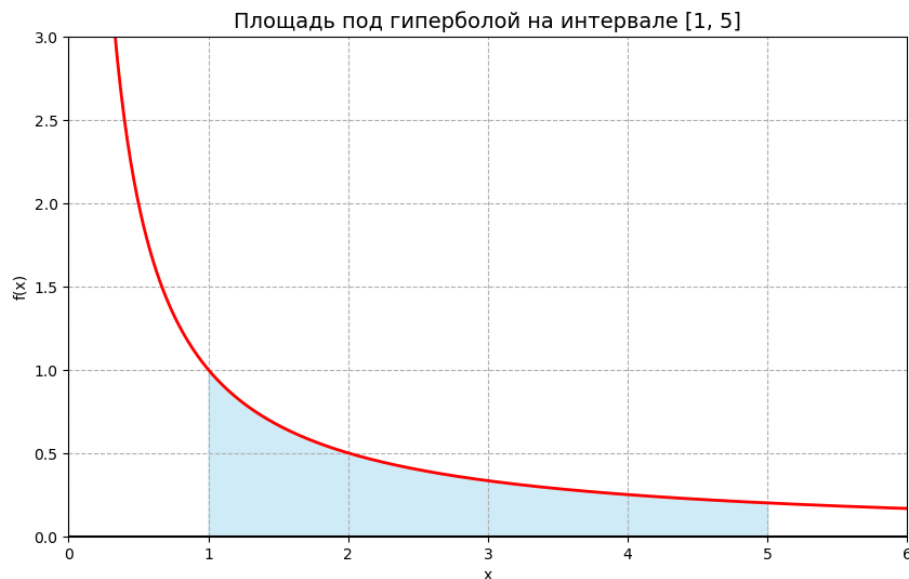


Рисунок 14. Результат выполнения индивидуального задания

13. Вычислили площадь под графиком функции $f(x) = \sqrt{x^2 - y^2}$ в диапазонах $x, y \in [-5, 5]$.

Требуется:

- Построить трехмерный график функции $f(x, y)$ в заданных пределах,
- Использовать библиотеку Matplotlib для визуализации,
- Оформить график: добавить заголовок, подписи осей и цветовую карту (если уместно).

```
[249]: from mpl_toolkits.mplot3d import Axes3D

x = np.linspace(-5, 5)
y = np.linspace(-5, 5)
X, Y = np.meshgrid(x, y)
Z = np.sqrt(X**2 + Y**2)
fig = plt.figure(figsize=(12, 10))
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none', rstride=2, cstride=2)

ax.set_title('Коническая поверхность:  $f(x, y) = \sqrt{x^2 + y^2}$ ')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

fig.colorbar(surf, shrink=0.5, aspect=10, label='Высота')

ax.view_init(elev=30, azim=45)

plt.show()
```

Рисунок 15. Выполнение четвертого индивидуального задания

Коническая поверхность: $f(x, y) = \sqrt{x^2 + y^2}$

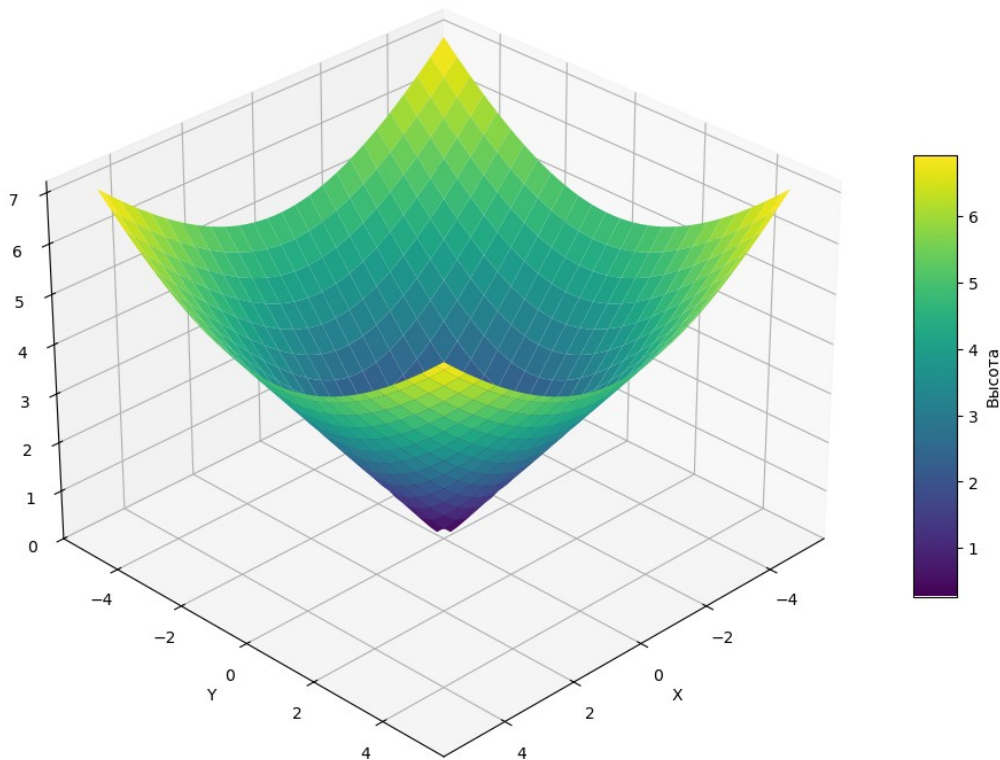


Рисунок 15. Четвертое индивидуальное задание

Контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Через команду `pip install matplotlib`.

2. Какая "магическая" команда должна присутствовать в ноутбуках Jupyter для корректного отображения графиков matplotlib?

Через команду `%matplotlib inline`.

3. Как отобразить график с помощью функции plot?

Через команду `plt.plot` или `plt.show()`.

4. Как отобразить несколько графиков на одном поле?

Используется передача нескольких наборов данных в `plt.plot()` или метод `plt.subplot()`.

Разница между методами:

- В первом случае оба графика отображаются в одном поле.
- Во втором случае создаются две области (`subplot`).

5. Какой метод Вам известен для построения диаграмм категориальных данных?

Для построения диаграмм категориальных данных используется метод `bar()`.

6. Какие основные элементы графика Вам известны?

Основные элементы графика `matplotlib`:

- Оси (Axes) – координатные оси графика.
- Функция (Plot) – линия, соединяющая точки.
- Заголовок (Title) – название графика.
- Подписи осей (`xlabel`, `ylabel`) – названия осей X и Y.
- Легенда (Legend) – описание кривых на графике.
- Сетка (Grid) – вспомогательные линии для удобства чтения

данных.

- Маркировка (Ticks) – деления на осях X и Y.

7. Как осуществляется управление текстовыми надписями на графике?

Управление текстом выполняется с помощью следующих функций:

- `plt.xlabel("Название оси X", fontsize=«Размер»)` – подпись оси X.
- `plt.ylabel("Название оси Y", fontsize=«Размер»)` – подпись оси Y.
- `plt.title("Заголовок", fontsize=«Размер»)` – заголовок графика.
- `plt.text(x, y, "Текст", fontsize=«Размер»)` – произвольный текст в

точке (x, y).

8. Как осуществляется управление легендой графика?

Легенда добавляется с помощью `plt.legend()` и имеет несколько параметров:

- `loc="upper left"` – положение (верхний левый угол).
- `fontsize=10` – размер шрифта.
- `title="Обозначения"` – заголовок легенды.
- `title_fontsize=12` – размер шрифта заголовка

9. Как задать цвет и стиль линий графика?

Цвет и стиль задаются аргументами `color`, `linestyle`, `marker` в `plt.plot()`.

10. Как выполнить размещение графика в разных полях?

Размещение нескольких графиков выполняется с помощью `plt.subplots()`.

11. Как выполнить построение линейного графика с помощью `matplotlib`?

Линейный график строится с помощью функции `plot()`:

Построение графика,

```
plt.plot(x, y, color='blue', linestyle='-', marker='o', label="Линейный график").
```

12. Как выполнить заливку области между графиком и осью? Между двумя графиками?

Используется `fill_between()`:

Например: `fill_between(x, y, y2, where=(y > y2))` – заливает область между $\sin(x)$ и $\cos(x)$, где $\sin(x) > \cos(x)$.

13. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию?

Можно использовать `where` в `fill_between()`:

Например:

- `where=(y > 0)` – заливает область только там, где $y > 0$ (зеленым).
- `where=(y < 0)` – заливает область только там, где $y < 0$ (красным).

14. Как выполнить двухцветную заливку?

```
plt.fill_between(x, y, where=(y > 0), color='blue', alpha=0.3) # Верхняя часть.
```

```
plt.fill_between(x, y, where=(y < 0), color='red', alpha=0.3) # Нижняя часть.
```

15. Как выполнить маркировку графиков?

Маркировка (метки) выполняется с помощью `text()` и `annotate()` (добавляет метку с указателем).

16. Как выполнить обрезку графиков?

Графики можно обрезать с помощью `xlim()` и `ylim()`.

Например:

- `plt.plot(x, y)`
- `plt.xlim(2, 8)` # Ограничение оси X от 2 до 8.

17. Как построить ступенчатый график? В чем особенность ступенчатого графика?

Ступенчатый график строится с помощью `plt.step()`. Он отображает изменения дискретных данных, где значения остаются постоянными до определенного момента.

Особенности:

- Значение остается постоянным до изменения.
- Используется для дискретных процессов (например, квантованные сигналы, изменение состояний).
- Параметр `where` задает тип отрисовки ('pre', 'mid', 'post').

18. Как построить стековый график? В чем особенность стекового графика?

Стековый (Stacked) график – это несколько графиков, сложенных друг на друга (`plt.stackplot()`). Используется для отображения накопленных данных.

Особенности:

- Значения накапливаются, создавая суммарный эффект.
- Позволяет видеть вклад каждой категории в общий результат.
- Часто используется в экономике, финансах, статистике.

19. Как построить stem-график? В чем особенность stem-графика?

Stem-график (диаграмма стебель-лист) (`plt.stem()`) – это вертикальные линии, обозначающие дискретные точки.

Особенности:

- Подходит для дискретных данных.
- Используется для анализов сигналов, цифровых измерений.
- Показывает изолированные значения вместо непрерывной линии.

20. Как построить точечный график? В чем особенность точечного графика?

Точечный график (scatter plot) (`plt.scatter()`) показывает распределение точек без соединения линиями.

Особенности:

- Используется для анализа взаимосвязей между переменными.
- Позволяет разброс точек по координатам.
- Полезен в статистике, машинном обучении.

21. Как осуществляется построение столбчатых диаграмм с помощью `matplotlib`?

Столбчатая диаграмма (`plt.bar()`) отображает категориальные данные.

Например:

```
# Данные
```

```
categories = ["A", "B", "C", "D"]
```

```
values = [15, 30, 45, 10]
```

```
# Построение столбчатой диаграммы
```

```
plt.bar(categories, values, color=['blue', 'green', 'red', 'purple'])
```

22. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с `errorbar` элементом?

Групповая столбчатая диаграмма используется для сравнения нескольких категорий внутри одной группы. В `matplotlib` это достигается с помощью `plt.bar()` с разными смещениями по оси X.

`errorbar` показывает разброс данных с указанием доверительных интервалов или погрешностей.

23. Как выполнить построение круговой диаграммы средствами `matplotlib`?

Круговая диаграмма (`plt.pie()`) используется для отображения долей категорий.

```
autopct='%-' – добавляет проценты.
```

24. Что такое цветовая карта? Как осуществляется работа с цветовыми картами в `matplotlib`?

Цветовая карта (colormap) – это градиент цветов, используемый для отображения данных на графиках (heatmap, 3D-графики, scatter и др.).

25. Как отобразить изображение средствами matplotlib?

Для отображения изображений используется plt.imshow(). Чаще всего изображения загружаются через PIL или OpenCV.

- mpimg.imread('image.jpg') – загружает изображение.
- imshow(img) – отображает изображение.
- axis("off") – скрывает оси.

Дополнительно:

Если изображение в формате NumPy ((height, width, 3)), его можно напрямую передать в imshow().

26. Как отобразить тепловую карту средствами matplotlib?

Тепловая карта (heatmap) создается с помощью imshow() и цветовой карты cmap.

27. Как выполнить построение линейного 3D-графика с помощью matplotlib?

Линейный 3D-график создается с помощью plot3D().

28. Как выполнить построение точечного 3D-графика с помощью matplotlib?

Точечный 3D-график создается с помощью scatter3D().

29. Как выполнить построение каркасной поверхности с помощью matplotlib?

Каркасная поверхность (wireframe) используется для отображения структуры 3D-объекта.

plot_wireframe() строит скелетную модель поверхности.

30. Как выполнить построение трехмерной поверхности с помощью matplotlib?

Полная 3D-поверхность (plot_surface()) строится с заливкой цветом.

Вывод: в ходе лабораторной работы были исследованы базовые возможности библиотеки NumPy языка программирования Python.