

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 № 4

Выполнил:
Левашев Тимур Рашидович
2 курс, группа ИВТ-б-о-23-2,
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 В.И

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема работы: “Основы работы с библиотекой matplotlib”

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

Порядок выполнения работы:

Ссылка на Git репозиторий: https://github.com/mazy99/ml_prakt_3

1. Построение простого графика.

```
import matplotlib.pyplot as plt
import numpy as np
x_array = np.linspace(-10,10,50)
y_array = x_array**2
plt.figure(figsize=(5,5))
plt.plot(x_array,y_array)
plt.title("График функции  $y = x^2$ ")
plt.grid(True)
plt.ylabel('Y')
plt.xlabel('X')
plt.show()
```

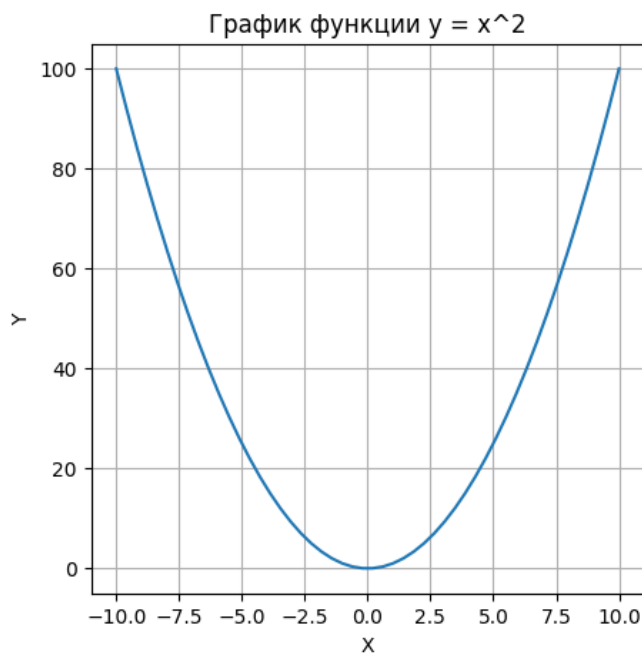


Рисунок 1 – Код и графика к заданию 1

2. Настройка стилей и цветов.

```
x_array = np.linspace(-10,10,50)
y1_array = x_array
y2_array = x_array**2
y3_array = x_array**3

x_min, x_max = -10, 10
y_min, y_max = -1000, 1000

plt.figure(figsize=(10,10))

plt.plot(x_array,y1_array,label="y = x", color = 'blue', linestyle='--')
plt.plot(x_array,y2_array, label="y = x^2", color = 'green', linestyle='-.')
plt.plot(x_array,y3_array,label="y = x^3",color = 'red', linestyle=None)
plt.title("Графики функций")
plt.grid(True)
plt.ylabel('Y')
plt.xlabel('X')
plt.legend()
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.gca().set_aspect(1.0 / plt.gca().get_data_ratio(), adjustable='box')

plt.show()
```

Рисунок 2 – Код к заданию 2

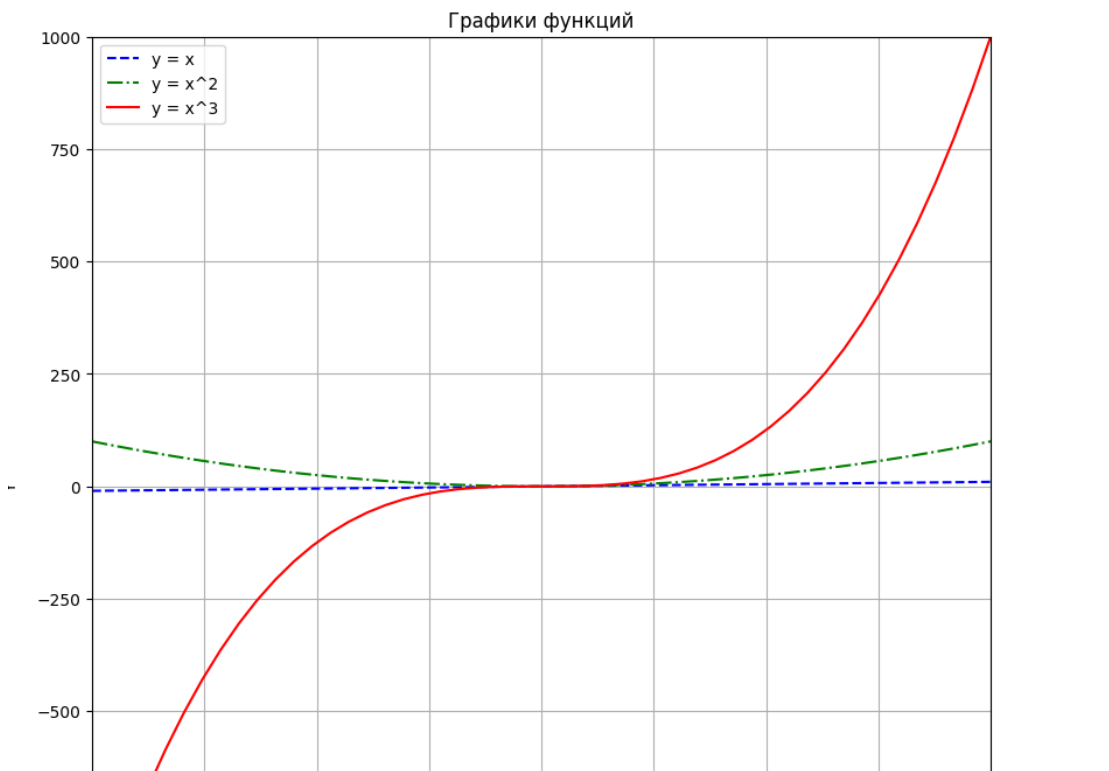


Рисунок 3 – Графика к заданию 2

3. Использование различных типов графики.

```
] x = np.random.rand(50)*10
y = np.random.rand(50)*10
colors = y
sizes = y * 100

plt.figure(figsize=(8, 6))
plt.scatter(x, y, c=colors, s=sizes, alpha=0.7, cmap='viridis')
plt.title("Диаграмма рассеяния с цветом и размером точек")
plt.xlabel('X')
plt.ylabel('Y')

plt.colorbar(label='Значение по оси X')

plt.show()
```

Рисунок 4 – Код к заданию 3

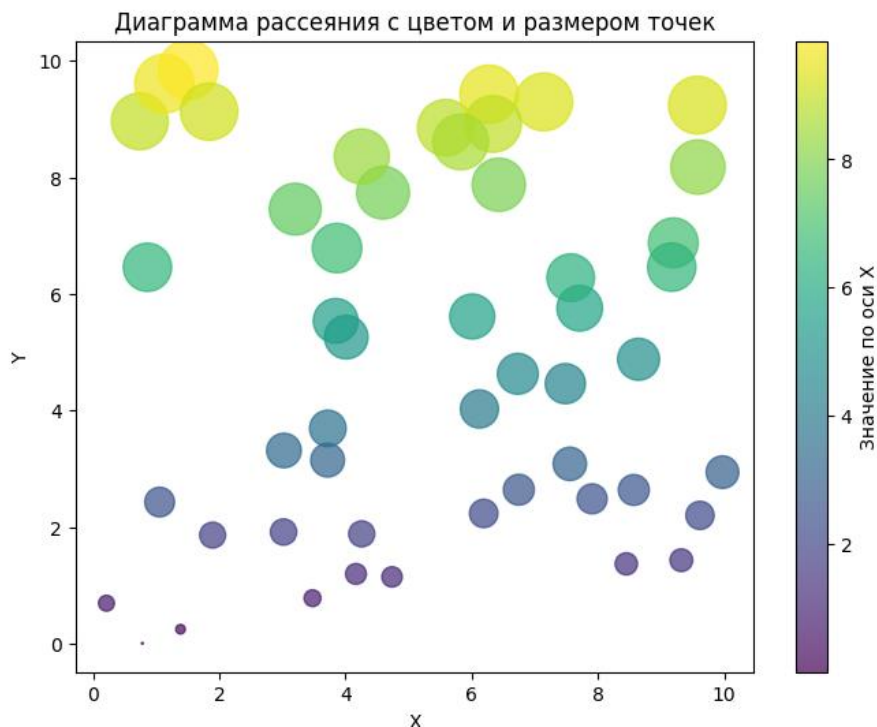


Рисунок 5 – Графика к заданию 3

4. Гистограмма распределения.

```
[72]: data = np.random.normal(0, 1, 1000)
plt.hist(data, bins=30, color='skyblue', edgecolor='black', alpha=0.7)
mean = np.mean(data)
plt.axvline(mean, color='red', linestyle='--', label=f'Среднее: {mean:.2f}')
plt.title("Гистограмма с вертикальной линией на среднем значении")
plt.xlabel("Значения")
plt.ylabel("Частота")
plt.legend()
plt.show()
```

Рисунок 6 – Код к заданию 4

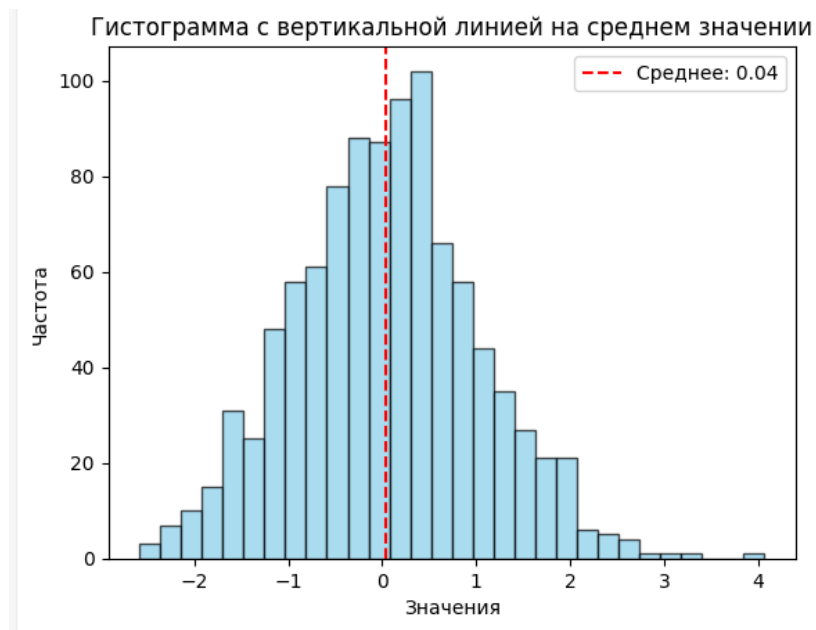


Рисунок 7 – Графика к заданию 4

5. Столбчатая диаграмма

```
96]: students = {"great": 20,"good":35,"satisfactorily": 30, "unsatisfactory":15}
categories = list(students.keys())
values = list(students.values())
plt.title('Распределение студентов по оценкам')
plt.bar(categories,values,color='skyblue')
plt.xlabel("assessments")
plt.ylabel("students")

plt.show()
```

Рисунок 8 – Код к заданию 5

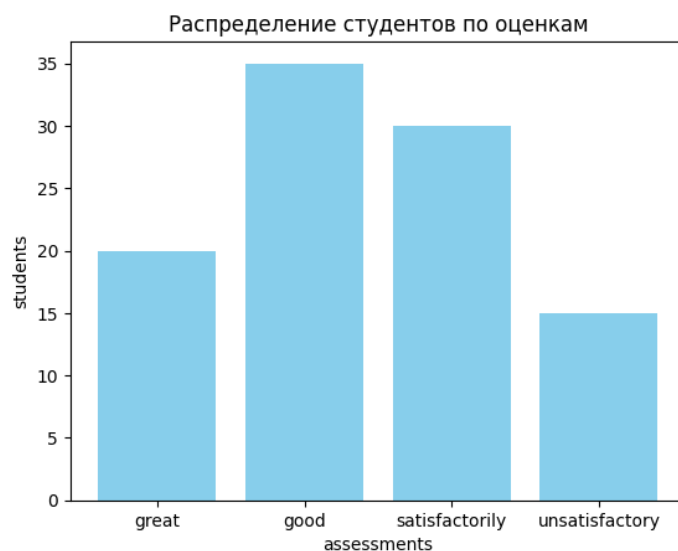


Рисунок 9 – Графика к заданию 5

6. Круговая диаграмма.

```
3]: students = {"great": 20, "good": 35, "satisfactorily": 30, "unsatisfactory": 15}
categories = list(students.keys())
values = list(students.values())
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99']
plt.title('Распределение студентов по оценкам')
plt.pie(values, labels=values, autopct='%1.1f%%', colors=colors, startangle=90)
plt.axis('equal')

plt.show()
```

Рисунок 10 – Код к заданию 6



Рисунок 11 – Графика к заданию 6

7. Трехмерный график поверхности.

```
20]: from mpl_toolkits.mplot3d import Axes3D
x_arr = np.linspace(-5,5,100)
y_arr = np.linspace(-5,5,100)
X, Y = np.meshgrid(x_arr, y_arr)
Z = np.sin(np.sqrt(X**2 + Y**2))
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='k')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title(r'$z = \sin(\sqrt{x^2 + y^2})$')
plt.show()
```

Рисунок 12 – Код к заданию 7

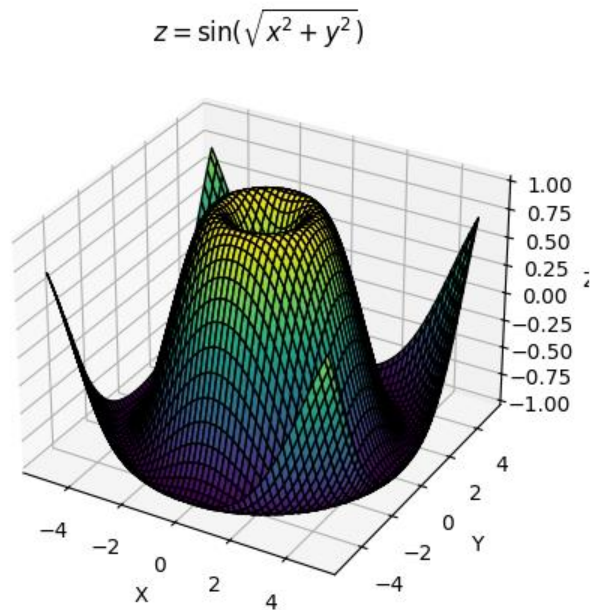


Рисунок 13 – Графика к заданию 7

8. Множественные подграфики(subplots).

```
[7]: x_arr = np.linspace(-5,5,100)
     y1_arr = x_arr
     y2_arr = x_arr**2
     y3_arr = np.sin(x_arr)
     y4_arr = np.cos(x_arr)

     fig, axes = plt.subplots(2, 2, figsize=(10, 8))
     axes[0, 0].plot(x_arr, y1_arr, label='y = x', color='b')
     axes[0, 0].set_title('Линейный график')
     axes[0, 0].set_xlabel('x')
     axes[0, 0].set_ylabel('y')
     axes[0, 0].grid(True)

     axes[0, 1].plot(x_arr, y2_arr, label='y = x^2', color='r')
     axes[0, 1].set_title('Парабола')
     axes[0, 1].set_xlabel('x')
     axes[0, 1].set_ylabel('y')
     axes[0, 1].grid(True)

     axes[1, 0].plot(x_arr, y3_arr, label='y = sin(x)', color='g')
     axes[1, 0].set_title('Синус')
     axes[1, 0].set_xlabel('x')
     axes[1, 0].set_ylabel('y')
     axes[1, 0].grid(True)

     axes[1, 1].plot(x_arr, y4_arr, label='y = cos(x)', color='m')
     axes[1, 1].set_title('Косинус')
     axes[1, 1].set_xlabel('x')
     axes[1, 1].set_ylabel('y')
     axes[1, 1].grid(True)

     fig.suptitle('Четыре графика в одной фигуре', fontsize=14)
```

Рисунок 14 – Код к заданию 8

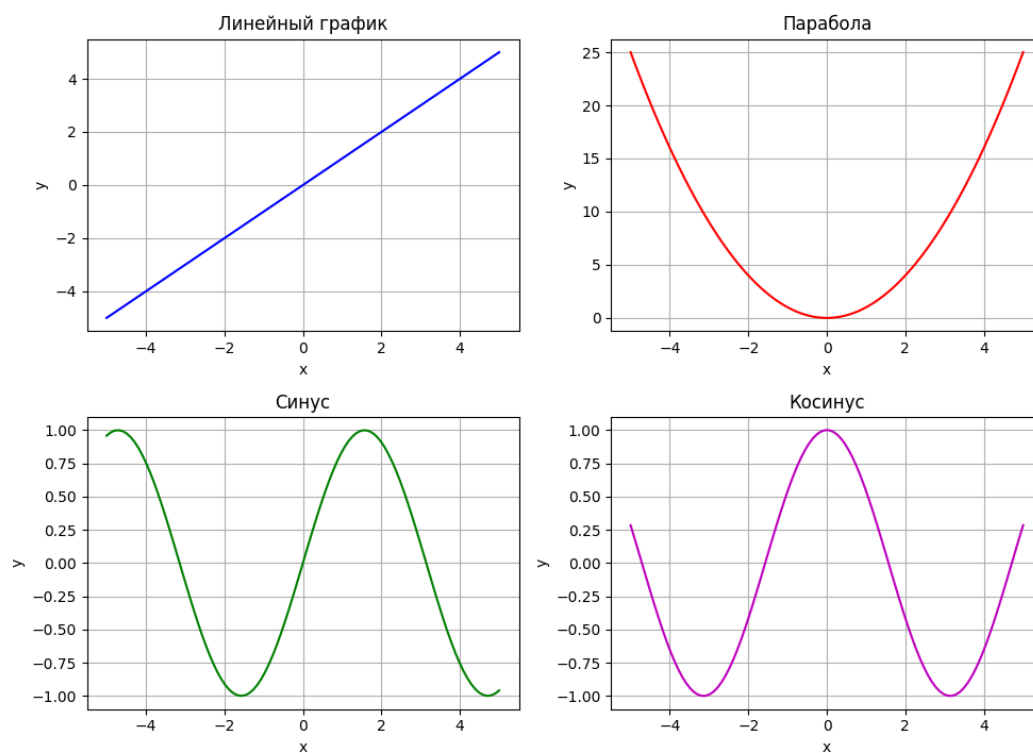


Рисунок 15 – Графика к заданию 8

9. Тепловая карта(imshow).

```
|: data = np.random.rand(10, 10)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
plt.colorbar()
plt.show()
```

Рисунок 16 – Код к заданию 9

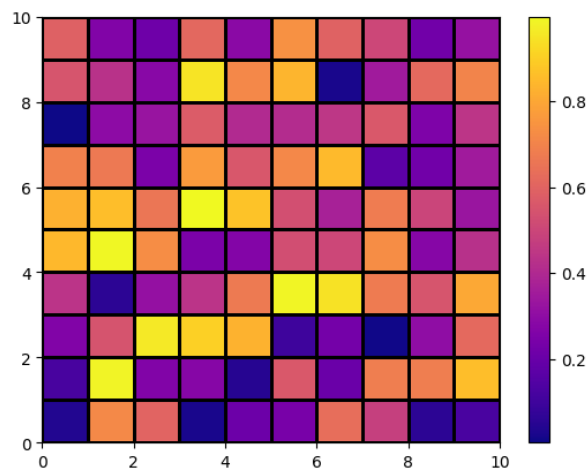


Рисунок 17 – Графика к заданию 9

10. Выполнение индивидуального задания.

```
[5]: exp_arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
salary_arr = np.array([40, 42, 45, 50, 55, 60, 65, 70, 75, 80])
plt.figure(figsize=(5,5))
plt.plot(exp_arr,salary_arr,linewidth=1)
plt.title('График функции зарплат от стажа')
plt.xlabel('Стаж')
plt.ylabel('Зарплата')
plt.grid(True)
plt.show()
```

Рисунок 18 – Код к индивидуальному заданию

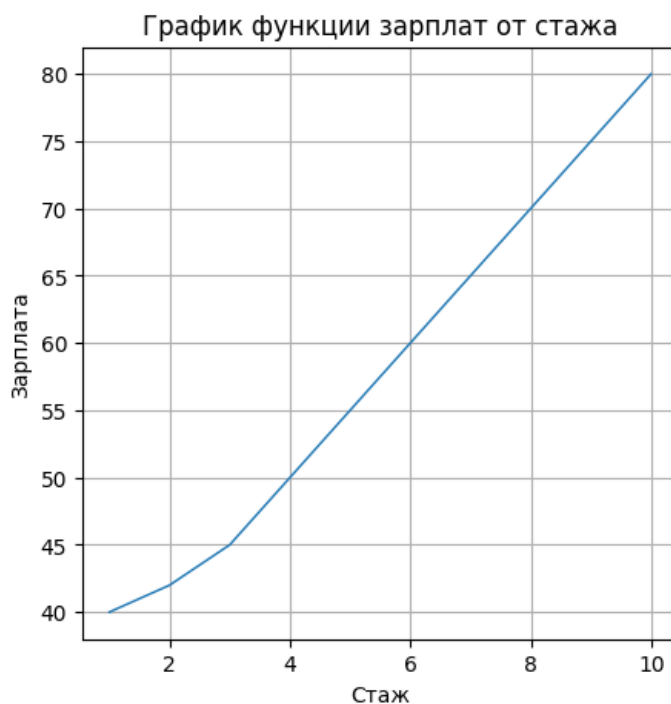


Рисунок 19 – Графика к индивидуальному заданию

11. Задачи на построение столбчатых диаграмм.

```
[5]: mark_arr = ['Toyota', 'BMW', 'Mercedes', 'Ford', 'Honda']
cars_arr = np.array([25, 15, 10, 20, 18])
plt.figure(figsize=(5,5))
plt.bar(mark_arr,cars_arr)
plt.title('Количество автомобилей разных марок')
plt.xlabel('Марка авто')
plt.ylabel('Количество автомобилей')
plt.grid(True)
for i, value in enumerate(cars_arr):
    plt.text(mark_arr[i], value +0.5, str(value), ha='center', va='bottom')
plt.show()
```

Рисунок 20 – Код для столбчатых диаграмм

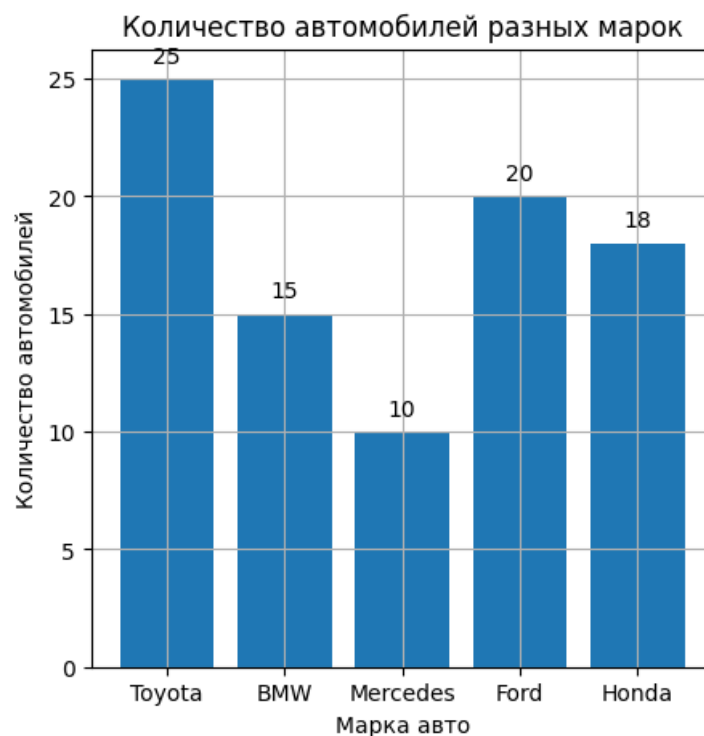


Рисунок 21 – Столбчатые диаграммы

12. Задачи на вычисление определенного интеграла с помощью Matplotlib.

```

: x_arr = np.linspace(1,4,50)
  fx_arr = np.log(x_arr)
  plt.figure(figsize=(5,5))
  plt.plot(x_arr, fx_arr)
  plt.title('График логарифмической кривой')
  plt.fill_between(x_arr, fx_arr, color='skyblue', alpha=0.4)
  plt.xlabel('X')
  plt.ylabel('Y')
  plt.grid(True)

def f(x):
    return np.log(x)

def trapezoidal_rule(f, a, b, n):
    x = np.linspace(a, b, n+1)
    y = f(x)
    h = (b - a) / n
    integral = (h / 2) * (y[0] + 2 * np.sum(y[1:n]) + y[n])
    return integral

result = trapezoidal_rule(f, 1, 4, 50)
print(f"Приближенное значение определенного интеграла: {result}")

```

Рисунок 22 – Код для вычисления определенного интеграла

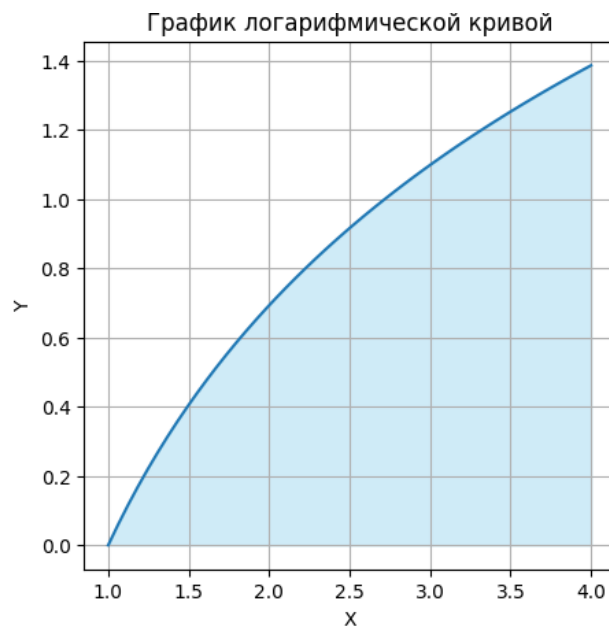


Рисунок 23 – График логарифмической кривой

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

```
: x_arr = np.linspace(-np.pi,np.pi,50)
  y_arr = np.linspace(-np.pi,np.pi,50)
  func_arr = np.sin(x_arr)*np.cos(y_arr)
  X, Y = np.meshgrid(x_arr, y_arr)
  Z = np.sin(X) * np.cos(Y)
  fig = plt.figure(figsize=(8, 7))
  ax = fig.add_subplot(111, projection='3d')
  ax.plot_surface(X, Y, Z, cmap='plasma')
  ax.set_xlabel('X')
  ax.set_ylabel('Y')
  ax.set_zlabel('f(x, y)')
  ax.set_title('График функции f(x, y) = sin(x) * cos(y)')
  plt.show()
```

Рисунок 24 – Код для построения 3D-графика

График функции $f(x, y) = \sin(x) * \cos(y)$

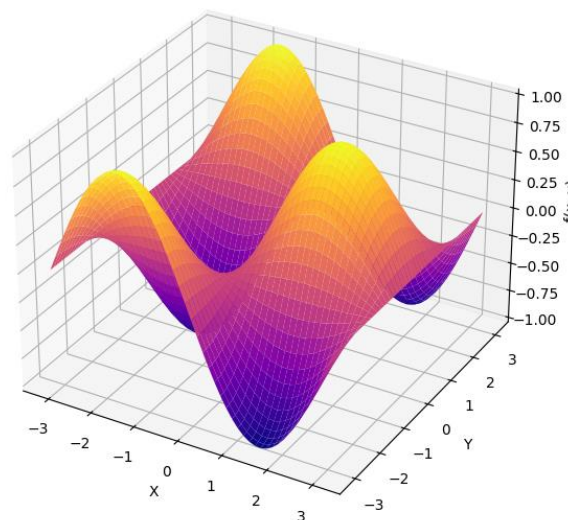


Рисунок 25 – Графика для задания

Контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Установка пакета matplotlib осуществляется с помощью команды `pip install matplotlib` в командной строке или терминале.

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

Для корректного отображения графиков в ноутбуках Jupyter используется магическая команда `%matplotlib inline`.

3. Как отобразить график с помощью функции plot ?

Для отображения графика с помощью функции plot нужно использовать команду `plt.plot(x, y)` и затем `plt.show()`.

4. Как отобразить несколько графиков на одном поле?

Чтобы отобразить несколько графиков на одном поле, можно использовать функцию `plt.subplot(rows, cols, index)` перед каждым графиком.

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

Для построения диаграмм категориальных данных используется метод `plt.bar()` для столбчатых диаграмм или `plt.boxplot()` для коробчатых диаграмм.

6. Какие основные элементы графика Вам известны?

Основные элементы графика включают оси (x, y), метки осей, легенду, заголовок, сетку и данные.

7. Как осуществляется управление текстовыми надписями на графике?

Для управления текстовыми надписями на графике используется метод `plt.text(x, y, 'text')` для добавления текста в указанные координаты.

8. Как осуществляется управление легендой графика?

Для управления легендой графика используется метод `plt.legend()`.

9. Как задать цвет и стиль линий графика?

Цвет и стиль линий графика задаются с помощью параметров, таких как `color`, `linestyle` в функции `plot()`, например, `plt.plot(x, y, color='r', linestyle='--')`.

10. Как выполнить размещение графика в разных полях?

Для размещения графиков в разных полях используется метод `plt.subplot()`.

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

Линейный график строится с помощью функции `plt.plot(x, y)`.

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

Заливка области между графиком и осью или между двумя графиками выполняется с помощью метода `plt.fill_between(x, y1, y2)`.

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

Для выборочной заливки, которая удовлетворяет определенному условию, используется метод `plt.fill_between(x, y, condition)`.

14. Как выполнить двухцветную заливку?

Двухцветную заливку можно сделать с помощью `plt.fill_between(x, y1, y2, where=condition, color='color1', alpha=0.5)`.

15. Как выполнить маркировку графиков?

Для маркировки графиков используются функции `plt.text()` или `plt.annotate()` для добавления меток на график.

16. Как выполнить обрезку графиков?

Обрезка графиков осуществляется с помощью метода `plt.xlim()` и `plt.ylim()` для ограничения диапазонов осей.

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

Ступенчатый график строится с помощью функции `plt.step(x, y)`, и его особенность в том, что линии соединяют данные ступенями.

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

Стековый график строится с помощью функции `plt.stackplot(x, y)`, и его особенность в том, что области под графиком накладываются друг на друга.

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

Стем-график строится с помощью `plt.stem(x, y)`, его особенность в том, что он отображает данные в виде вертикальных линий с маркерами на вершинах.

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

Точечный график строится с помощью функции `plt.scatter(x, y)`, и его особенность в том, что отображает данные в виде точек.

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

Столбчатая диаграмма строится с помощью `plt.bar(x, height)`.

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

Групповая столбчатая диаграмма строится с использованием нескольких `plt.bar()` с раздвигом на оси `x`. Столбчатая диаграмма с элементом `errorbar` добавляется через `plt.errorbar()`.

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

Круговую диаграмму можно построить с помощью метода `plt.pie(data)`

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

Цветовая карта — это способ отображения данных через цвета. В `matplotlib` она используется с помощью функций типа `plt.imshow()` или `plt.contourf()`, где можно задавать различные схемы цветов.

25. Как отобразить изображение средствами `matplotlib`?

Изображение отображается с помощью `plt.imshow(image)`.

26. Как отобразить тепловую карту средствами `matplotlib`?

Тепловая карта отображается с помощью `plt.imshow(data, cmap='hot')`

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

Для построения линейного 3D-графика используется `ax.plot(x, y, z)` с осью 3D.

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

Для построения точечного 3D-графика используется `ax.scatter(x, y, z)` с осью 3D.

29. Как выполнить построение каркасной поверхности с помощью `matplotlib`?

Каркасную поверхность можно построить с помощью `ax.plot_wireframe(X, Y, Z)`.

30. Как выполнить построение трехмерной поверхности с помощью `matplotlib`?

Трехмерную поверхность можно построить с помощью `ax.plot_surface(X, Y, Z)` для 3D-графиков.

Вывод: В ходе практической работы, исследовал базовые возможности библиотеки `matplotlib` языка программирования Python.