

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Визуализация данных с помощью matplotlib»

ОТЧЁТ
по лабораторной работе №5
дисциплины
<<Технологии Распознавания образов>>

Выполнил:
Зиёдуллаев Жавохир
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Проработка Примеров:

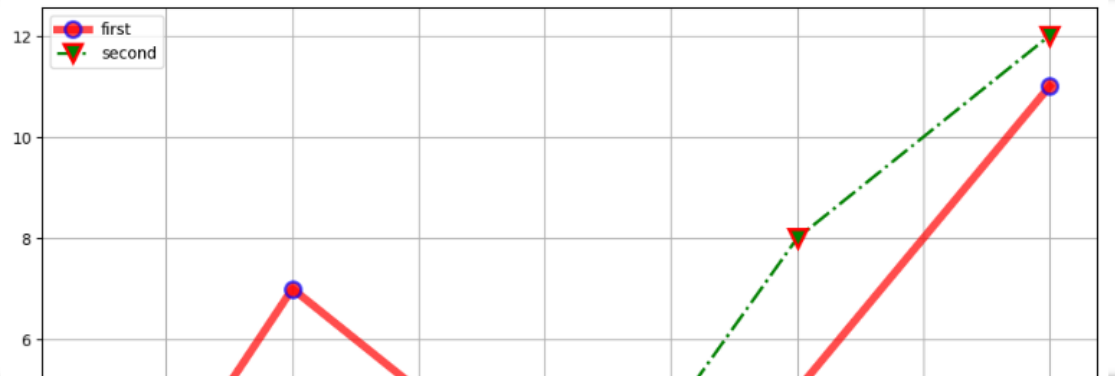
```

Ввод [3]: x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [4, 3, 1, 8, 12]

plt.figure(figsize=(12, 7))
plt.plot(x, y1, 'o-r', alpha=0.7, label="first", lw=5, mec='b', mew=2, ms=10)
plt.plot(x, y2, 'v-.g', label="second", mec='r', lw=2, mew=2, ms=12)

plt.legend()
plt.grid(True)

```



Заливки области между графиком и осью

```

Ввод [6]: import numpy as np

x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)

```

Рисунок -1 Проработка примеров

Заливки области между графиком и осью

```

Ввод [6]: import numpy as np

x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)

```

```

Ввод [7]: plt.plot(x, y, c = "r")
plt.fill_between(x, y)

```

Out[7]: <matplotlib.collections.PolyCollection at 0x1a3cef33340>

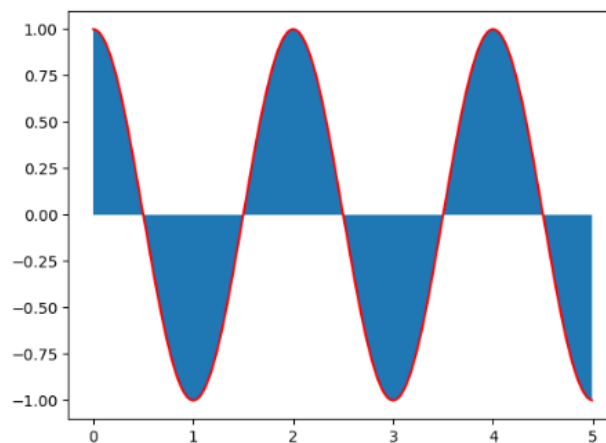
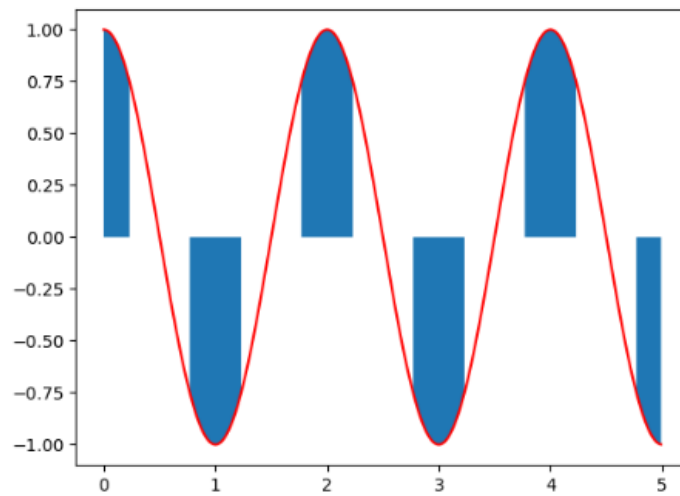


Рисунок -2 Проработка примеров

```
Ввод [8]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))

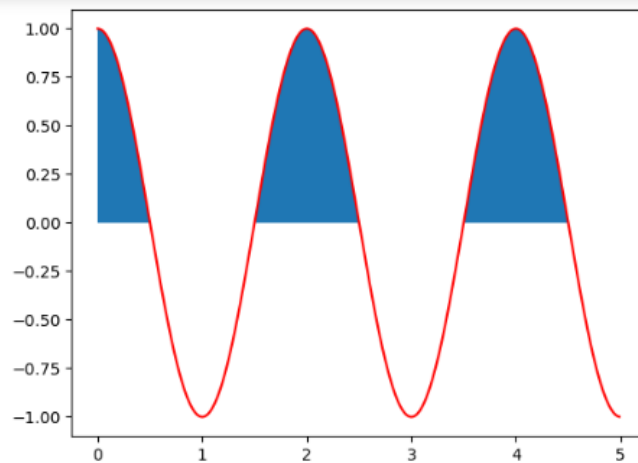
Out[8]: <matplotlib.collections.PolyCollection at 0x1a3cecc8e0>
```



```
Ввод [9]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0))

Out[9]: <matplotlib.collections.PolyCollection at 0x1a3cef949a0>
```

Рисунок -3 Проработка примеров



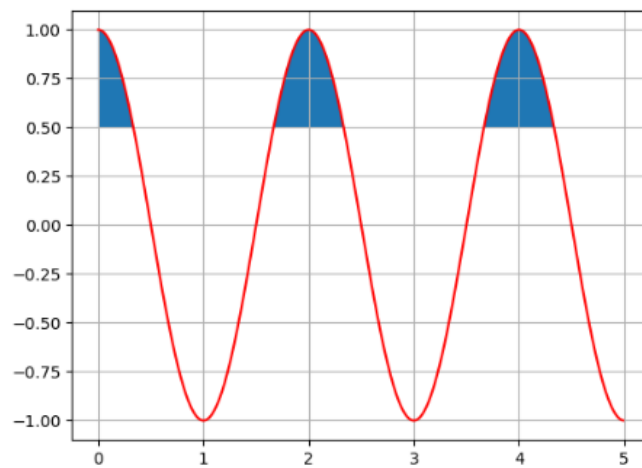
```
Ввод [10]: plt.plot(x, y, c="r")
plt.grid()

plt.fill_between(x, 0.5, y, where=(y >= 0.5))

Out[10]: <matplotlib.collections.PolyCollection at 0x1a3cf080fd0>
```

Рисунок -4 Проработка примеров

Out[10]: <matplotlib.collections.PolyCollection at 0x1a3cf080fd0>

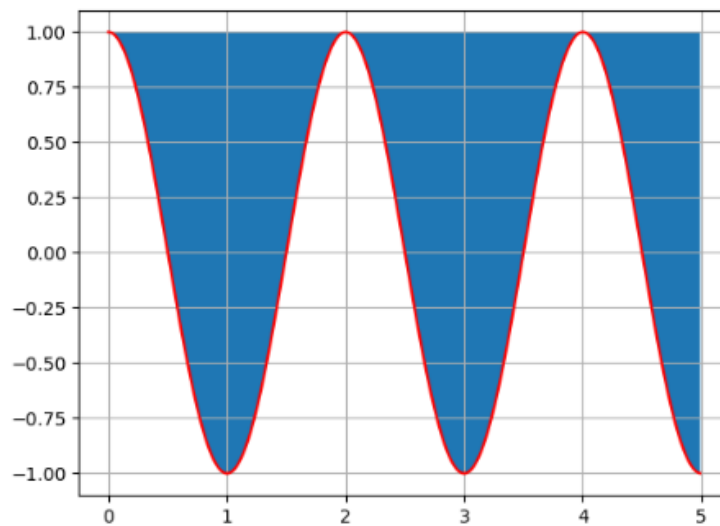


```
Ввод [11]: plt.plot(x, y, c="r")
plt.grid()

plt.fill_between(x, y, 1)
```

Рисунок -5 Проработка примеров

Out[11]: <matplotlib.collections.PolyCollection at 0x1a3cf0f4fa0>

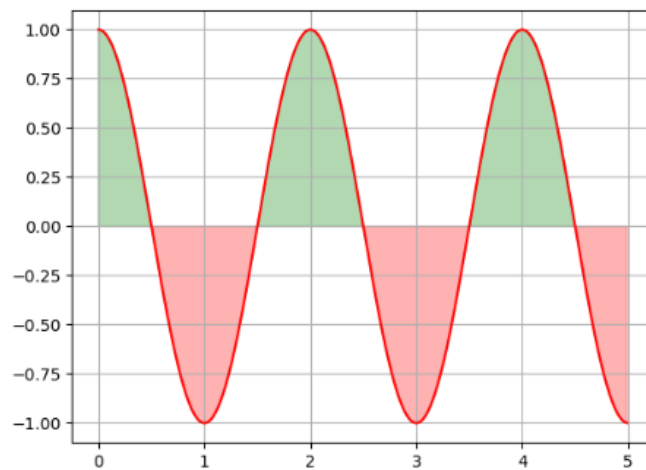


```
Ввод [12]: plt.plot(x, y, c="r")
plt.grid()

plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)
plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)
```

Рисунок -6 Проработка примеров

```
Out[12]: <matplotlib.collections.PolyCollection at 0x1a3cf1766a0>
```

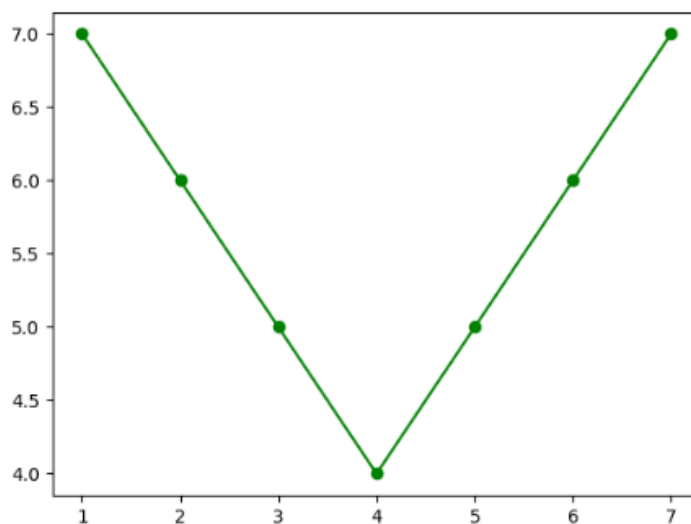


Настройка маркировки графиков

```
Ввод [13]: x = [1, 2, 3, 4, 5, 6, 7]
           y = [7, 6, 5, 4, 5, 6, 7]
           plt.plot(x, y, marker="o", c="g")
```

Рисунок -7 Приработка примеров

```
Out[13]: [<matplotlib.lines.Line2D at 0x1a3d033f0a0>]
```

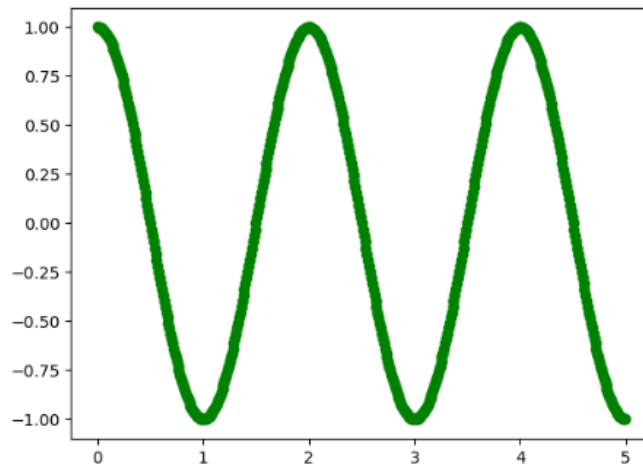


```
Ввод [14]: import numpy as np
           x = np.arange(0.0, 5, 0.01)
           y = np.cos(x*np.pi)
```

```
Ввод [15]: plt.plot(x, y, marker="o", c="g")
```

Рисунок -8 Проработка примеров

```
Out[15]: [matplotlib.lines.Line2D at 0x1a3cf250610]
```



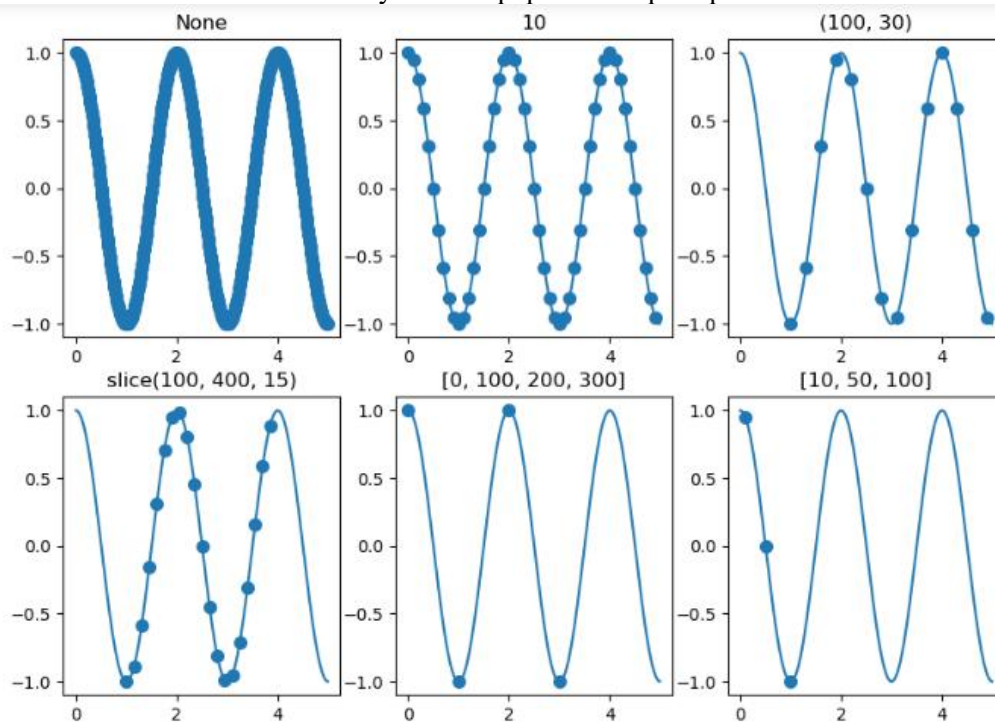
```
Ввод [16]: x = np.arange(0.0, 5, 0.01)

y = np.cos(x * np.pi)
m_ev_case = [None, 10, (100, 30), slice(100, 400, 15), [0, 100, 200, 300], [10, 50, 100]]

fig, ax = plt.subplots(2, 3, figsize=(10, 7))
axs = [ax[i, j] for i in range(2) for j in range(3)]

for i, case in enumerate(m_ev_case):
    axs[i].set_title(str(case))
    axs[i].plot(x, y, "o", ls='-', ms=7, markevery=case)
```

Рисунок -9 Проработка примеров



Обрезка графика

```
Out[16]: [matplotlib.figure.Figure at 0x1a3cf250610]
```

Рисунок -10 Проработка примеров

Обрезка графика

```
Ввод [17]: x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)

y_masked = np.ma.masked_where(y < -0.5, y)
plt.ylim(-1, 1)

plt.plot(x, y_masked, linewidth=3)

Out[17]: [<matplotlib.lines.Line2D at 0x1a3d044d790>]
```

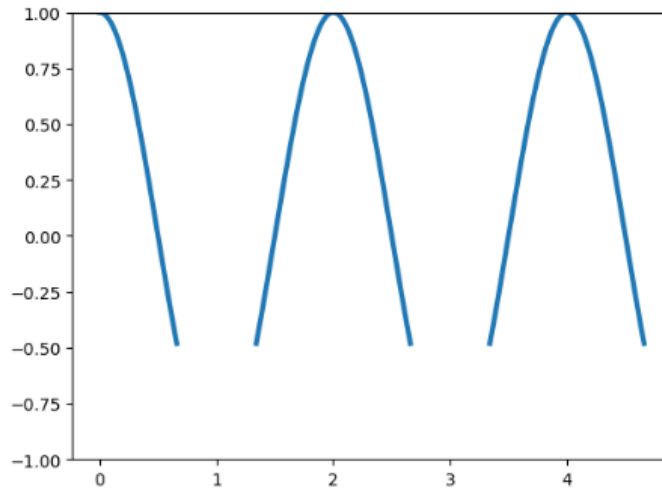


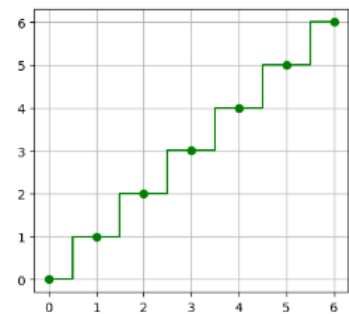
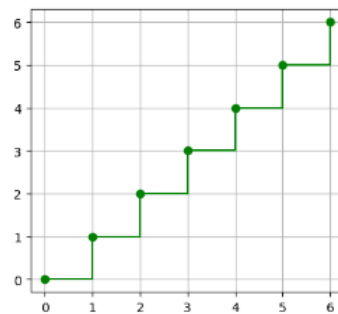
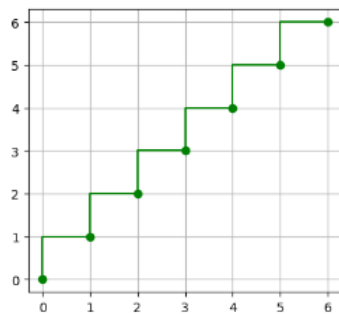
Рисунок -11 Проработка примеров

Ступенчатый график

```
Ввод [18]: x = np.arange(0, 7)
y = x

where_set = ['pre', 'post', 'mid']
fig, axs = plt.subplots(1, 3, figsize=(15, 4))

for i, ax in enumerate(axs):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()
```



Стековый график

```
Ввод [19]: x = np.arange(0, 11, 1)

y1 = np.array([(-0.2)*i**2+2*i for i in x])
y2 = np.array([(-0.4)*i**2+4*i for i in x])
y3 = np.array([2*i for i in x])
```

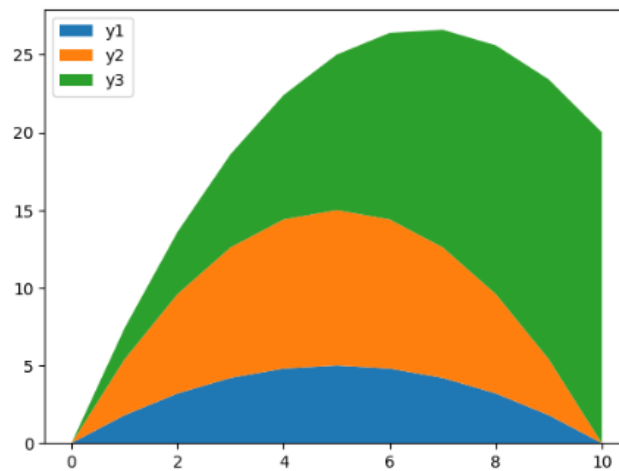
Рисунок -12 Проработка примеров

```

labels = ["y1", "y2", "y3"]
fig, ax = plt.subplots()
ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')

```

Out[19]: <matplotlib.legend.Legend at 0x1a3d0557b20>



Stem-график

```

Ввод [20]: x = np.arange(0, 10.5, 0.5)
           y = np.array([(-0.2)*i**2+2*i for i in x])

```

Рисунок -13 Проработка примеров

Цветовые карты (colormaps)

#Отображение изображение

```

Ввод [39]: from PIL import Image
           import requests

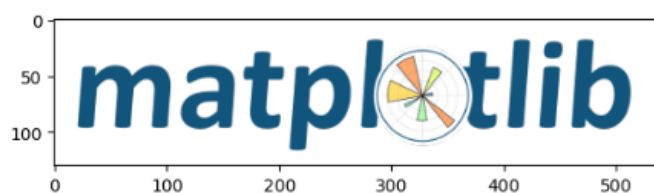
           from io import BytesIO

           response = requests.get('https://matplotlib.org/_static/logo2.png')
           img = Image.open(BytesIO(response.content))

           plt.imshow(img)

```

Out[39]: <matplotlib.image.AxesImage at 0x1a3d2216040>



```

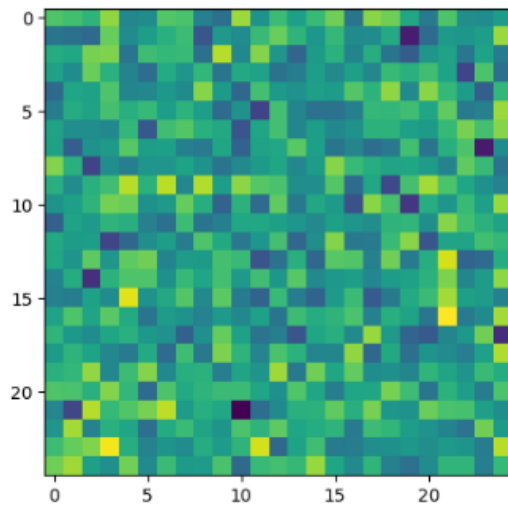
Ввод [40]: np.random.seed(19680801)

           data = np.random.randn(25, 25)
           plt.imshow(data)

```

Рисунок -14 Проработка примеров

Out[40]: <matplotlib.image.AxesImage at 0x1a3d22726d0>



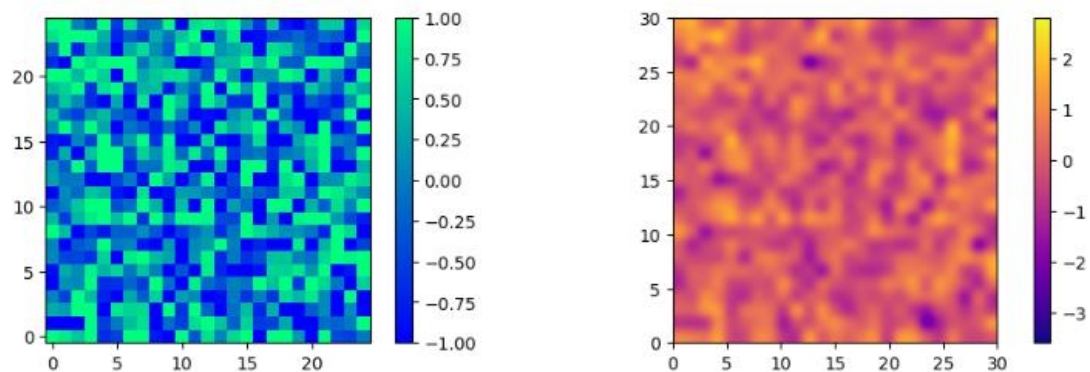
```
Ввод [41]: fig, axs = plt.subplots(1, 2, figsize=(10,3), constrained_layout=True)

p1 = axs[0].imshow(data, cmap='winter', aspect='equal', vmin=-1, vmax=1,
origin="lower")
fig.colorbar(p1, ax=axs[0])

p2 = axs[1].imshow(data, cmap='plasma', aspect='equal',
interpolation='gaussian', origin="lower", extent=(0, 30, 0, 30))
fig.colorbar(p2, ax=axs[1])
```

Рисунок -15 Проработка примеров

Out[41]: <matplotlib.colorbar.Colorbar at 0x1a3d23fac10>



```
Ввод [43]: np.random.seed(123)

data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
```

Out[43]: <matplotlib.collections.QuadMesh at 0x1a3d2478730>

Рисунок -16 Проработка примеров

Задание №1: Демонстрация работы с линейным графиком

Задание №1

Решение физической задачи с использованием линейного графика

Условие: Автомобильная камера накачана до давления $p_1 = 220\text{кПа}$ при температуре $T_1 = 290\text{К}$. Во время движения она нагрелась до температуры $T_2 = 330\text{К}$ и с шумом лопнула. Считая процесс, происходящий после повреждения камеры, адиабатным, определить изменение температуры вышедшего из неё воздуха. Внешнее давление $p_0 = 100\text{кПа}$

Аналитическое решение: Выделим два термодинамических процесса задачи. Первый - изохорическое нагревание воздуха в камере, а второй адиабатическое расширение воздуха:

1. $p_1/T_1 = p_2/T_2$, $p_2 = p_1 T_2/T_1$
2. $T_2 p_2^{(1-\gamma)/\gamma} = T_0 p_0^{(1-\gamma)/\gamma}$, где T_0 - температура воздуха в конце адиабатического расширения,
 $T_0 = T_2 p_2^{(1-\gamma)/\gamma} / p_0^{(1-\gamma)/\gamma}$.
3. Теперь найдём разность температур $\Delta T = T_0 - T_2$, $\Delta T = T_2 (p_2/p_0)^{(1-\gamma)/\gamma} - T_2$ и
 $\Delta T = T_2 \times ((p_2/p_0)^{(1-\gamma)/\gamma} - 1)$

Рисунок 29 – Задача №1

Составим программу для графического решения данной задачи:

```
In [7]: import matplotlib.pyplot as plt
import math as mt
R=8.31
p0=1.0e5
p1=2.2e5
T1=290
T2=330
g=1.4
b=(1-g)/g
c=1/b
p2=p1*T2/T1
T0=T2*mt.pow(p2/p0,b)
DT=T2-T0
print ("p2=%8.3e"%p2, "T0=%5.1f"%T0, "DT=%5.1f"%DT)
N1=100; dT1=(T2-T1)/N1; dT2=(T0-T2)/N1;
t=[]
DN=[]
t.append(T1)
DN.append(p1)
for i in range(1,N1):
    t1=T1+i*dT1; t.append(t1); DN.append(p1*t1/T1)
for i in range(1,N1):
    t1=T2+i*dT2
    t.append(t1); DN.append(p2*mt.pow(T2/t1,c))
plt.plot(t,DN,'k-')
plt.xlabel('$T$', fontsize=14)
plt.ylabel('$p$', fontsize=14)
plt.show()
```

p2=2.503e+05 T0=253.9 DT= 76.1

Рисунок 30 – Задача №1

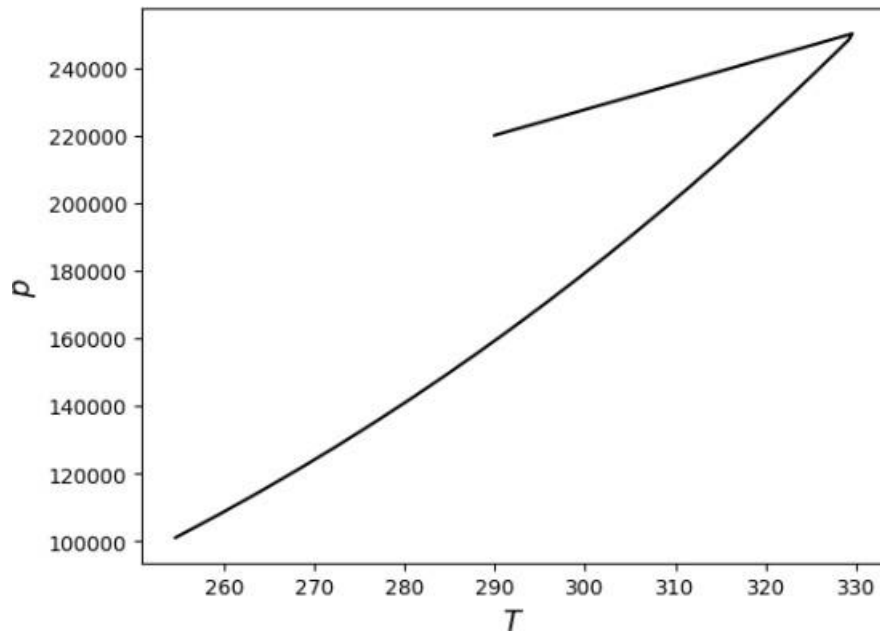


Рисунок 31 – Задача №1

Задание №2: Демонстрация работы с гистограммой

Условие задачи: Найти давление воздуха в откачиваемом сосуде как функцию времени откачки t . Объем сосуда $V = 100$ л. Процесс считать изотермическим и скорость откачки независимой от давления и равной $C = 0.01$ л/с. Скоростью откачки называют объем газа, откачиваемый за единицу времени, причем этот объем измеряется при давлении газа в данный момент времени.

Аналитическое решение поставленной задачи: Рассмотрим изотермический процесс при изменении объема на dV и давления на dp , тогда $pV = (p + dp)(V + dV)$, или $pV = pV + pdV + Vdp + dpdV$; пренебрегая последним слагаемым, мы получаем дифференциальное уравнение первого порядка с разделяющимися переменными $dp/p = -dV/V$. Учитывая, что изменение объема $dV = Cdt$, уравнение можно проинтегрировать:

$\ln(p) - \ln(p_0) = -\frac{C}{V}t$ или $\frac{p}{p_0} = \exp(-\frac{C}{V}t)$ Полученная формула является решением задачи

Программа для решения поставленной задачи:

```
In [6]: import matplotlib.pyplot as plt
import math
N=20; C=0.01; tmax=50.0; dt=tmax/N ; V=0.1;
dV=0.0005;qV=V/(V+dV); b1=C/V;dt1=V*(1.0/qV-1.0)/C
p0=1.0
print (" q =",qV," b1=",b1," \Delta t=",dt1)
t=[]; p1=[]; p2=[]
t.append(0); p1.append(p0); p2.append(p0)
for i in range(1,N):
    t1=i*dt; t.append(t1)
    p1.append(math.exp(-b1*t1))
    p2.append(qV**(t1/dt1))
plt.bar(t,p1)
plt.xlabel('$t$',fontsize=14)
plt.ylabel('$\eta_1, \eta_2$',fontsize=14)
plt.show()

q = 0.9950248756218906 b1= 0.09999999999999999 \Delta t= 0.0499999999999998934
```

Рисунок 32 – Задача №2

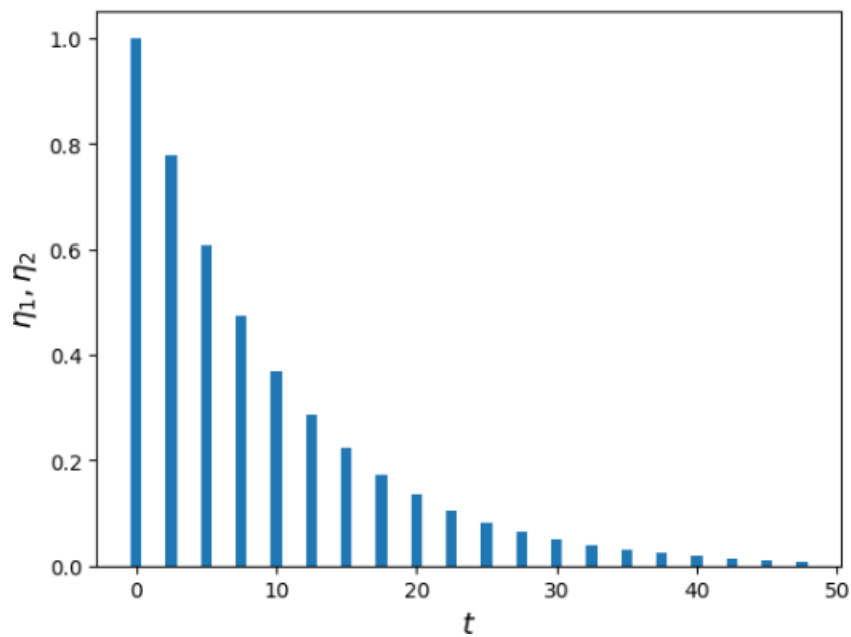


Рисунок 33 – Задача №2

Задание №3: Демонстрация работы с круговой диаграммой

Соотношение людей женского и мужского пола которые встречались с буллингом.

In [2]: `import matplotlib.pyplot as plt`
`%matplotlib inline`

In [3]: `import csv`

```
with open('bullying.csv', 'r', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=',')
    values=[]
    male = 0
    female = 0
    for rowdata in data:
        if rowdata[5] == "Female" and (rowdata[1] or rowdata[2] or rowdata[3]) == "Yes":
            female +=1
        elif rowdata[5] == "Male" and (rowdata[1] or rowdata[2] or rowdata[3]) == "Yes":
            male +=1
    values.append(male)
    values.append(female)
    plt.figure(figsize=(30, 30))
    explode = (0.1, 0.15)
    labels = ["Женщины", "Мужчины"]
    fig, ax = plt.subplots()
    ax.pie(values, labels=labels, shadow=True, explode=explode)
    plt.show()
```

<Figure size 3000x3000 with 0 Axes>



Рисунок 34 – Задача №3

Задание №4: Работа с изображениями

```
Ввод [1]: import matplotlib.pyplot as plt
%matplotlib inline
from PIL import Image
import requests
from io import BytesIO
plt.figure(figsize=(10, 10))
response_cat1 = requests.get('https://i.yapx.cc/MvQ5k.gif')
img_cat1 = Image.open(BytesIO(response_cat1.content))
plt.imshow(img_cat1)

Out[1]: <matplotlib.image.AxesImage at 0x1575e807e80>
```



Рисунок 35 – Задача №4

Контрольные вопросы

1. Как выполнить построение линейного графика с помощью matplotlib?

Для построения линейного графика используется функция `plot()`, со следующей сигнатурой:

```
plot([x], y, [fmt], *, data=None, **kwargs)
plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)
```

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

Для заливки областей используется функция `fill_between()`. Сигнатура функции:

```
fill_between(x, y1, y2=0, where=None, interpolate=False, step=None, *,
data=None, **kwargs)
```

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

```
plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0))
```

4. Как выполнить двухцветную заливку?

Вариант двухцветной заливки:

```
plt.plot(x, y, c="r")  
plt.grid()  
  
plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)  
plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)
```

5. Как выполнить маркировку графиков?

```
plt.plot(x, y, marker="o", c="g")
```

6. Как выполнить обрезку графиков?

Для того, чтобы отобразить только часть графика, которая отвечает определенному условию используйте предварительное маскирование данных с помощью функции *masked_where* из пакета *numpy*.

```
x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)

y_masked = np.ma.masked_where(y < -0.5, y)
plt.ylim(-1, 1)

plt.plot(x, y_masked, linewidth=3)
```

7. Как построить ступенчатый график?

```
x = np.arange(0, 7)
y = x

where_set = ['pre', 'post', 'mid']
fig, axs = plt.subplots(1, 3, figsize=(15, 4))

for i, ax in enumerate(axs):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()
```

8. Как построить стековый график?

```
x = np.arange(0, 11, 1)

y1 = np.array([(-0.2)*i**2+2*i for i in x])
y2 = np.array([(-0.4)*i**2+4*i for i in x])
y3 = np.array([2*i for i in x])

labels = ["y1", "y2", "y3"]

fig, ax = plt.subplots()

ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')
```

9. Как построить stem-график?

```
x = np.arange(0, 10.5, 0.5)
y = np.array([(-0.2)*i**2+2*i for i in x])

plt.stem(x, y)
```


10. Как построить точечный график?

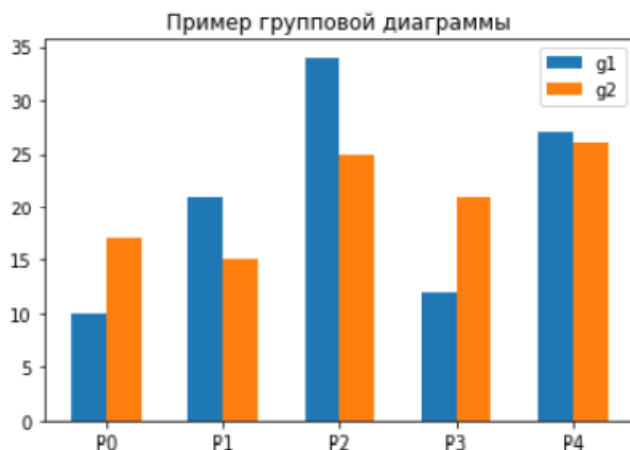
```
x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.scatter(x, y)
```

11. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

bar() – для построения вертикальной диаграммы
barh() – для построения горизонтальной диаграммы.

12. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с errorbar элементом?



Errorbar элемент позволяет задать величину ошибки для каждого элемента графика. Для этого используются параметры *xerr*, *yerr* и *ecolor* (для задания цвета):

13. Как выполнить построение круговой диаграммы средствами matplotlib?

Круговые диаграммы – это наглядный способ показать доли компонент в наборе. Они идеально подходят для отчетов, презентаций и т.п. Для построения круговых диаграмм в *Matplotlib* используется функция *pie()*.

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

Цветовая карта представляет собой подготовленный набор цветов, который хорошо подходит для визуализации того или иного набора данных. Подробное руководство по цветовым картам вы можете найти на официальном сайте *Matplotlib* (<https://matplotlib.org/tutorials/colors/colormaps.html#sphx-glr-tutorials-colors-colormaps-py>). Также отметим, что такие карты можно создавать самостоятельно, если среди существующих нет подходящего решения. Ниже представлены примеры некоторых цветовых схем, из библиотеки *Matplotlib*.

15. Как отобразить изображение средствами matplotlib?

```
from PIL import Image
import requests

from io import BytesIO

response = requests.get('https://matplotlib.org/_static/logo2.png')
img = Image.open(BytesIO(response.content))

plt.imshow(img)
```

16. Как отобразить тепловую карту средствами matplotlib?

```
np.random.seed(123)

data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
```