

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Северо-Кавказский федеральный университет»**

**Кафедра инфокоммуникаций**

**Отчёт по практическому занятию №3.5  
«Визуализация данных с помощью matplotlib»**

**по дисциплине «Теории распознавания образов»**

Выполнил студент группы ИВТ-б-о-21-1  
Хашиев Адам М.  
Проверил Воронкин Р.А.

(подпись)

Ставрополь 2023

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).

2. Проработать примеры лабораторной работы.



Создать ноутбук, в котором выполнить решение трех вычислительных задач (например, задачи из области физики, экономики, математики, статистики и т. д.) требующих построения графика (линейного, кругового, столбчатой), условия которых предварительно необходимо согласовать с преподавателем.

## Вычисляем время падения для сферических частиц

```
time_sphere = np.linspace(0, 2, 100) # Время от 0 до 2 секунды с равными интервалами
height_sphere = initial_velocity_sphere * time_sphere - 0.5 * 9.8 * time_sphere**2
```

[11]

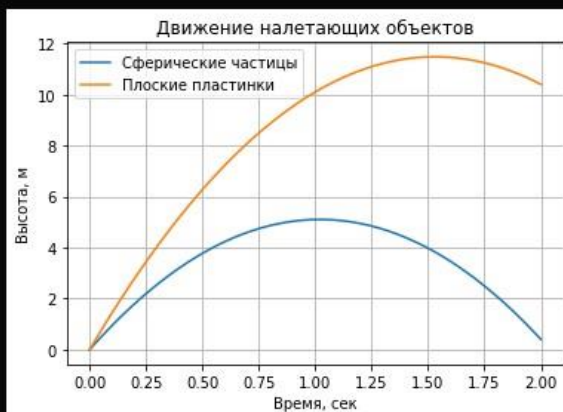
## Вычисляем время падения для плоских пластинок

```
time_plate = np.linspace(0, 2, 100) # Время от 0 до 2 секунды с равными интервалами
height_plate = initial_velocity_plate * time_plate - 0.5 * 9.8 * time_plate**2
```

[12]

```
# Строим график
plt.plot(time_sphere, height_sphere, label='Сферические частицы')
plt.plot(time_plate, height_plate, label='Плоские пластинки')
plt.xlabel('Время, сек')
plt.ylabel('Высота, м')
plt.title('Движение налетающих объектов')
plt.legend()
plt.grid(True)
plt.show()
```

[13]



Среднее количество посещений кардио тренировок: 97.5  
Среднее количество посещений силовых тренировок: 125.0  
Среднее количество посещений групповых тренировок: 95.0

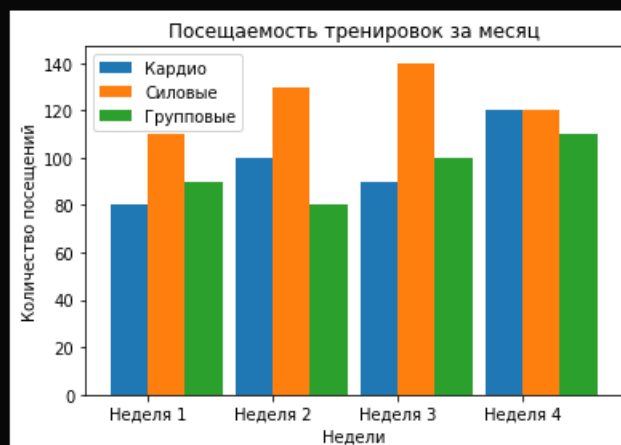
## Выводим график

```
labels = ['Неделя 1', 'Неделя 2', 'Неделя 3', 'Неделя 4']
x = np.arange(len(labels))
width = 0.3

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, cardio_visits, width, label='Кардио')
rects2 = ax.bar(x + width/2, strength_visits, width, label='Силовые')
rects3 = ax.bar(x + width*3/2, group_visits, width, label='Групповые')

ax.set_ylabel('Количество посещений')
ax.set_xlabel('Недели')
ax.set_title('Посещаемость тренировок за месяц')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

plt.show()
```



## Задаём параметры

```
pizza_types = ['Маргарита', 'Пепперони', 'Гавайская']  
orders = [120, 90, 150]  
explode = (0.1, 0.1, 0.15)
```

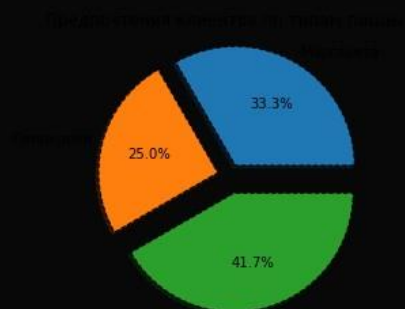
[3]

## Выводим диаграмму

```
fig, ax = plt.subplots()  
  
ax.pie(orders, labels=pizza_types, autopct='%1.1f%%', shadow=True, explode=explode,  
      wedgeprops={'lw': 1, 'ls': '--', 'edgecolor': 'k'})  
  
ax.axis("equal")  
ax.set_title('Предпочтения клиентов по типам пиццы')  
  
plt.show()
```

[4]

...



бо изображение в сети Интернет. Создать ноутбук, в котором будет  
отображено выбранное изображение средствами библиотеки matplotlib по  
URL из сети Интернет.

## Самостоятельное задание

Найти какое-либо изображение в сети Интернет. Создать ноутбук, в котором будет отображено выбранное изображение средствами библиотеки matplotlib URL из сети Интернет.

```
import matplotlib.pyplot as plt

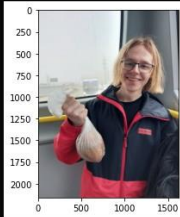
from PIL import Image
import requests

from io import BytesIO

response = requests.get('https://sun9-66.userapi.com/img/FE0Iwn69LNr1X3H51A-e4AMQbgsLC3-78bNDUw/BLqyDgxcNTo.jpg?size=1620x2160&quality=95&sign=486e5b24e00ab0ae6de2c53a114c1247&type=album')
img = Image.open(BytesIO(response.content))

plt.imshow(img)
```

<matplotlib.image.AxesImage at 0x1c0e6b4cc70>



## Вопросы для защиты работы

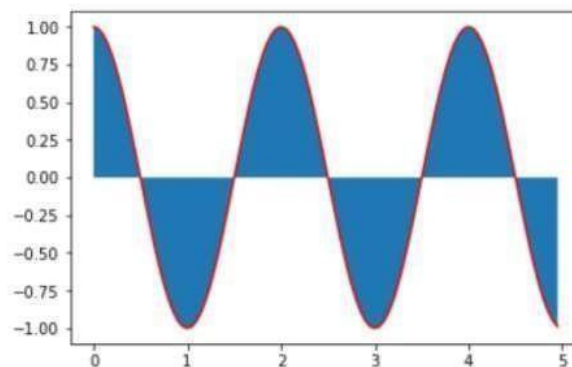
1. Как выполнить построение линейного графика  
спомощью matplotlib?

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)
```

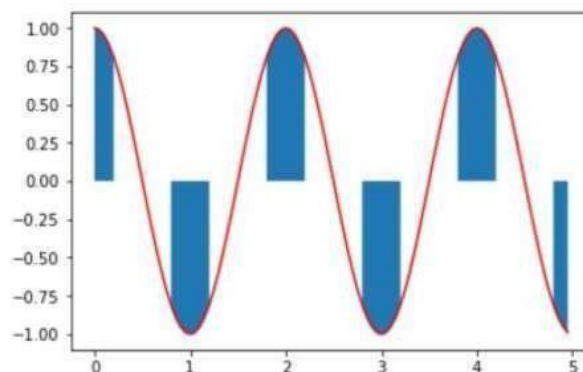
2. Как выполнить заливку области между графиком и  
осью? Между двумя графиками?

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



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

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



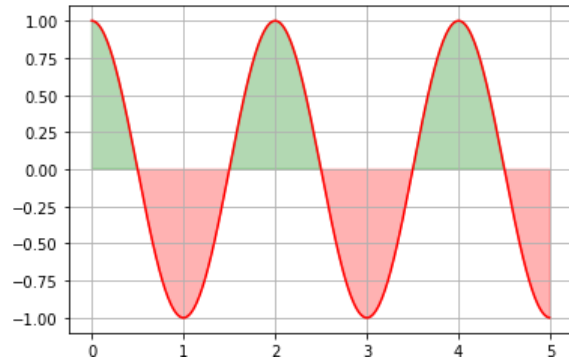
4. Как выполнить двухцветную заливку?



```
In [14]: 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)
```

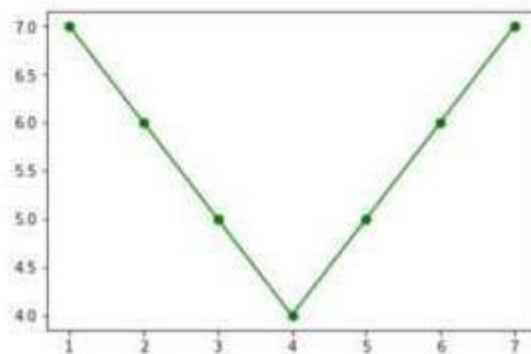
Out[14]: <matplotlib.collections.PolyCollection at 0x1b7583e04f0>



## 5. Как выполнить маркировку графиков?

```
x = [1, 2, 3, 4, 5, 6, 7]
y = [7, 6, 5, 4, 5, 6, 7]

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

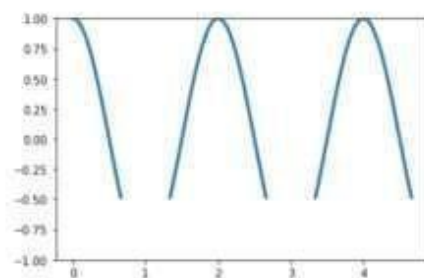


## 6. Как выполнить обрезку графиков?

```
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)
```

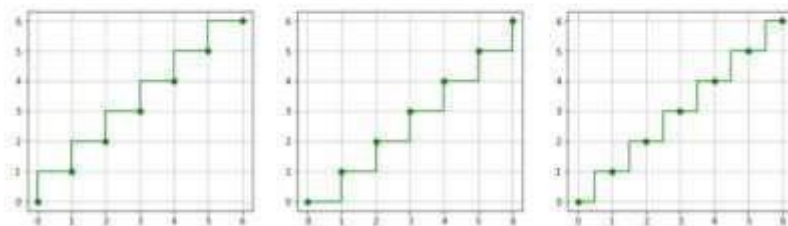


7. Как построить ступенчатый график? В чем особенность ступенчатого графика?

```
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? В чем особенность стекового графика?

Для построения стекового графика используется функция `stackplot()`. Суть его в том, что графики отображаются друг над другом, и каждый следующий является суммой предыдущего и заданного набора данных.

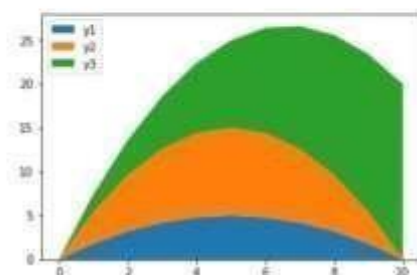
```
x = np.arange(0, 11, 1)

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

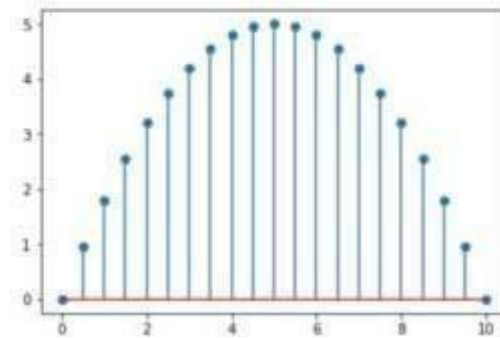


9. Как построить stem-график? В чем особенность stem-графика?

Визуально этот график выглядит как набор линий от точки с координатами (x, y) до базовой линии, в верхней точке ставится маркер.

```
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)
```

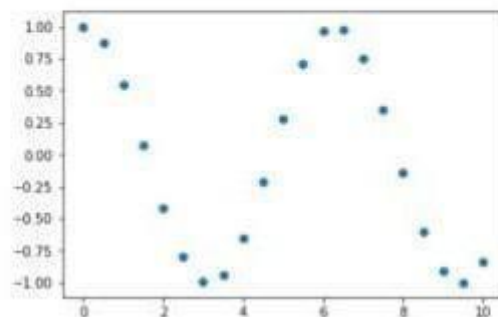


10. Как построить точечный график? В чем особенность точечного графика?

Для отображения точечного графика предназначена функция `scatter()`. В простейшем виде точечный график можно получить передав функции `scatter()` наборы точек для `x`, `y` координат.

```
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`?

```
np.random.seed(123)

groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(1, 10, len(groups))

plt.bar(groups, counts)
```



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

```
cat_par = [f"P{i}" for i in range(5)]

g1 = [10, 21, 34, 12, 27]
g2 = [17, 15, 25, 21, 26]

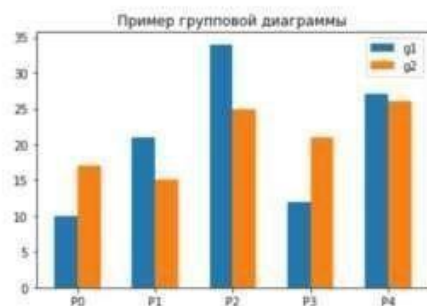
width = 0.3

x = np.arange(len(cat_par))

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, g1, width, label='g1')
rects2 = ax.bar(x + width/2, g2, width, label='g2')

ax.set_title('Пример групповой диаграммы')
ax.set_xticks(x)
ax.set_xticklabels(cat_par)

ax.legend()
```



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

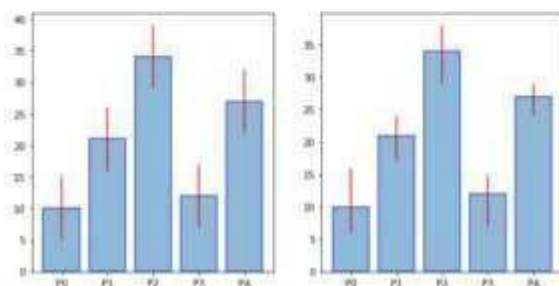
```
np.random.seed(123)

rnd = np.random.randint

cat_par = [f"P{i}" for i in range(5)]
g1 = [10, 21, 34, 12, 27]

error = np.array([[rnd(2,7), rnd(2,7)] for _ in range(len(cat_par))]).T
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].bar(cat_par, g1, yerr=5, ecolor='r', alpha=0.5, edgecolor='k',
           linewidth=2)
axs[1].bar(cat_par, g1, yerr=error, ecolor='r', alpha=0.5, edgecolor='k',
           linewidth=2)
```

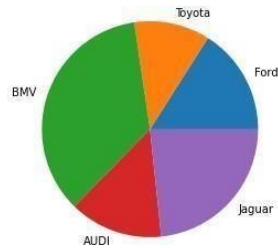


13. Как выполнить построение круговой диаграммы средствами matplotlib?

```
In [34]: vals = [24, 17, 53, 21, 35]
labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]

fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis("equal")
```

```
Out[34]: (-1.1163226287452406,
1.1007772680354877,
-1.1107362350259515,
1.1074836529113834)
```



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.

15. Как отобразить изображение средствами matplotlib?

Рассмотрим две функции для построения цветовой сетки: `imshow()` и `colormesh()`.

```
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))

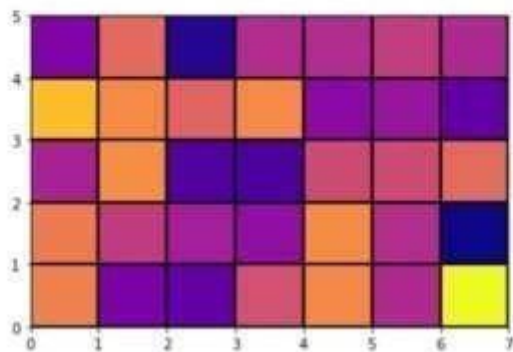
plt.imshow(img)
```

В результате получим изображение логотипа Matplotlib.



```
np.random.seed(123)

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



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')
```