

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

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 3.5 Визуализация
данных с помощью matplotlib»**

**ОТЧЕТ
по лабораторной работе №5
дисциплины
«Основы распознавания образов»**

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Лабораторная работа 3.5 Визуализация данных с помощью matplotlib

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

Ход работы:

18. Дана целочисленная квадратная матрица. Определить:

- сумму элементов в тех строках, которые не содержат отрицательных элементов;
- минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

Поршневым воздушным насосом откачивают сосуд объемом $V = 100$ л. За один цикл (ход поршня) насос захватывает объем, равный $\Delta V = 0.5$ л. Через сколько циклов n давление в сосуде уменьшится в p_0/p_n раз? Процесс считать изотермическим, а газ – идеальным.

Пусть $\eta_n = p_n/p_0$ и $q = V/(V + \Delta V)$, тогда изотермический процесс можно записать в виде $p_{n-1}V = p_n(V + \Delta V)$. Значит, $\eta_n = q^n$, а $n = \frac{\lg \eta_n}{\lg q}$

```
N = 950
V = 0.1
dV = 0.0005
p0 = 1.0e5

qV = V / (V + dV)
n1 = np.log10(0.01) / np.log10(qV);
print (f"n = {n1}")

x = np.empty(N)
p = np.empty(N)
x[0] = 0
p[0] = p0
for i, item in enumerate(x):
    p0 = p0 * qV;
    x[i] = i
    p[i] = p0

plt.plot(x,p)
plt.xlabel('n')
plt.ylabel('η')
plt.show()
```

n = 923.3347082516211

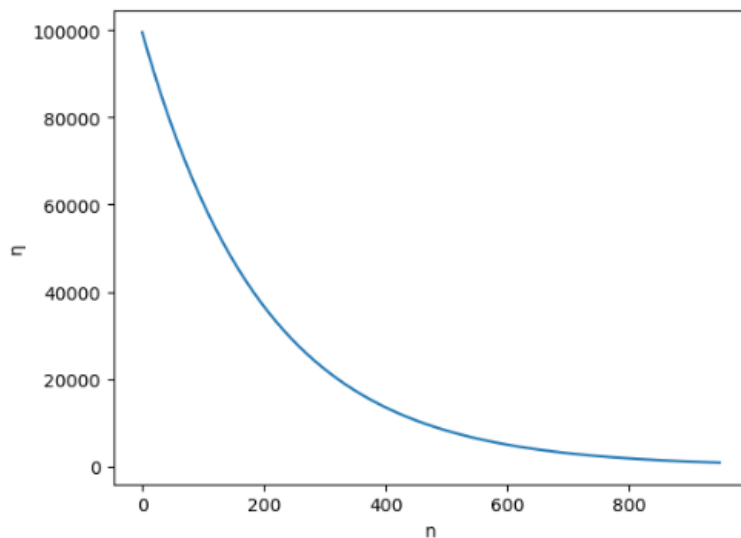


Рисунок 1 – Индивидуальное задание 1

Фóрмула Циолкóвского

определяет скорость, которую развивает летательный аппарат под воздействием тяги ракетного двигателя, неизменной по направлению, при отсутствии всех других сил. Эта скорость называется характеристической скоростью. Формула Циолковского для многоступенчатой ракеты может быть записана в следующем виде:

$$V = \sum_{i=1}^N I_i * g * \ln\left(\frac{M_0 + \sum_{j=i}^N M_{1j}}{M_0 + M_{2i} - M_{1i} + \sum_{j=i}^N M_{1j}}\right), \text{ где:}$$

M_{1i} - масса заправленной i -й ступени ракеты

M_{2i} - масса i -й ступени ракеты без топлива

I_i - удельный импульс двигателя i -й ступени

M_0 - масса полезной нагрузки

N - число ступеней ракеты

Для ракеты СОЮЗ-2:

$$N = 3$$

$$M_{11} = 44413 \text{ кг}$$

$$M_{21} = 3784 \text{ кг}$$

$$I_1 = 263 \text{ с}$$

$$M_{12} = 99765 \text{ кг}$$

$$M_{22} = 6545 \text{ кг}$$

$$I_2 = 255 \text{ с}$$

$$M_{13} = 27755 \text{ кг}$$

$$M_{23} = 2355 \text{ кг}$$

$$I_3 = 326 \text{ с}$$

$$M_0 = 2800 \text{ кг}$$

Рисунок 2 – Индивидуальное задание 2

```

I = np.array([263, 255, 326])
M1 = np.array([44413 * 4, 99765, 27755])
M2 = np.array([3784 * 4, 6545, 2355])
M0 = 2800
N = 3
V = ([0, 0, 0])

for i, mass2 in enumerate(M2):
    M1_sum = np.sum([M1[j] for j in range(i, N)])
    V[i] = 9.8 * I[i] * np.log((M0 + M1_sum) / (M0 + mass2 - M1[i] + M1_sum))

labels = ["1 ступень", "2 ступень", "3 ступень"]
plt.bar(labels, V)

```

<BarContainer object of 3 artists>

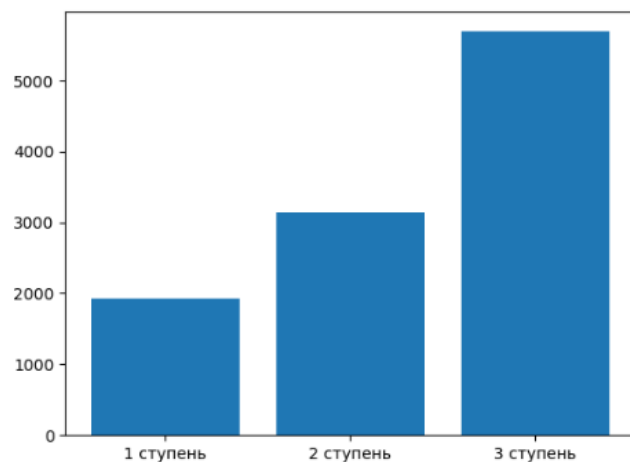


Рисунок 3 – Индивидуальное задание 3

Задача на вероятность

Вероятность изготовления на автоматическом станке стандартной детали равна 0,8. Найти вероятности возможного числа появления бракованных деталей среди 5 отобранных

```

p = 0.8
q = 0.2
n = 5
C = ([np.math.factorial(n)/(np.math.factorial(k)*np.math.factorial(n - k)) for k in range (0, n + 1)])
P = ([C[i] * p**(n-i) * q**i for i in range (0, n + 1)])

labels = ["0 брака", "1 брак", "2 брак", "3 брак", "4 брак", "5 брака"]
fig, ax = plt.subplots()
ax.pie(P, labels=labels, rotatelabels=True)
ax.axis("equal")

(-1.1242383649520327,
1.1011542078548586,
-1.114992393262487,
1.1051563713948485)

```

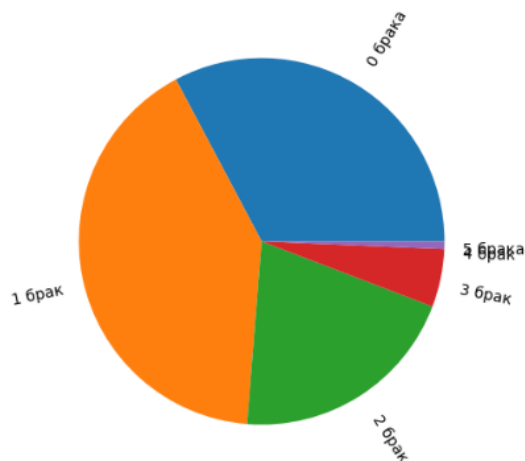


Рисунок 4 – Индивидуальное задание 3

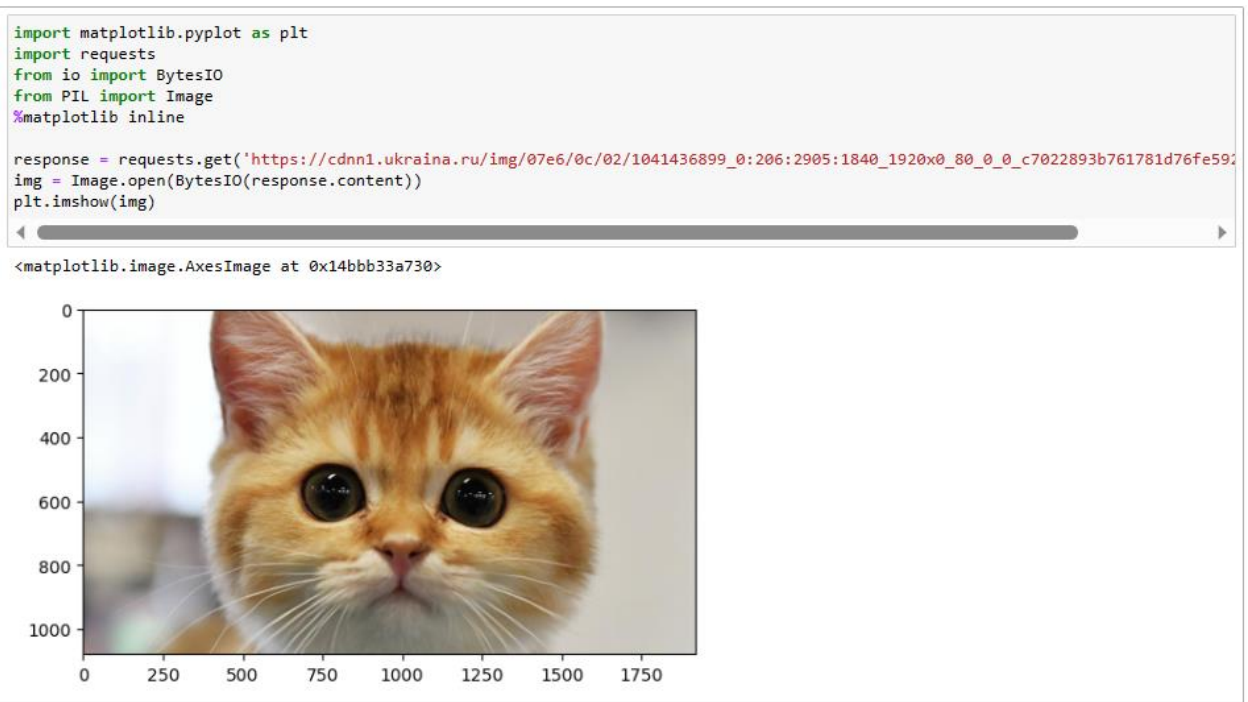


Рисунок 5 – Индивидуальное задание 4

Ответы на контрольные вопросы:

1. Как выполнить построение линейного графика с помощью **matplotlib**? Для построения линейного графика используется функция `plot()`
2. Как выполнить заливку области между графиком и осью? Между двумя графиками? `plt.fill_between(x, y)`
3. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию? `plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))`
4. Как выполнить двухцветную заливку? `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

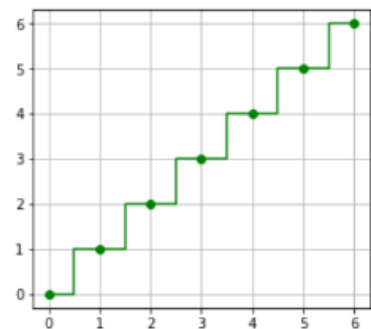
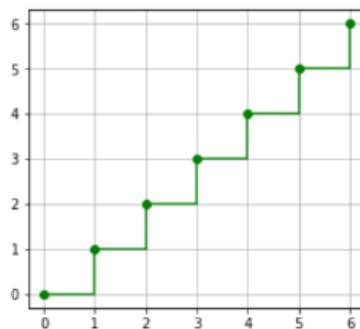
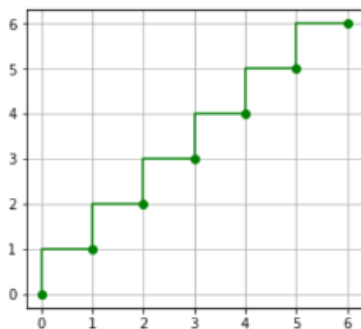
предварительное маскирование данных с помощью функции `masked_where` из пакета `numpy`.

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?

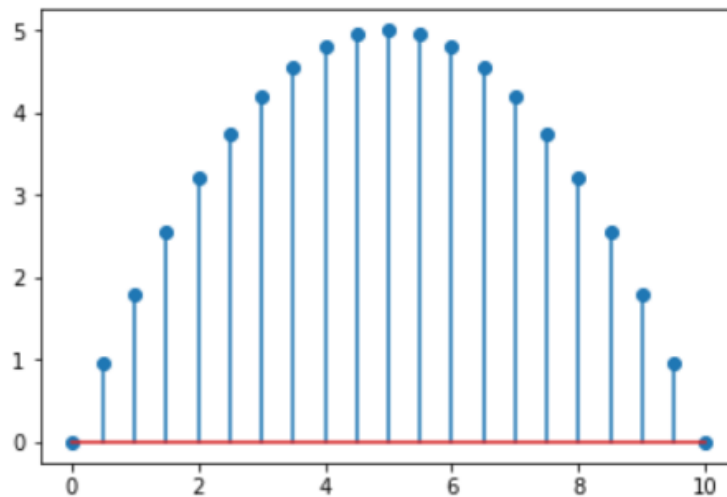
Для построения стекового графика используется функция `stackplot()`. Суть его в том, что графики отображаются друг над другом, и каждый следующий является суммой предыдущего и заданного набора данных.

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])

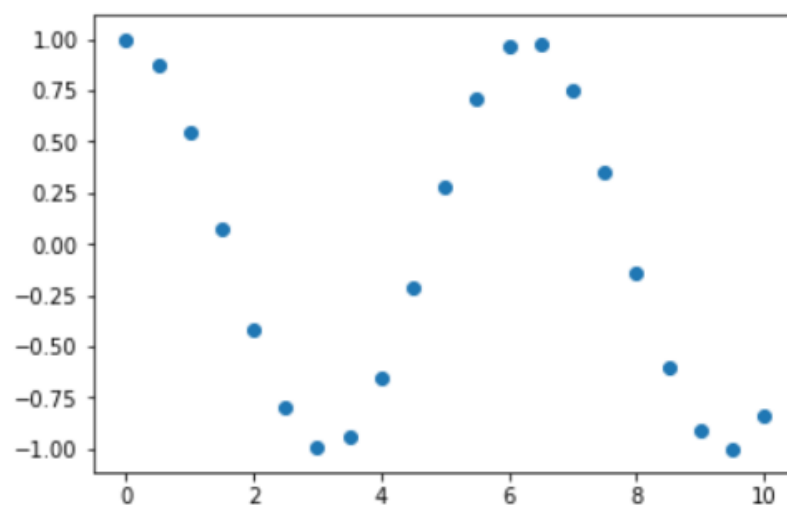
plt.stem(x, y)
```



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?

Для визуализации категориальных данных хорошо подходят столбчатые диаграммы. Для их построения используются функции:

`bar()` – для построения вертикальной диаграммы

`barh()` – для построения горизонтальной диаграммы.

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]

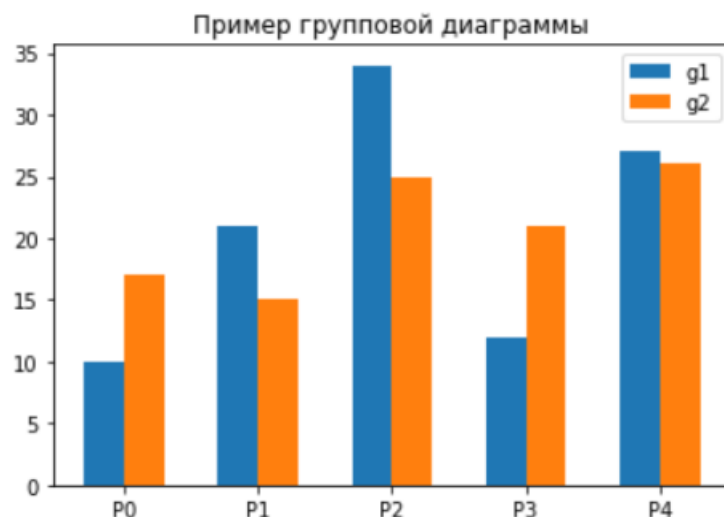
width = 0.3

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 `xerr`, `yerr` и `ecolor` (для задания цвета)


```

np.random.seed(123)

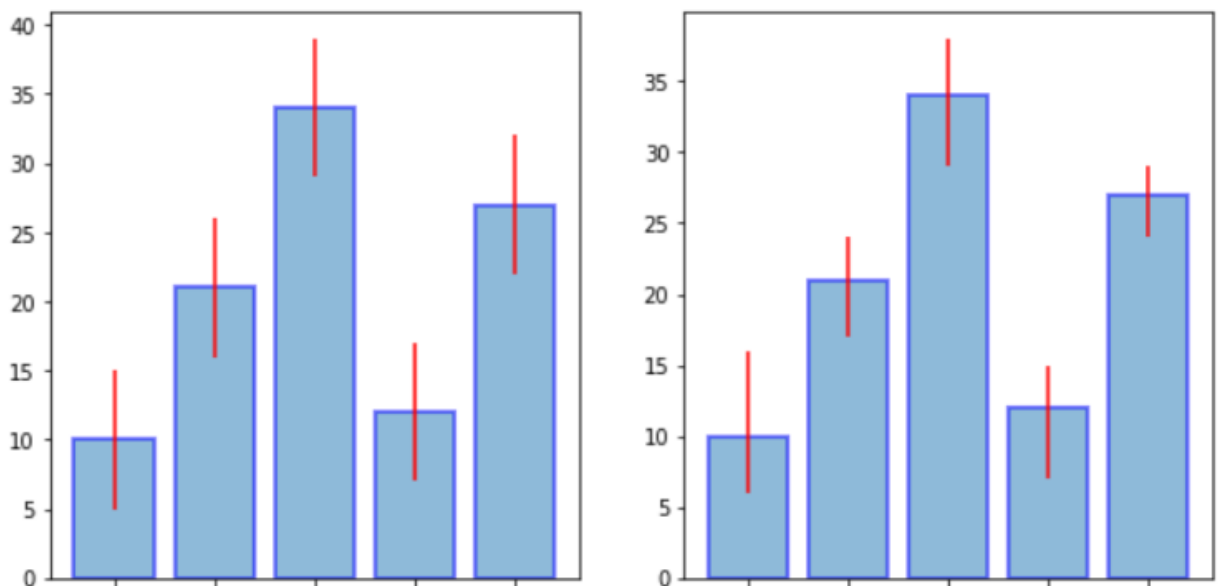
rnd = np.random.randint

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="b",
linewidth=2)
axs[1].bar(cat_par, g1, yerr=error, ecolor="r", alpha=0.5, edgecolor="b",
linewidth=2)

```



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.

15. Как отобразить изображение средствами matplotlib?

```

from PIL import Image
import requests

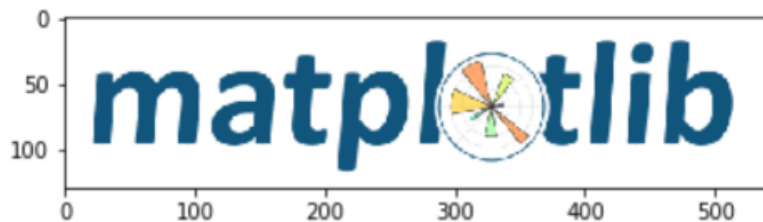
from io import BytesIO

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

plt.imshow(img)

```

В результате получим изображение логотипа *Matplotlib*.



16. Как отобразить тепловую карту средствами matplotlib?

Рассмотрим ещё одну функцию для визуализации 2D наборов данных – `pcolormesh()`. В библиотеке Matplotlib есть ещё одна функция с аналогичным функционалом – `pcolor()`, в отличие от нее рассматриваемая нами `pcolormesh()` более быстрая и является лучшим вариантом в большинстве случаев. Функция `pcolormesh()` похожа по своим возможностям на `imshow()`, но есть и отличия.

```

np.random.seed(123)

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

```

