

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №11  
на тему: «Визуализация данных с помощью matplotlib»  
Дисциплина «Введение в системы искусственного интеллекта»**

Выполнил: студент группы ИВТ-б-о-18-1 (1)  
Скориков А.Ю.

\_\_\_\_\_ (подпись)

Проверил: доцент кафедры  
инфокоммуникаций  
Воронкин Роман Александрович

\_\_\_\_\_ (подпись)

Ставрополь, 2022 г.

**Цель работы:** исследовать базовые возможности визуализации данных на плоскости средствами библиотеки matplotlib языка программирования Python.

## Ход работы

Таблица 1 – Исходные данные

Номер варианта	10
----------------	----

Решение:

```
Ввод [1]: # -*- coding: utf-8 -*-
#Разбор работы функции plot()
import matplotlib.pyplot as plt
plt.plot()

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)
```

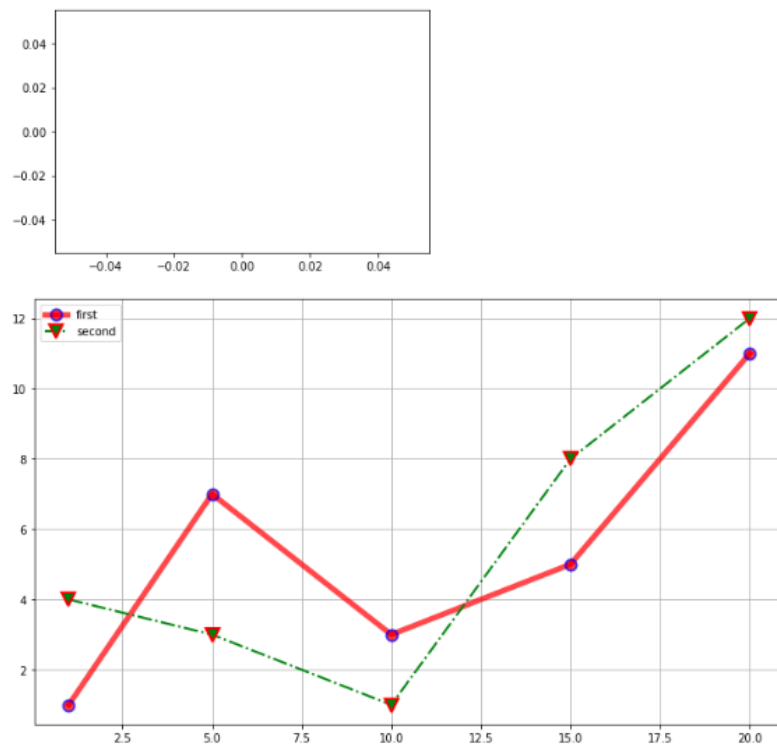


Рисунок 1 – Решение задачи 1

```

Ввод [2]: # -*- coding: utf-8 -*-
           #Разбор работы функции plot()
           import numpy as np
           import matplotlib.pyplot as plt

           x = np.arange(0.0, 5, 0.01)
           y = np.cos(x*np.pi)
           #Отобразим график с заливкой:
           plt.plot(x, y, c = "r")
           plt.fill_between(x, y)

```

Out[2]: <matplotlib.collections.PolyCollection at 0x136fd7e5250>

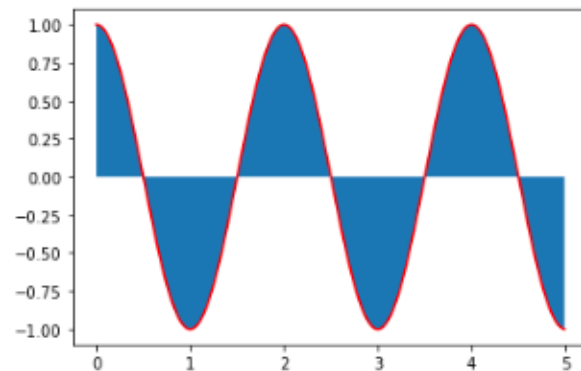


Рисунок 2 – Решение задачи 2

```

Ввод [3]: # -*- coding: utf-8 -*-
           #Настройка маркировки графиков

           import numpy as np
           import matplotlib.pyplot as plt

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

```

Out[3]: [<matplotlib.lines.Line2D at 0x136fd6ca5e0>]

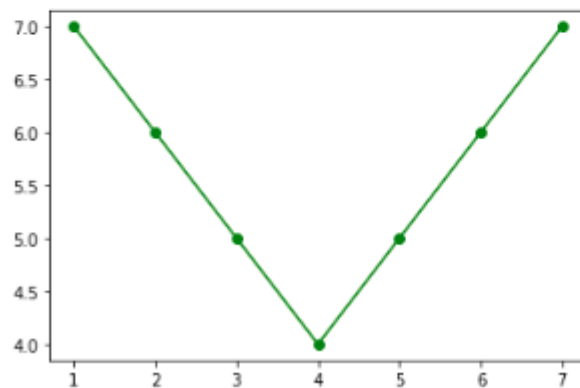


Рисунок 3 – Решение задачи 3

```

Ввод [4]: # -*- coding: utf-8 -*-
#настройка маркировки графиков
import numpy as np
import matplotlib.pyplot as plt
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)

```

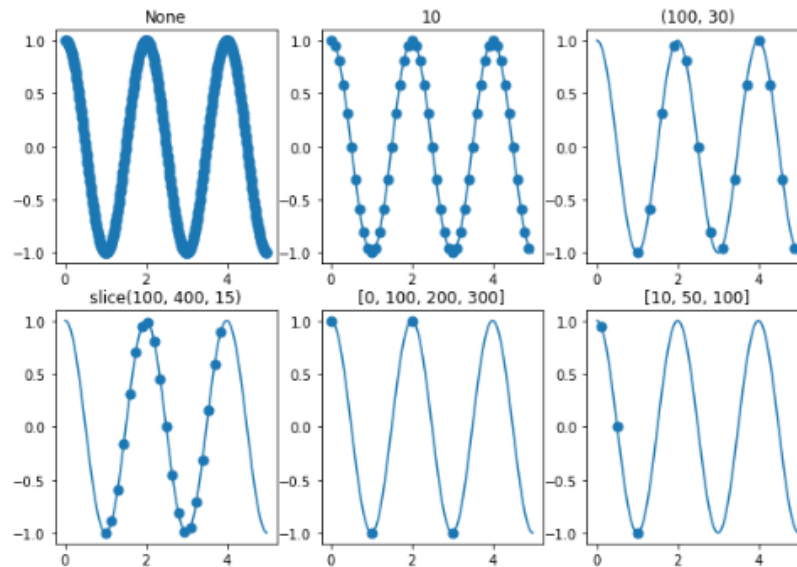


Рисунок 4 – Решение задачи 4

```

Ввод [5]: 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[5]: [<matplotlib.lines.Line2D at 0x136fd9e5880>]

```

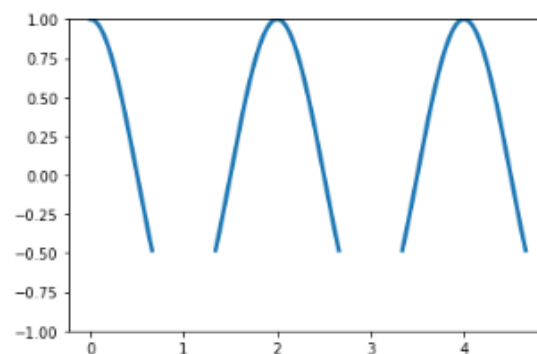


Рисунок 5 – Решение задачи 5

```
Ввод [6]: # -*- coding: utf-8 -*-
#Пример построения ступенчатого графика
import numpy as np
import matplotlib.pyplot as plt

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

where_set = ['pre', 'post', 'mid']
fig, axes = plt.subplots(1, 3, figsize=(15, 4))
for i, ax in enumerate(axes):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()
```

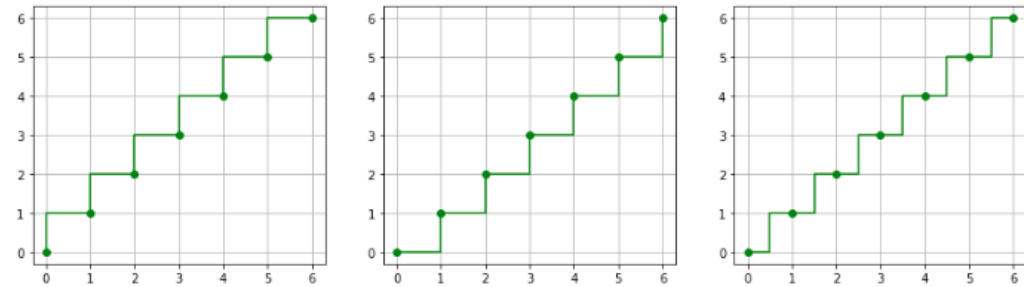


Рисунок 6 – Решение задачи 6

```
Ввод [7]: # -*- coding: utf-8 -*-
#Пример построения стекового графика с помощью функции stackplot().
import numpy as np
import matplotlib.pyplot as plt

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')
```

Out[7]: <matplotlib.legend.Legend at 0x136fdb0c820>

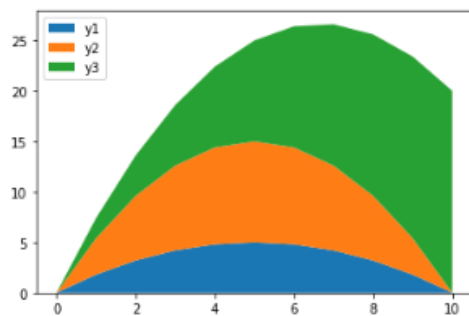


Рисунок 7 – Решение задачи 7

```
Ввод [8]: # -*- coding: utf-8 -*-
#Пример построения Stem-графика

import numpy as np
import matplotlib.pyplot as plt
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)
```

Out[8]: <StemContainer object of 3 artists>

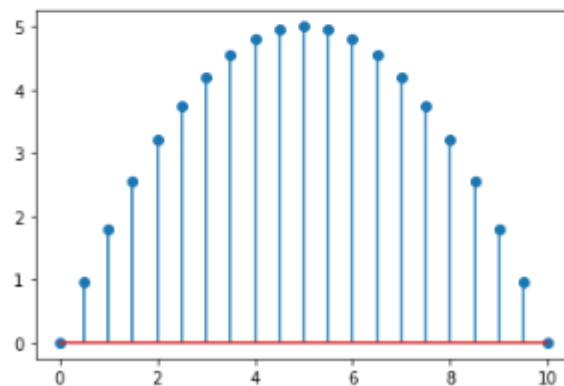


Рисунок 8 – Решение задачи 8

```
Ввод [9]: # -*- coding: utf-8 -*-
#Пример построения точечного графика с помощью функции scatter().
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 10.5, 0.5)
y = np.cos(x)
plt.scatter(x, y)
```

Out[9]: <matplotlib.collections.PathCollection at 0x136fdc3f310>

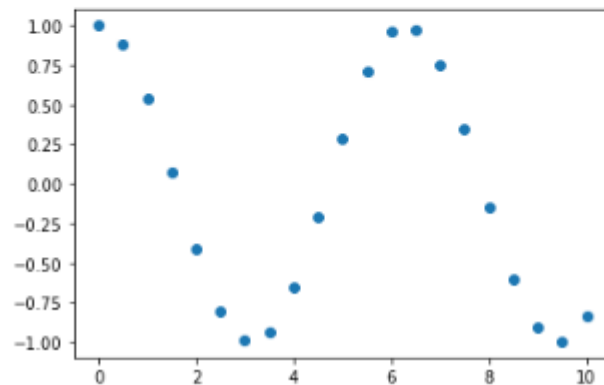


Рисунок 9 – Решение задачи 9

```

Ввод [10]: # -*- coding: utf-8 -*-
            #Пример построения точечного графика с помощью функции scatter().

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
bc = mcolors.BASE_COLORS
x = np.arange(0, 10.5, 0.25)
y = np.cos(x)
num_set = np.random.randint(1, len(mcolors.BASE_COLORS), len(x))
sizes = num_set * 35
colors = [list(bc.keys())[i] for i in num_set]
plt.scatter(x, y, s=sizes, alpha=0.4, c=colors, linewidths=2, edgecolors="face")
plt.plot(x, y, "g--", alpha=0.4)

Out[10]: [<matplotlib.lines.Line2D at 0x136feccb6d0>]

```

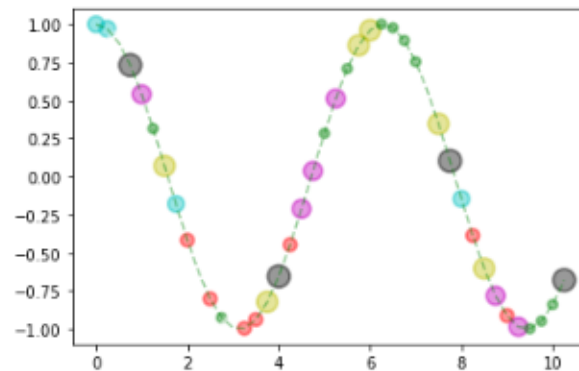


Рисунок 10 – Решение задачи 10

```

Ввод [11]: # -*- coding: utf-8 -*-
            #Пример построения точечного графика с помощью функции scatter() с изменением цвета

import numpy as np
import matplotlib.pyplot as plt
np.random.seed(123)
groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(3, 10, len(groups))
plt.bar(groups, counts)

Out[11]: <BarContainer object of 7 artists>

```

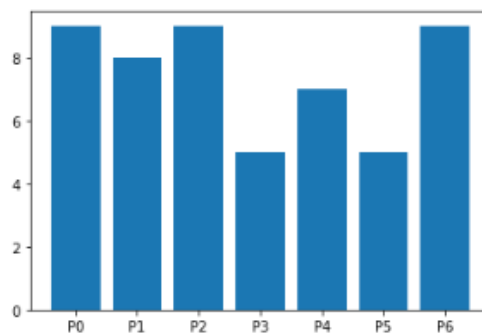


Рисунок 11 – Решение задачи 11

```

Ввод [13]: import matplotlib.colors as mcolors
bc = mcolors.BASE_COLORS
np.random.seed(123)
groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(0, len(bc), len(groups))
width = counts*0.1
colors = [{"r", "b", "g"}[int(np.random.randint(0, 3, 1))] for _ in counts]
plt.bar(groups, counts, width=width, alpha=0.6, bottom=2, color=colors,
edgecolor="k", linewidth=2)

```

Out[13]: <BarContainer object of 7 artists>

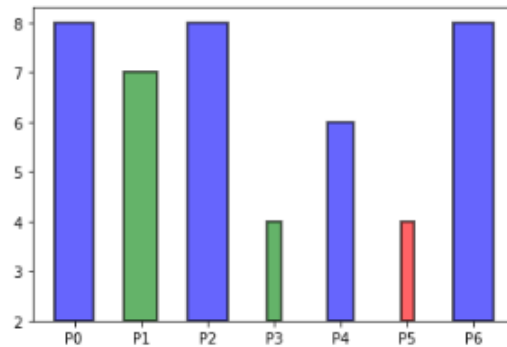


Рисунок 12 – Решение задачи 12

```

Ввод [14]: # -*- coding: utf-8 -*-

#Пример построения диаграмма с errorbar элементом.Для этого
#используются параметры xerr, yerr и ecolor (для задания цвета):
import numpy as np
import matplotlib.pyplot as plt

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)

```

Out[14]: <BarContainer object of 5 artists>

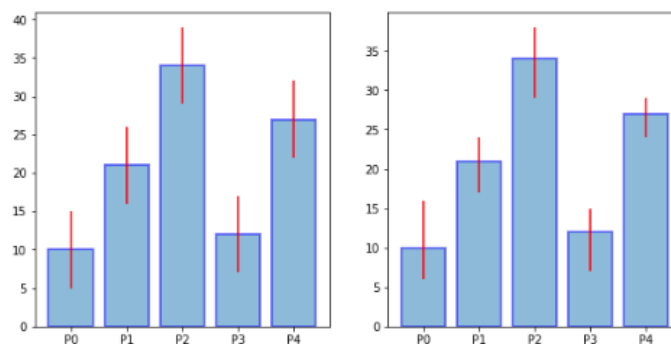


Рисунок 13 – Решение задачи 13



```
Ввод [15]: # -*- coding: utf-8 -*-
#Пример построения Классической круговой диаграммы

import numpy as np
import matplotlib.pyplot as plt

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[15]: (-1.1163226287452406,
1.1007772680354877,
-1.1107362350259515,
1.1074836529113834)
```

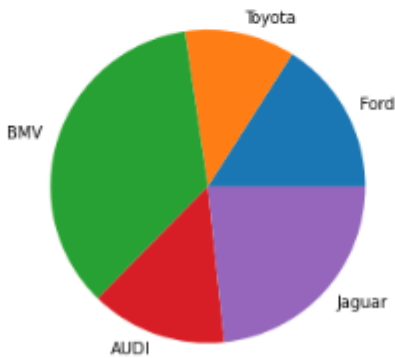


Рисунок 14 – Решение задачи 14

```
Ввод [16]: # -*- coding: utf-8 -*-
#Напишем простую программу, которая
#загружает картинку из интернета по заданному URL и отображает ее с использованием
#библиотеки Matplotlib:

import matplotlib.pyplot as plt
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[16]: <matplotlib.image.AxesImage at 0x136ff06a220>
```



Рисунок 15 – Решение задачи 15

```
Ввод [17]: # -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(123)
data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')

Out[17]: <matplotlib.collections.QuadMesh at 0x136ff0d51f0>
```

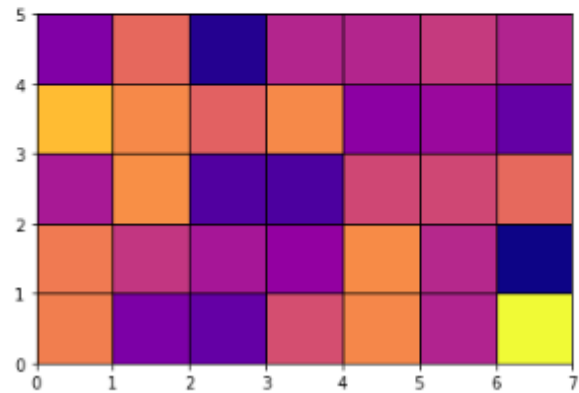


Рисунок 16 – Решение задачи 16

**Вывод:** были получены возможности визуализации данных на плоскости средствами библиотеки matplotlib языка программирования Python.