

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчёт о лабораторной работе №5 по дисциплине технологии распознавания
образов**

Выполнила:

Нестеренко Тамара Антоновна,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ

1. Практическая часть

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

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)
```

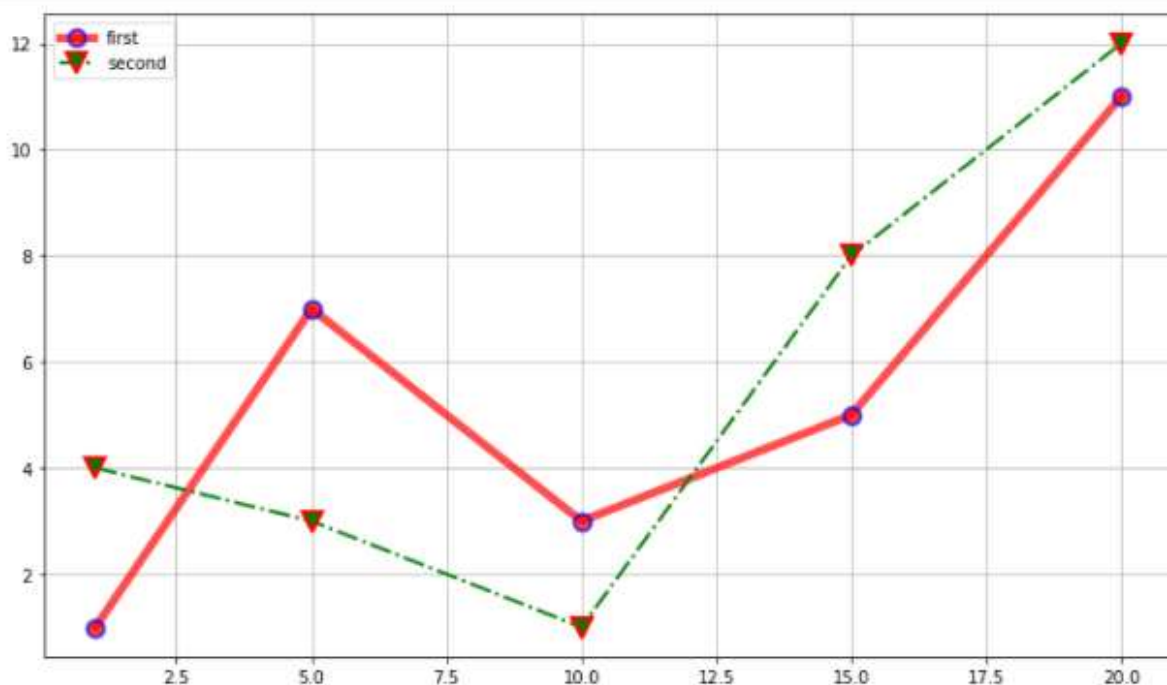


Рисунок 1 – Пример построения графика с помощью plot()

```
In [5]: import numpy as np

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

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

Out[5]: <matplotlib.collections.PolyCollection at 0x1b757ff6c70>

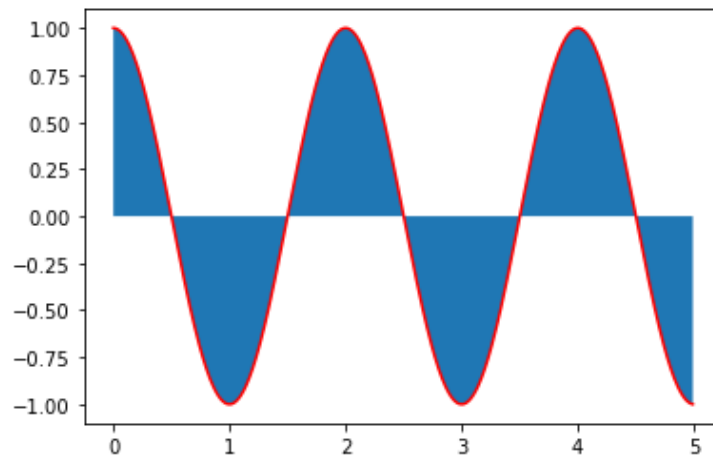


Рисунок 2 – Пример заливки графика

```
In [6]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))
```

Out[6]: <matplotlib.collections.PolyCollection at 0x1b757fc39d0>

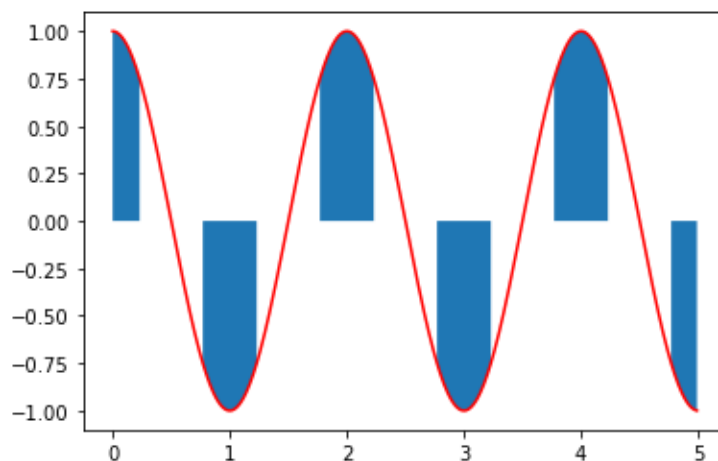


Рисунок 3 – Пример изменения правил заливки графика

```
In [11]: plt.plot(x, y, c="r")  
plt.fill_between(x, y, where=(y > 0))
```

```
Out[11]: <matplotlib.collections.PolyCollection at 0x1b758283220>
```

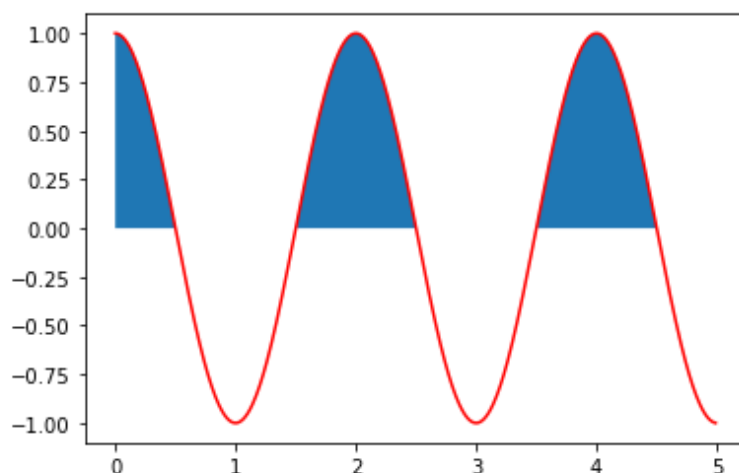


Рисунок 4 – Пример заливки графика между 0 и y , при условии, что $y \geq 0$

```
In [12]: plt.plot(x, y, c="r")  
plt.grid()  
  
plt.fill_between(x, 0.5, y, where=(y >= 0.5))
```

```
Out[12]: <matplotlib.collections.PolyCollection at 0x1b7582e8b80>
```

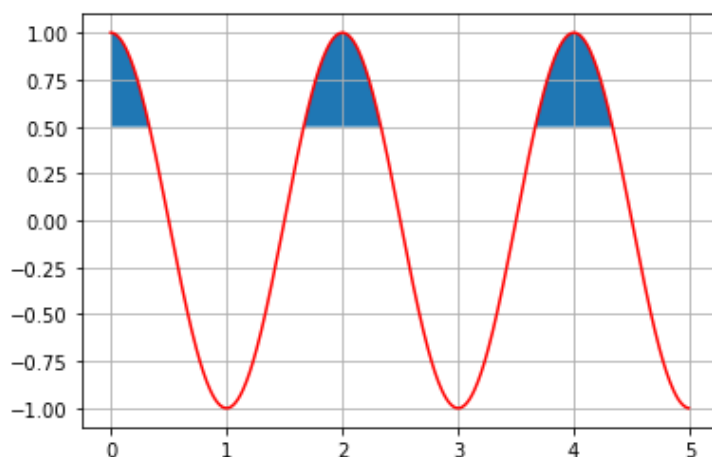


Рисунок 5 – Пример заливки области между 0.5 и y , при условии, что $y \geq 0.5$

```
In [13]: plt.plot(x, y, c="r")
plt.grid()

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

Out[13]: <matplotlib.collections.PolyCollection at 0x1b75835edf0>

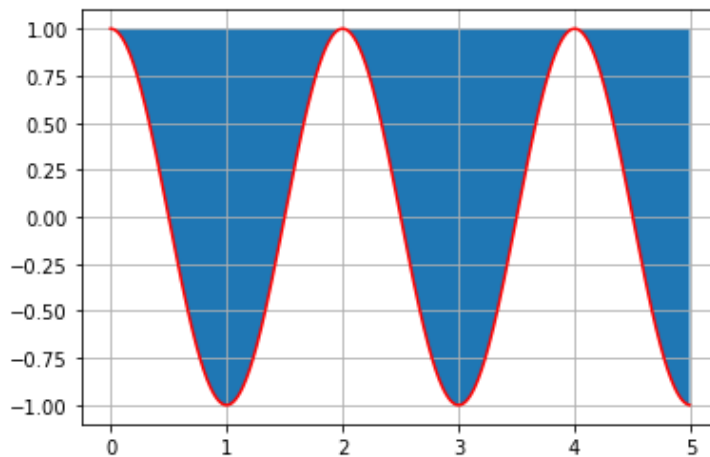


Рисунок 6 – Пример заливки область между y и 1

```
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>

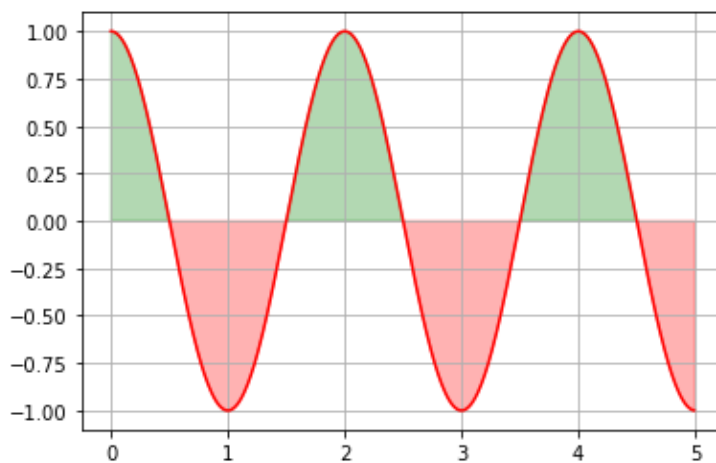


Рисунок 7 – Пример двухцветной заливки

```
In [15]: 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[15]: [<matplotlib.lines.Line2D at 0x1b7594b4280>]
```

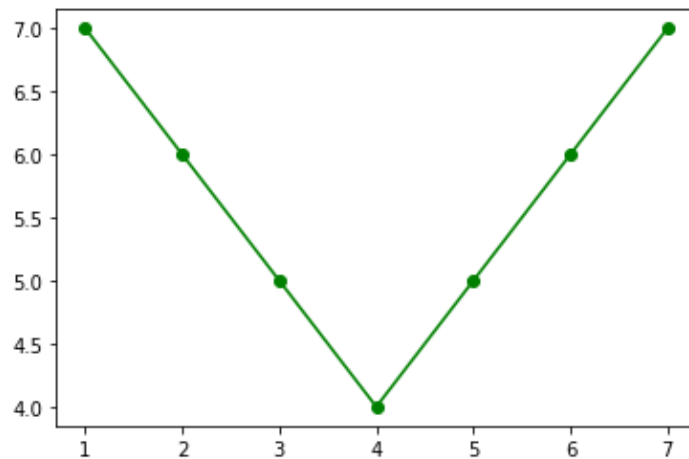


Рисунок 8 – Пример маркировки графика

```
In [17]: import numpy as np

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

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

```
Out[17]: [<matplotlib.lines.Line2D at 0x1b75951d580>]
```

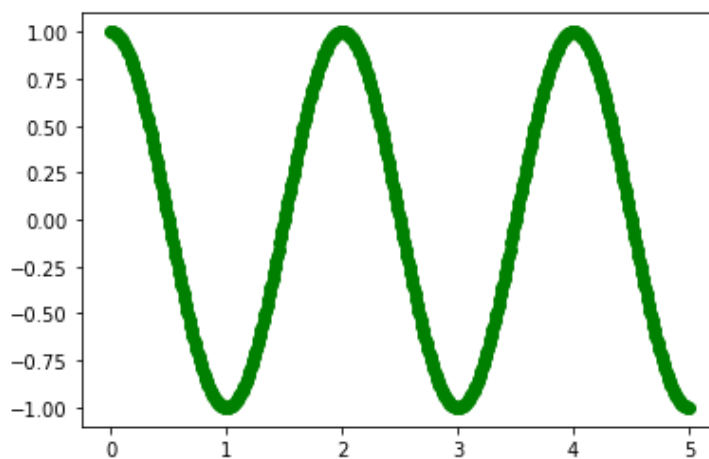


Рисунок 9 – Пример маркировки графика

```
In [19]: 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)
```

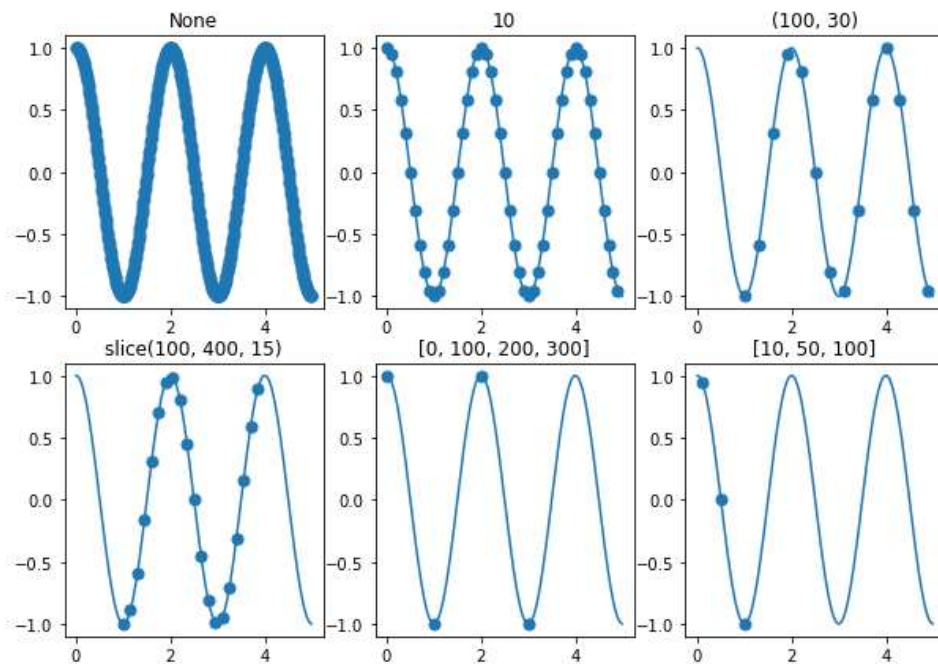


Рисунок 10 – Пример маркировки графика

```
In [20]: 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[20]: [<matplotlib.lines.Line2D at 0x1b759771f10>]

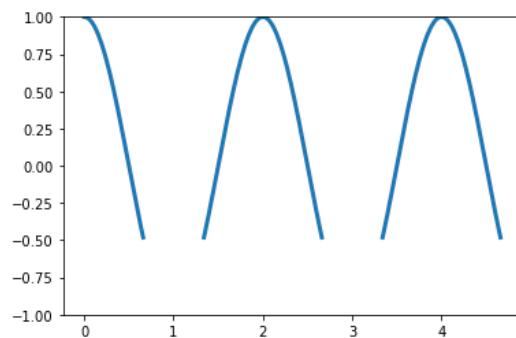


Рисунок 11 – Пример обрезки графика

```
In [21]: 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()
```

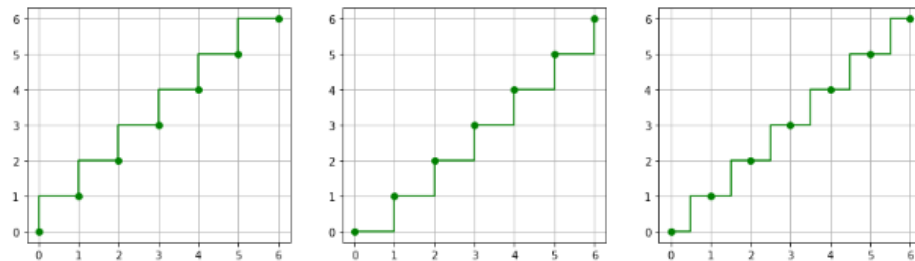


Рисунок 12 – Пример построения ступенчатого графика

```
In [22]: 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[22]: <matplotlib.legend.Legend at 0x1b759a44640>

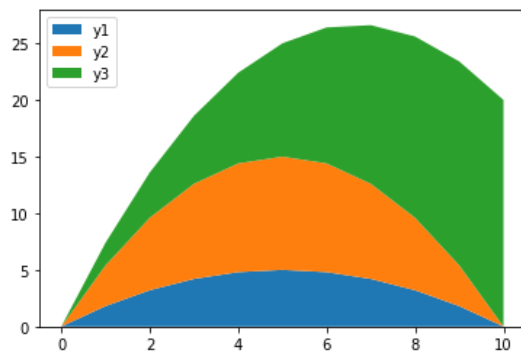


Рисунок 13 – Пример стекового графика


```
In [23]: 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[23]: <StemContainer object of 3 artists>

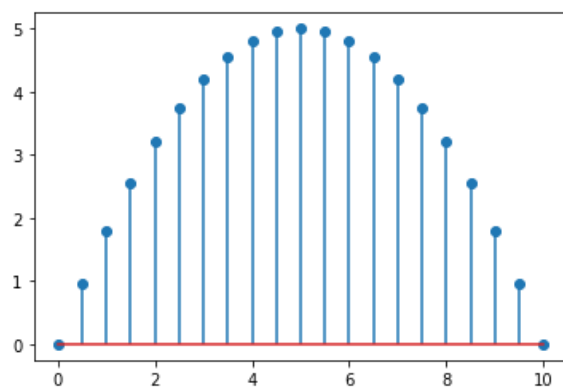


Рисунок 14 – Пример stem-графика

```
In [24]: plt.stem(x, y, linefmt="r--", markerfmt="^", bottom=1)
```

Out[24]: <StemContainer object of 3 artists>

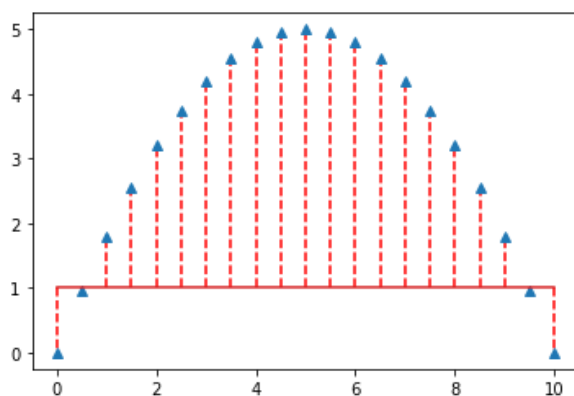


Рисунок 15 – Пример stem-графика

```
In [25]: x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.scatter(x, y)
```

Out[25]: <matplotlib.collections.PathCollection at 0x1b759a7e610>

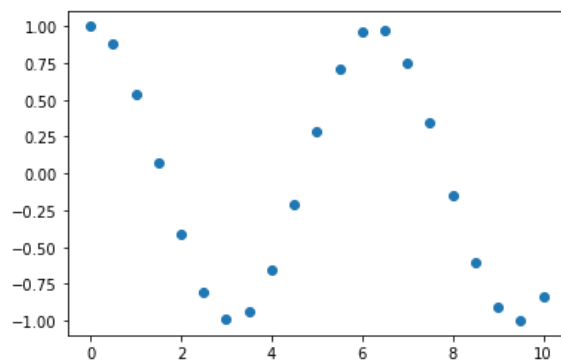


Рисунок 16 – Пример построения точечного графика

```
In [27]: x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.scatter(x, y, s=80, c="r", marker="D", linewidths=2, edgecolors="g")
```

Out[27]: <matplotlib.collections.PathCollection at 0x1b7599eef70>

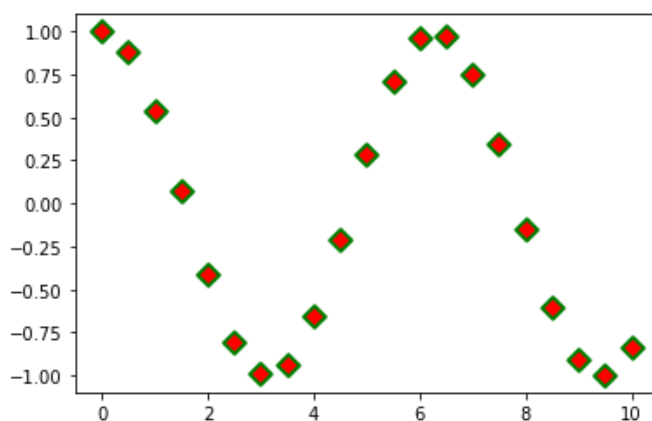


Рисунок 17 – Пример построения точечного графика

```
In [28]: 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[28]: [<matplotlib.lines.Line2D at 0x1b7596b8d30>]
```

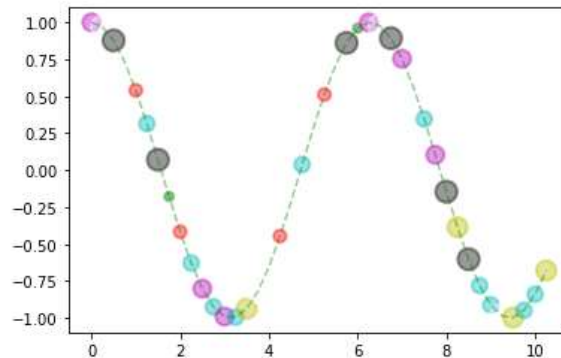


Рисунок 18 – Пример построения точечного графика

```
In [29]: 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[29]: <BarContainer object of 7 artists>

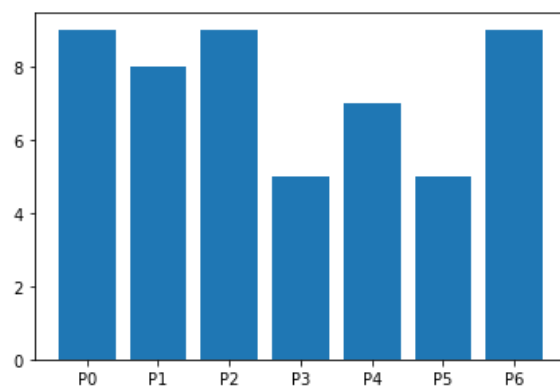


Рисунок 19 – Пример построения столбчатой диаграммы

```
In [30]: plt.barh(groups, counts)
```

```
Out[30]: <BarContainer object of 7 artists>
```

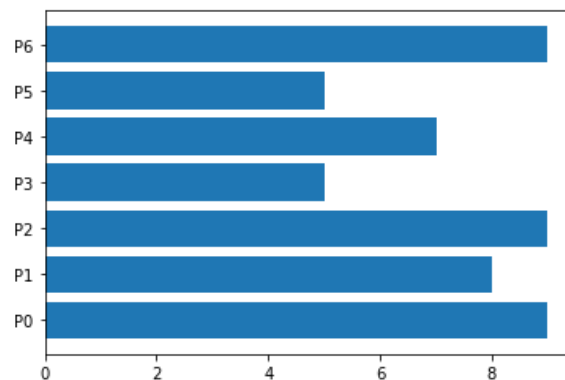


Рисунок 20 – Пример построения столбчатой диаграммы

```
In [31]: 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[31]: <BarContainer object of 7 artists>
```

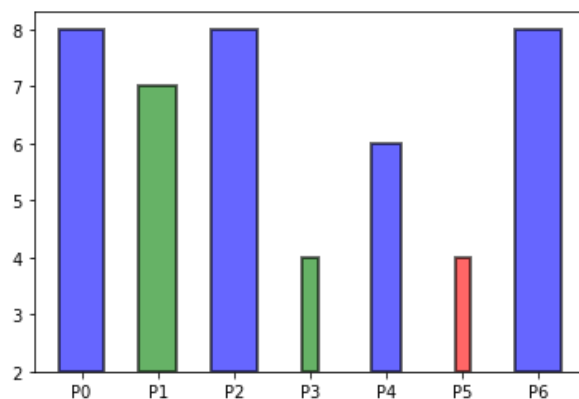


Рисунок 21 – Пример построения столбчатой диаграммы

```

In [32]: 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()

```

Out[32]: <matplotlib.legend.Legend at 0x1b75962ea60>

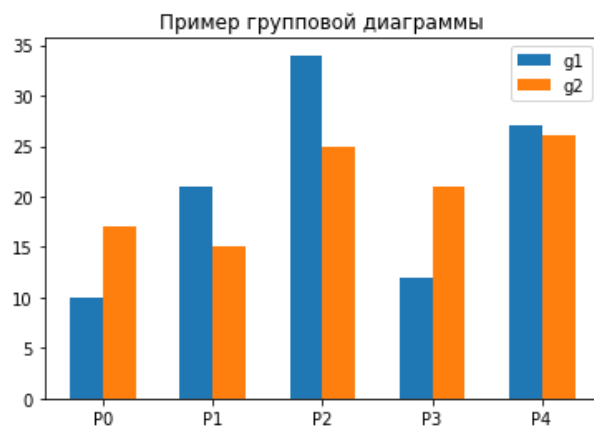


Рисунок 22 – Пример построения групповых столбчатых диаграмм

```

In [33]: 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[33]: <BarContainer object of 5 artists>

```

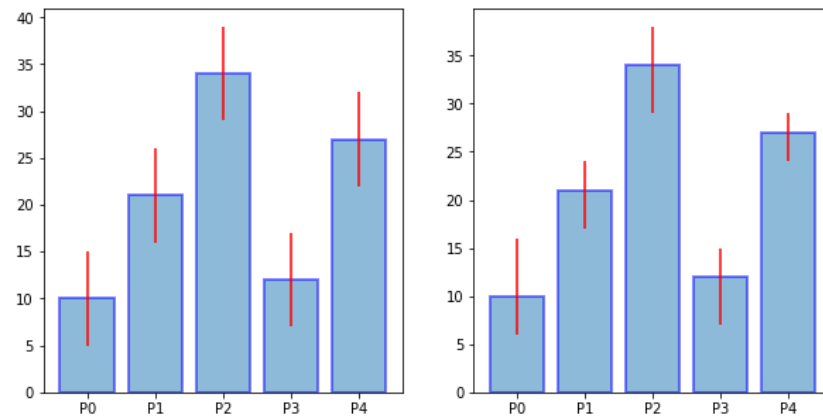


Рисунок 23 – Пример построения диаграмм с errorbar элементом

```

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)

```

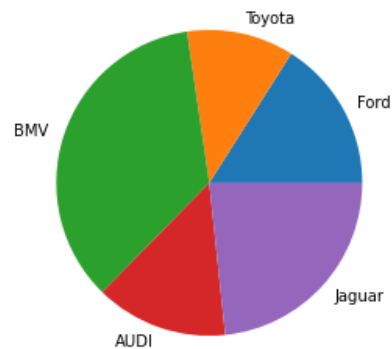


Рисунок 24 – Пример построения круговых диаграмм

```
In [35]: vals = [24, 17, 53, 21, 35]
labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]
explode = (0.1, 0, 0.15, 0, 0)

fig, ax = plt.subplots()

ax.pie(vals, labels=labels, autopct='%1.1f%%', shadow=True, explode=explode,
wedgeprops={'lw':1, 'ls':'--', 'edgecolor':'k'}, rotatelabels=True)

ax.axis("equal")

Out[35]: (-1.2704955621219602,
1.1999223938155328,
-1.1121847055183558,
1.1379015332518725)
```

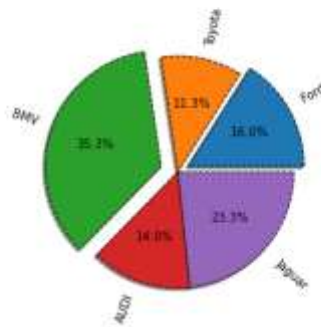


Рисунок 25 – Пример построения круговых диаграмм

```
In [36]: fig, ax = plt.subplots()

offset=0.4

data = np.array([[5, 10, 7], [8, 15, 5], [11, 9, 7]])
cmap = plt.get_cmap("tab20b")

b_colors = cmap(np.array([0, 8, 12]))
sm_colors = cmap(np.array([1, 2, 3, 9, 10, 11, 13, 14, 15]))

ax.pie(data.sum(axis=1), radius=1, colors=b_colors,
wedgeprops=dict(width=offset, edgecolor='w'))

ax.pie(data.flatten(), radius=1+offset, colors=sm_colors,
wedgeprops=dict(width=offset, edgecolor='w'))

Out[36]: ([<matplotlib.patches.Wedge at 0x1b759d41a90>,
<matplotlib.patches.Wedge at 0x1b759d41fd0>,
<matplotlib.patches.Wedge at 0x1b759d4e520>,
<matplotlib.patches.Wedge at 0x1b759d4ea00>,
<matplotlib.patches.Wedge at 0x1b759d4eee0>,
<matplotlib.patches.Wedge at 0x1b759d5b400>,
<matplotlib.patches.Wedge at 0x1b759d5b8e0>,
<matplotlib.patches.Wedge at 0x1b759d5bdc0>,
<matplotlib.patches.Wedge at 0x1b759d602e0>],
[Text(0.646314344414094, 0.13370777166859046, ''),
Text(0.4521935266177387, 0.48075047008298655, ''),
Text(0.040366679721656945, 0.6587643973138266, ''),
Text(-0.34542288787409087, 0.5623904591409097, ''),
Text(-0.6578039053946477, 0.05379611554331286, ''),
Text(-0.48987451889717687, -0.44229203934431896, ''),
Text(-0.12040006360635531, -0.6489073112075174, ''),
Text(0.39011356818311405, -0.532363976917521, ''),
Text(0.6332653697075483, -0.1859434632601054, '')])
```



Рисунок 26 – Пример построения вложенных круговых диаграмм

```
In [37]: vals = [24, 17, 53, 21, 35]

labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]

fig, ax = plt.subplots()
ax.pie(vals, labels=labels, wedgeprops=dict(width=0.5))
```

```
Out[37]: ([<matplotlib.patches.Wedge at 0x1b759daa280>,
<matplotlib.patches.Wedge at 0x1b759daa700>,
<matplotlib.patches.Wedge at 0x1b759daaa30>,
<matplotlib.patches.Wedge at 0x1b759daaf10>,
<matplotlib.patches.Wedge at 0x1b759dba430>],
[Text(0.9639373540021144, 0.5299290306818474, 'Ford'),
Text(0.22870287165240302, 1.075962358309037, 'Toyota'),
Text(-1.046162158377023, 0.3399187231970734, 'BMV'),
Text(-0.3617533684721028, -1.0388139873909512, 'AUDI'),
Text(0.8174592712713289, -0.7360437078139777, 'Jaguar')])
```

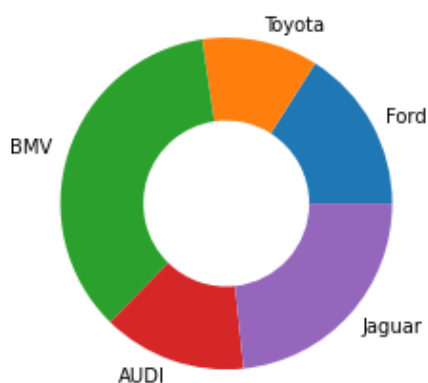


Рисунок 27 – Пример построения круговой диаграммы в виде бублика

```
In [38]: 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[38]: <matplotlib.image.AxesImage at 0x1b75b07c220>
```

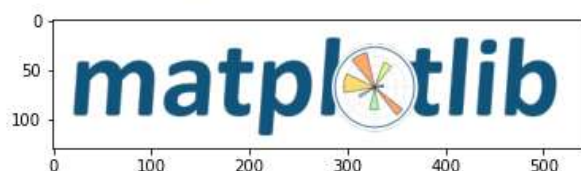


Рисунок 28 – Пример отображения изображения


```
In [41]: np.random.seed(19680801)

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

Out[41]: <matplotlib.image.AxesImage at 0x1b759c4b7f0>

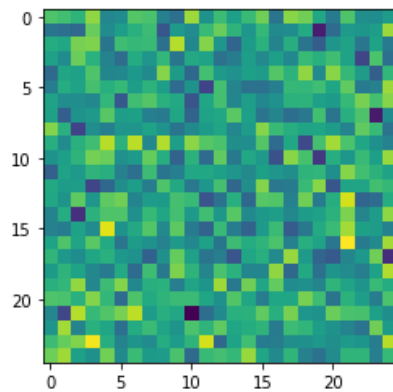


Рисунок 29 – Пример создания двумерного набора данных

```
In [42]: 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])
```

Out[42]: <matplotlib.colorbar.Colorbar at 0x1b759cafd60>

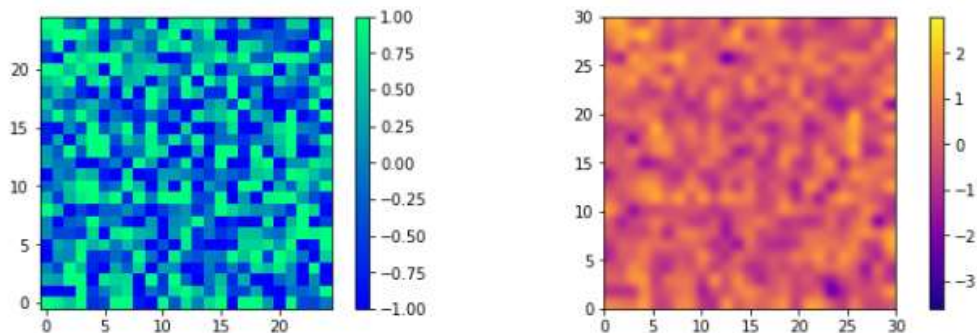


Рисунок 30 – Пример создания двумерного набора данных

```
In [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 0x1b75b174940>
```

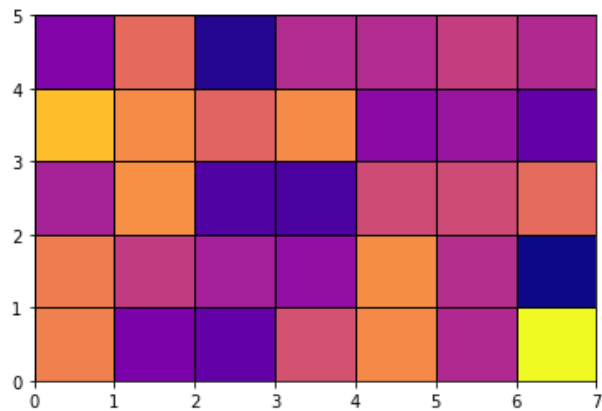


Рисунок 31 – Пример отображения тепловой карты

2. Вопросы для защиты

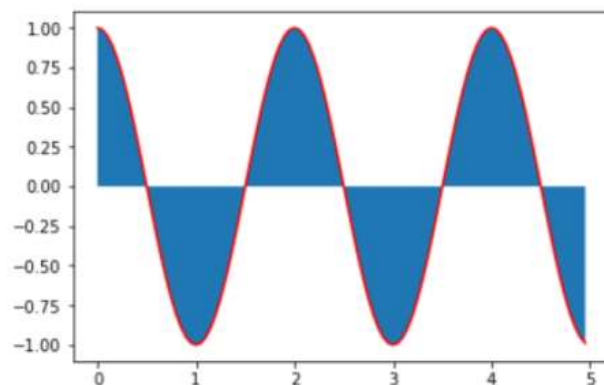
1. Как выполнить построение линейного графика с помощью matplotlib?

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)
```

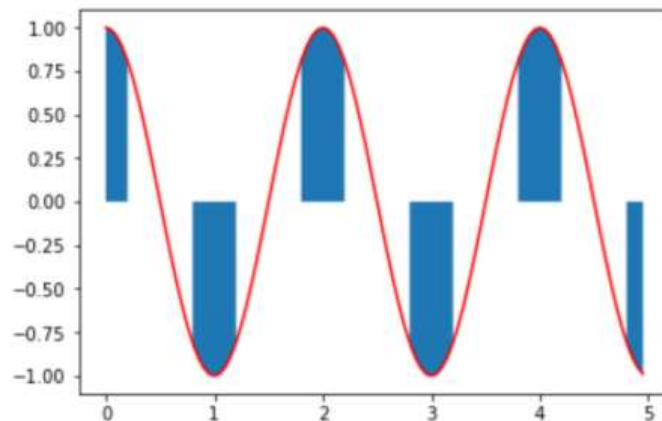
2. Как выполнить заливку области между графиком и осью? Между двумя графиками?

```
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))
```

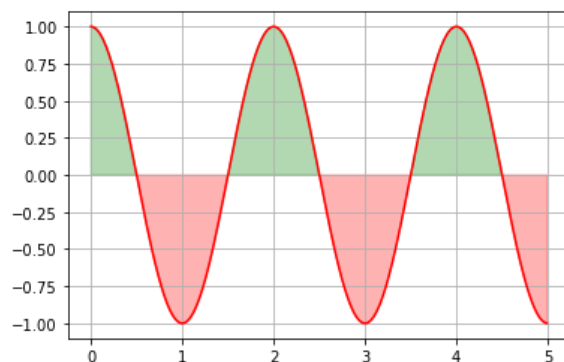


4. Как выполнить двухцветную заливку?

```
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)

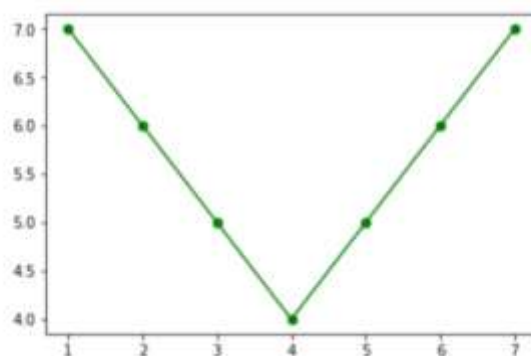
Out[14]: <matplotlib.collections.PolyCollection at 0x1b7583e04f0>
```



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")
```

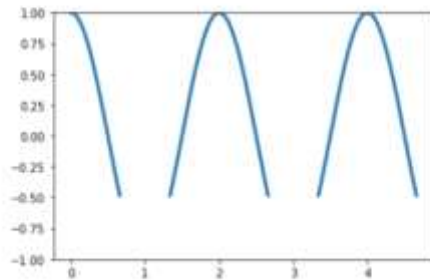


6. Как выполнить обрезку графиков?

```
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)
```

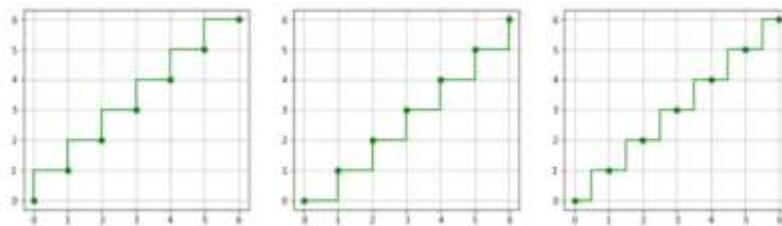


7. Как построить ступенчатый график? В чем особенность ступенчатого графика?

```
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.

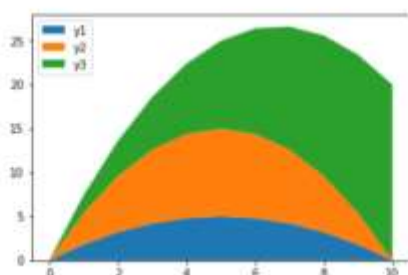
```
x = np.arange(0, 11, 1)

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

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

fig, ax = plt.subplots()

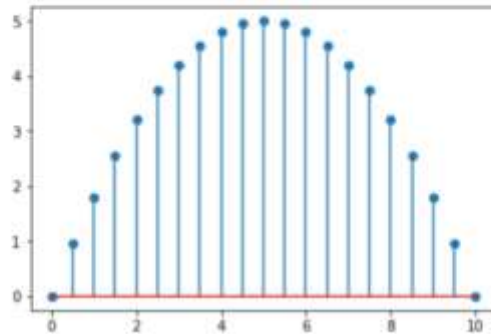
ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')
```



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.

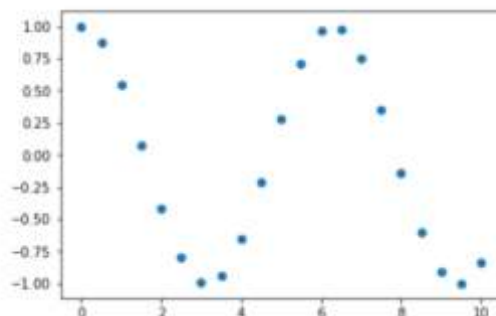
```
x = np.arange(0, 10.5, 0.5)
y = np.array([(-0.2)**i**2+2*i for i in x])
plt.stem(x, y)
```



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

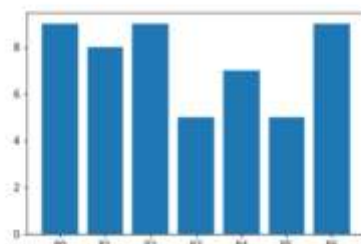
Для отображения точечного графика предназначена функция `scatter()`. В простейшем виде точечный график можно получить передав функции `scatter()` наборы точек для x, y координат.

```
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(3, 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]

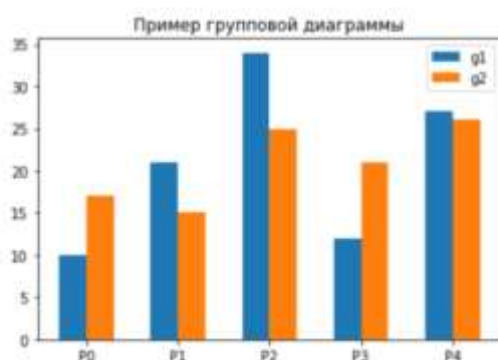
width = 0.3

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 `herr`, `yerr` и `ecolor` (для задания цвета).

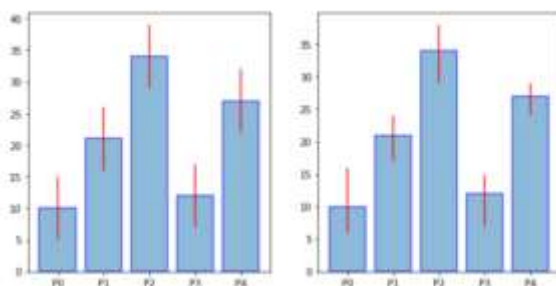
```
np.random.seed(123)

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

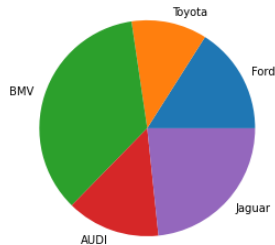


13. Как выполнить построение круговой диаграммы средствами matplotlib?

```
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 `pcolormesh()`.

```
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))

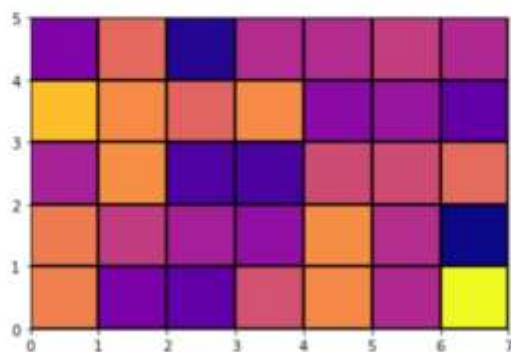
plt.imshow(img)
```

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')
```

Ссылка на репозиторий: https://github.com/tamaranesterenko/TRO_LR_5