

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ**

**ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Визуализация данных с помощью matplotlib»**

**Отчет по лабораторной работе № 3.5**

**по дисциплине «Технологии распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1

Гасанов Г. М. « » 2023г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 2023г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2023

**Цель работы:** исследовать базовые возможности библиотеки matplotlib языка программирования Python **Ход работы:**

**Выполнение работы:**

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.
6. Проработать примеры лабораторной работы.

```
In [28]: import matplotlib.pyplot as plt
import numpy as np
import matplotlib.colors as mcolors
```

```
In [29]: 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, ms=12)

plt.legend()
plt.grid(True)
```

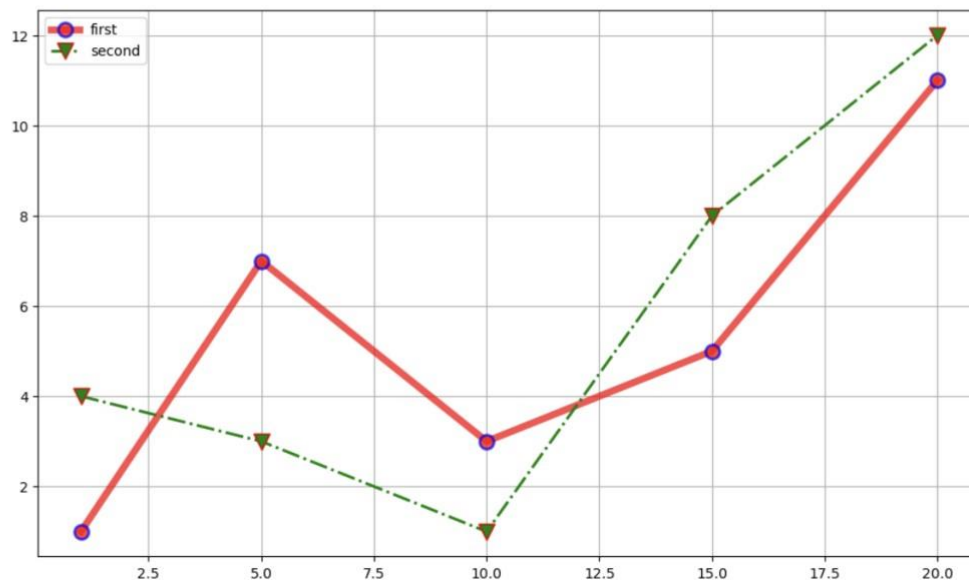
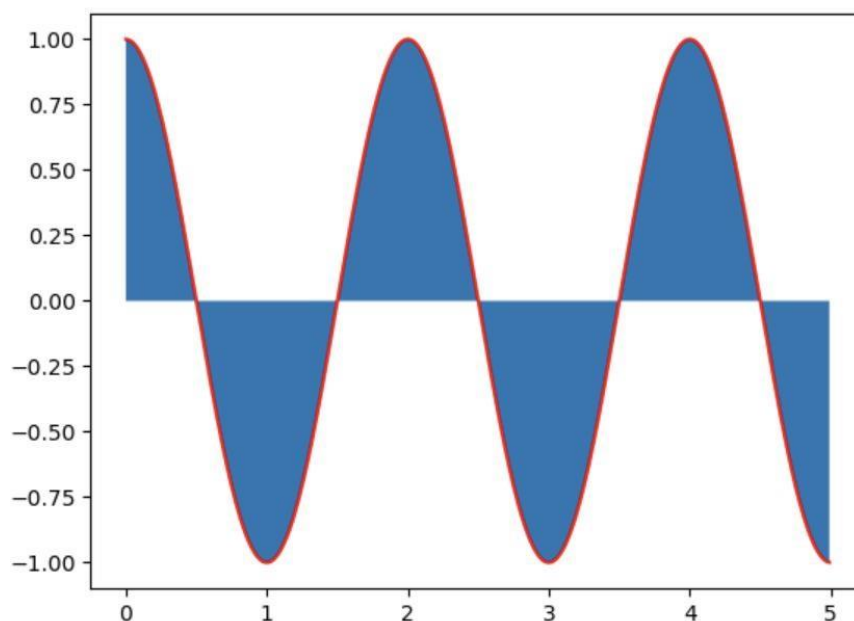


Рисунок 1 – Примеры лабораторной работы

```
In [30]: 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)
```

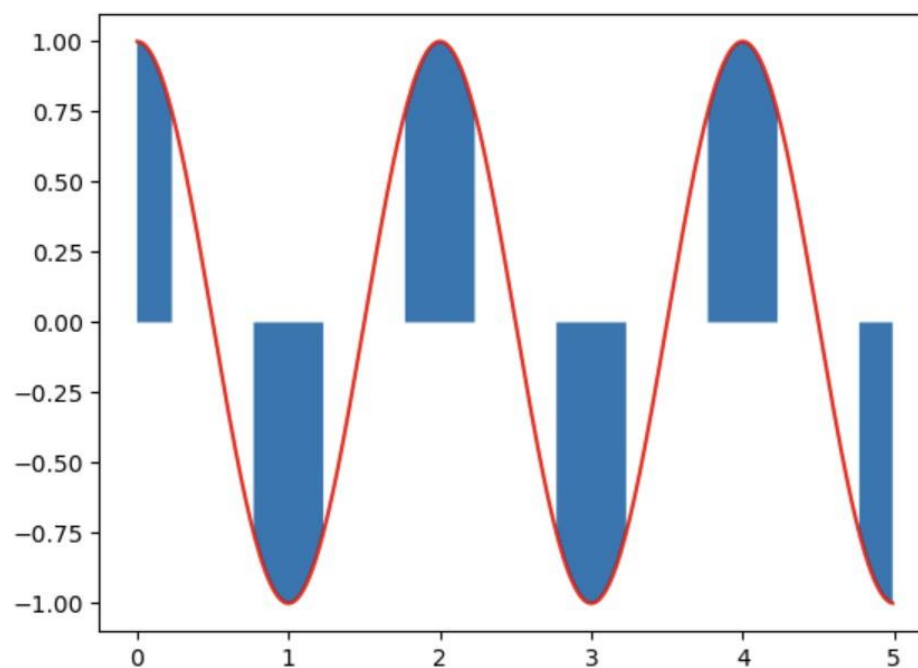
Out[30]: <matplotlib.collections.PolyCollection at 0x120bbb130>



## Рисунок 2 – Примеры лабораторной работы

```
In [31]: plt.plot(x, y, c='r')  
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))
```

Out[31]: <matplotlib.collections.PolyCollection at 0x120c50ca0>



```
In [32]: plt.plot(x, y, c="r")  
plt.fill_between(x, y, where=(y > 0))
```

Out[32]: <matplotlib.collections.PolyCollection at 0x120cc7190>

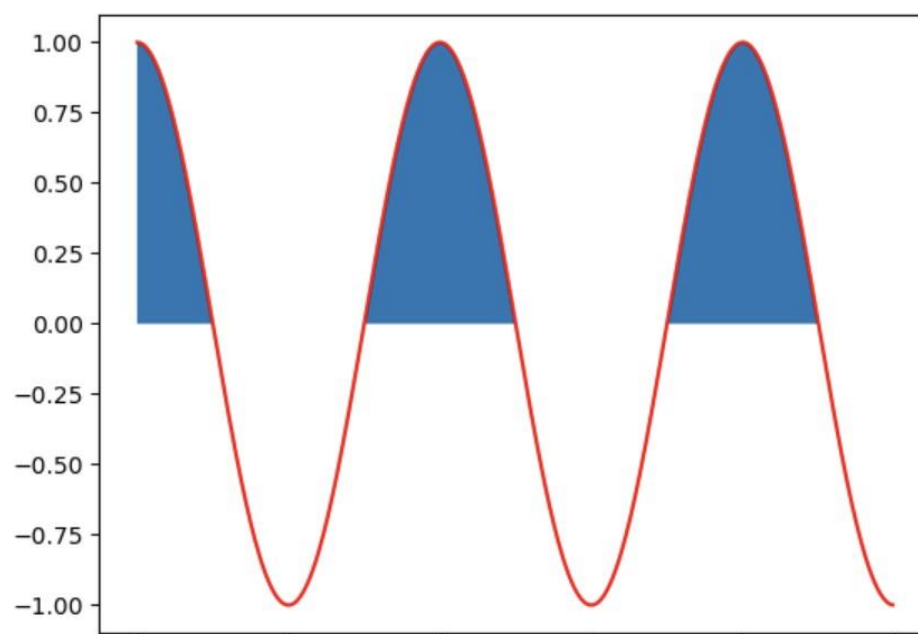
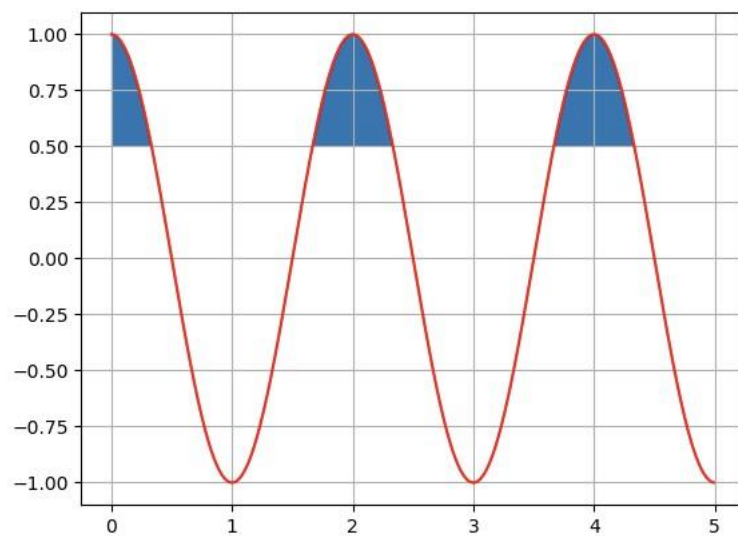


Рисунок 3 – Примеры лабораторной работы

```
In [33]: plt.plot(x, y, c="r")  
plt.grid()  
plt.fill_between(x, 0.5, y, where=(y>=0.5))
```

Out[33]: <matplotlib.collections.PolyCollection at 0x120d30910>



```
In [34]: plt.plot(x, y, c="r")  
plt.grid()  
plt.fill_between(x, y, 1)
```

Out[34]: <matplotlib.collections.PolyCollection at 0x120da7520>

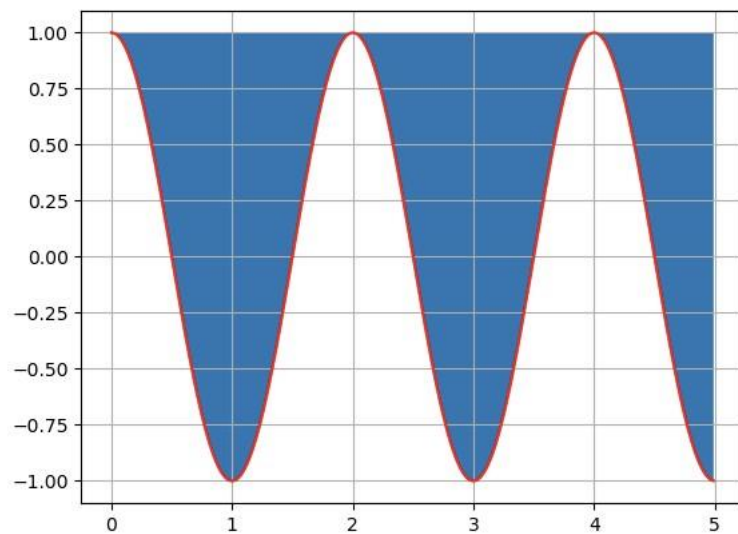
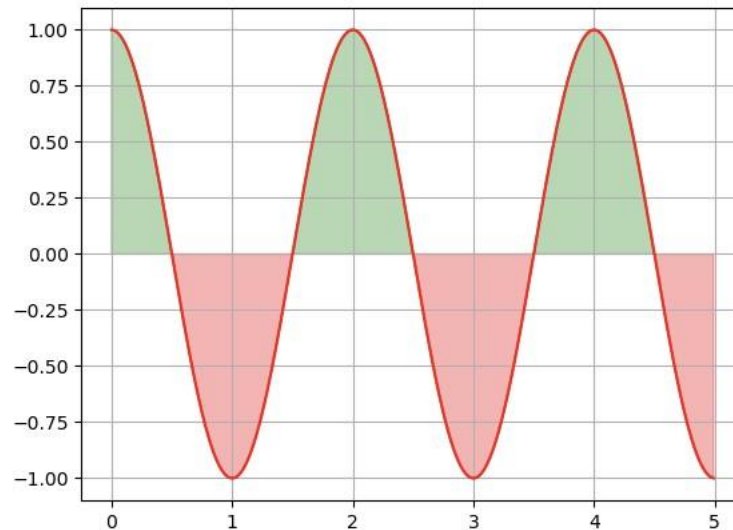


Рисунок 4 – Примеры лабораторной работы

```
In [35]: 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 [35]: <matplotlib.collections.PolyCollection at 0x120e19b20>



```
In [36]: 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 [36]: [<matplotlib.lines.Line2D at 0x120b97eb0>]

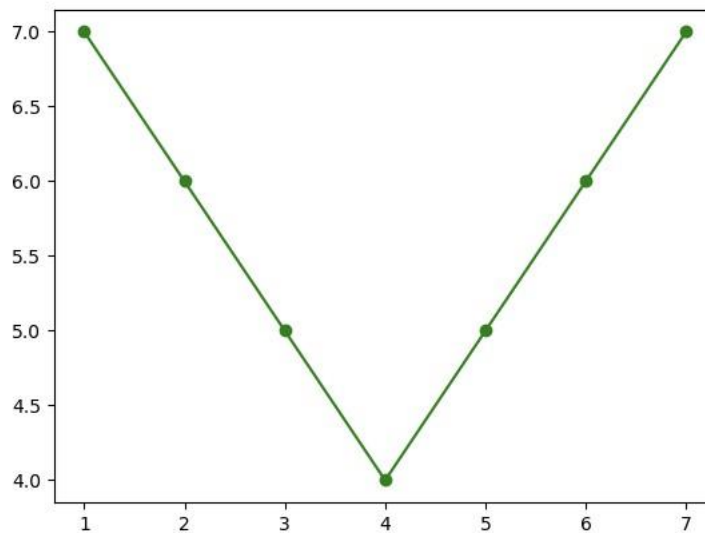
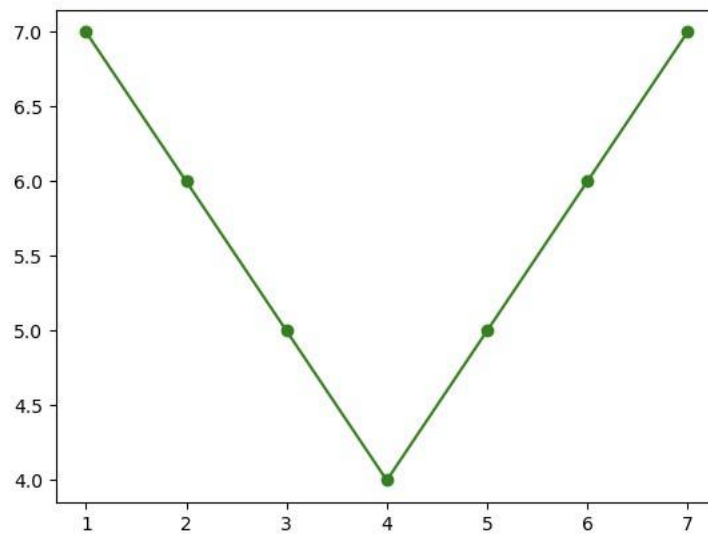


Рисунок 5 – Примеры лабораторной работы

```
In [36]: 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 [36]: [



```
In [37]: x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)

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

Out [37]: [

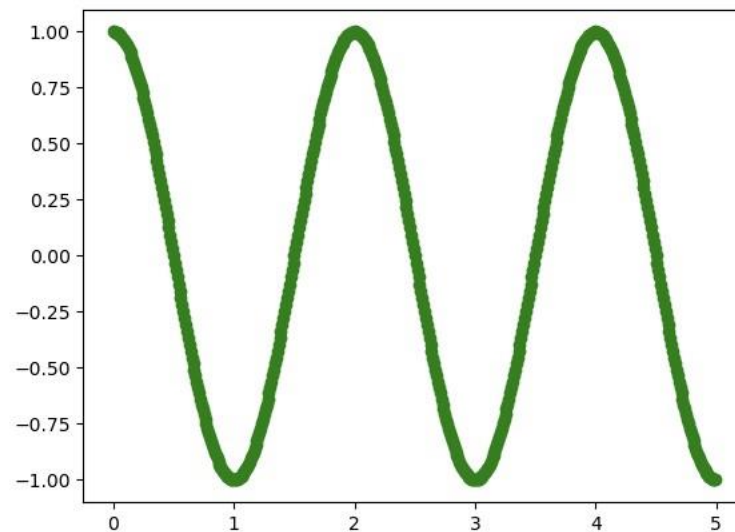


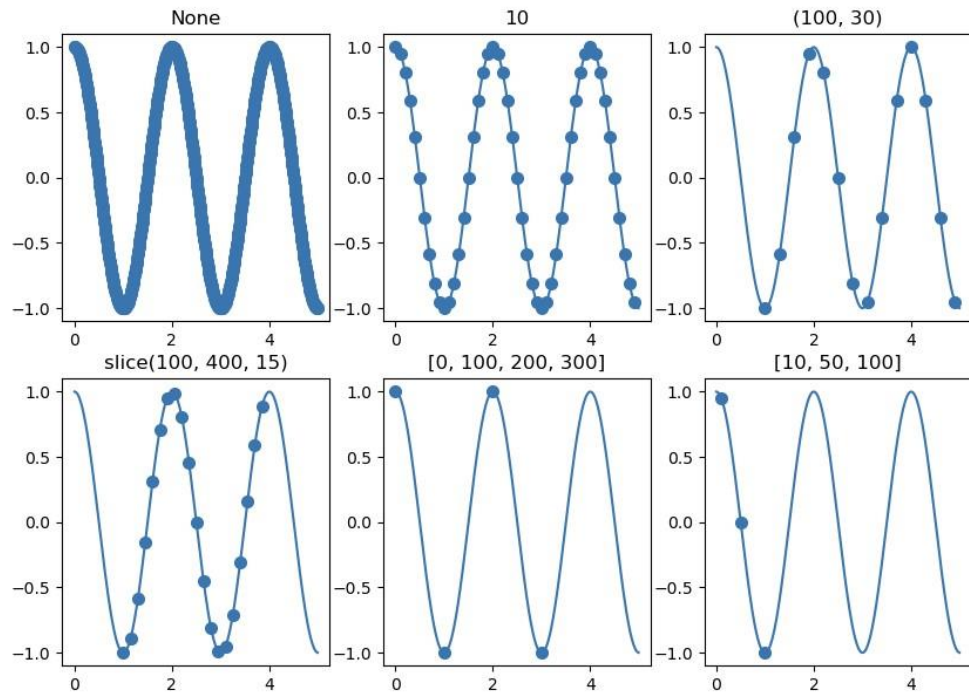
Рисунок 6 – Примеры лабораторной работы



```
In [38]: 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)
```



```
In [39]: 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 [39]: [<matplotlib.lines.Line2D at 0x120ec7130>]

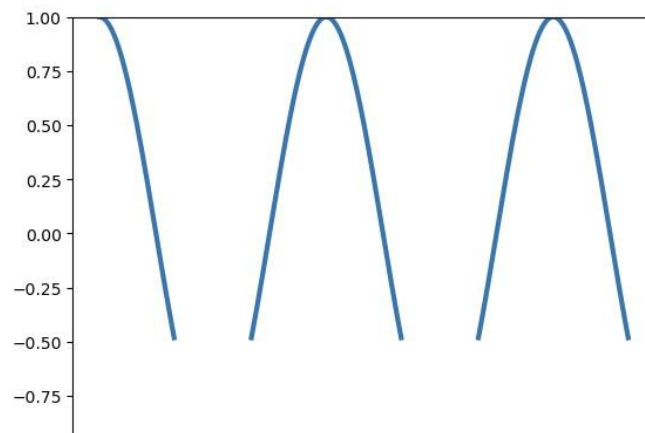
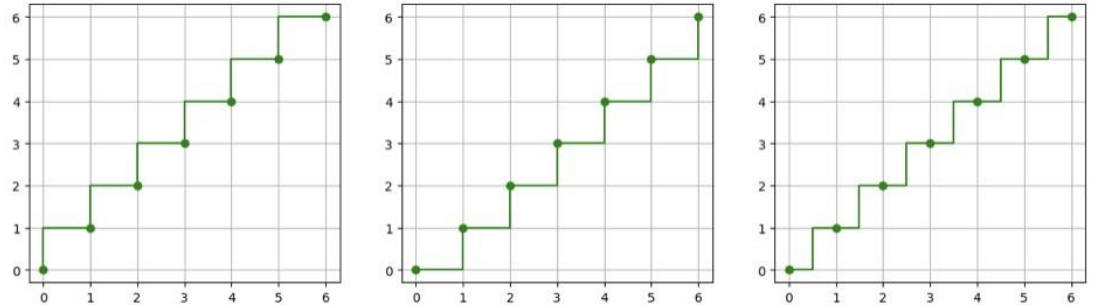


Рисунок 7 – Примеры лабораторной работы

```
In [40]: 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()
```



```
In [41]: 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[41]: <matplotlib.legend.Legend at 0x12181f490>

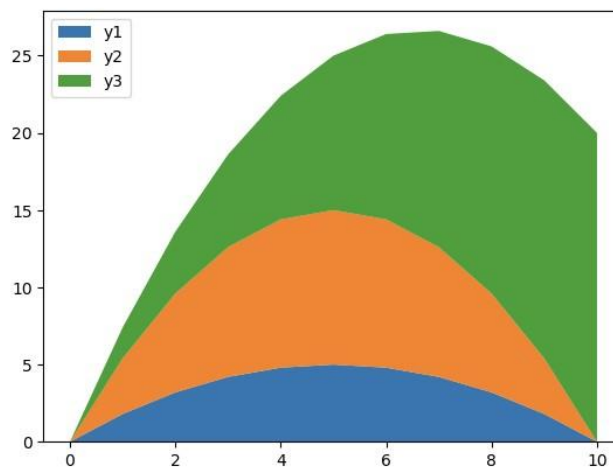
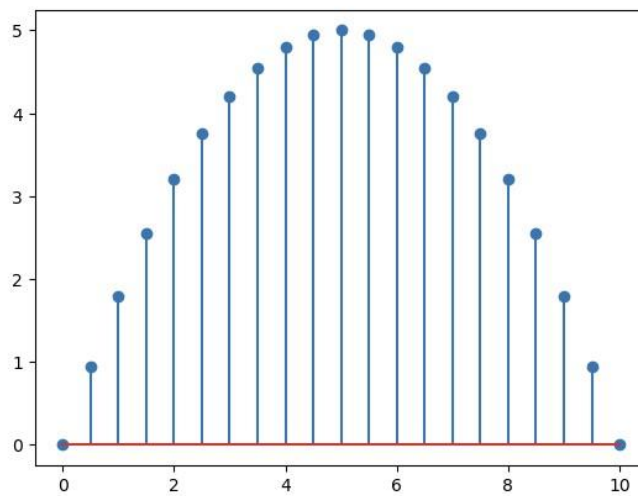


Рисунок 8 – Примеры лабораторной работы

```
In [42]: 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[42]: <StemContainer object of 3 artists>



```
In [43]: plt.stem(x, y, linefmt="r--", markerfmt="^", bottom=1)
```

Out[43]: <StemContainer object of 3 artists>

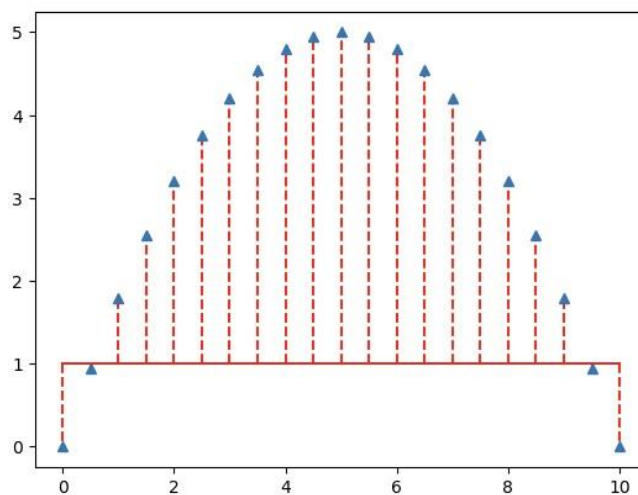
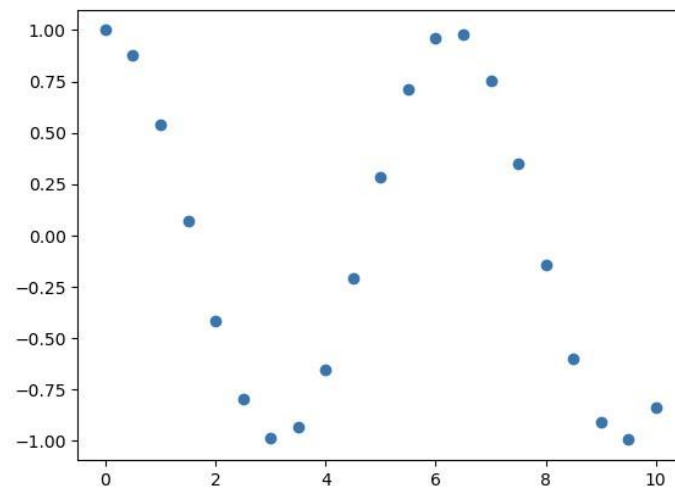


Рисунок 9 – Примеры лабораторной работы

```
In [44]: x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.scatter(x, y)
```

Out[44]: <matplotlib.collections.PathCollection at 0x1219c6f70>



```
In [45]: 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[45]: <matplotlib.collections.PathCollection at 0x121a3a340>

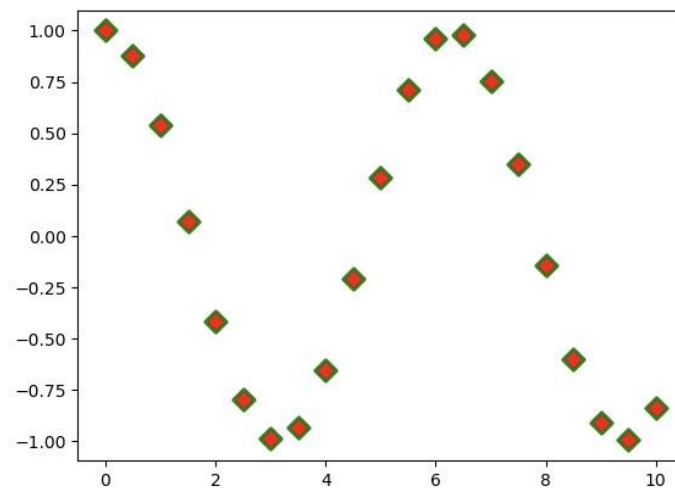


Рисунок 10 – Примеры лабораторной работы

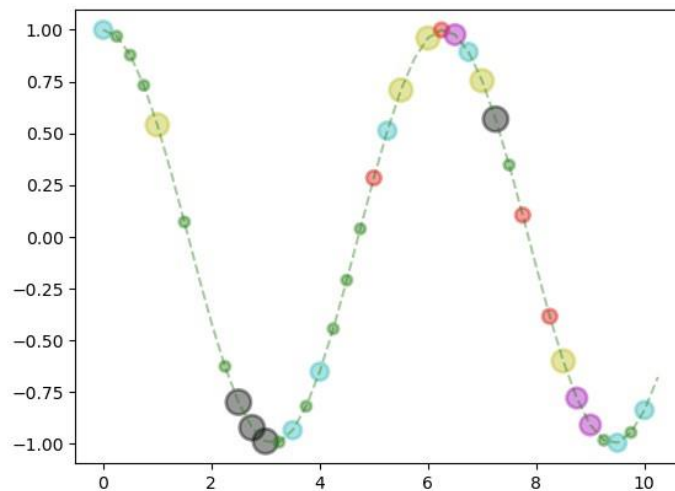
```
In [46]: bc = mcolors.BASE_COLORS
x = np.arange(0, 10.5, 0.25)
y = np.cos(x)

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[46]: [<matplotlib.lines.Line2D at 0x121ab3250>]



```
In [47]: 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[47]: <BarContainer object of 7 artists>

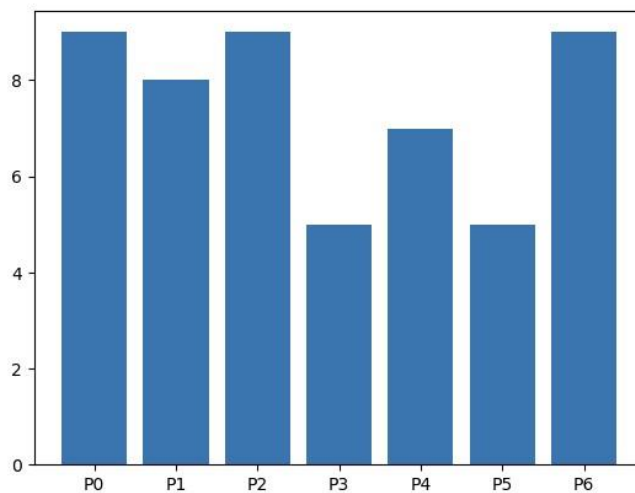
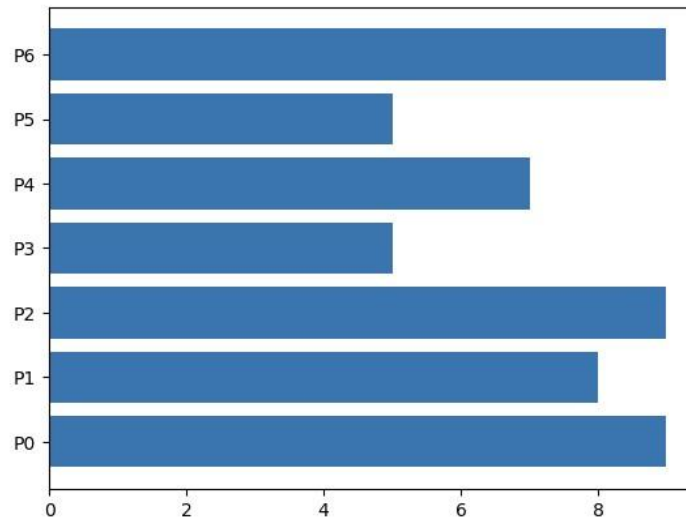


Рисунок 11 – Примеры лабораторной работы

```
In [48]: plt.barh(groups, counts)
```

```
Out[48]: <BarContainer object of 7 artists>
```



```
In [49]: 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[49]: <BarContainer object of 7 artists>
```

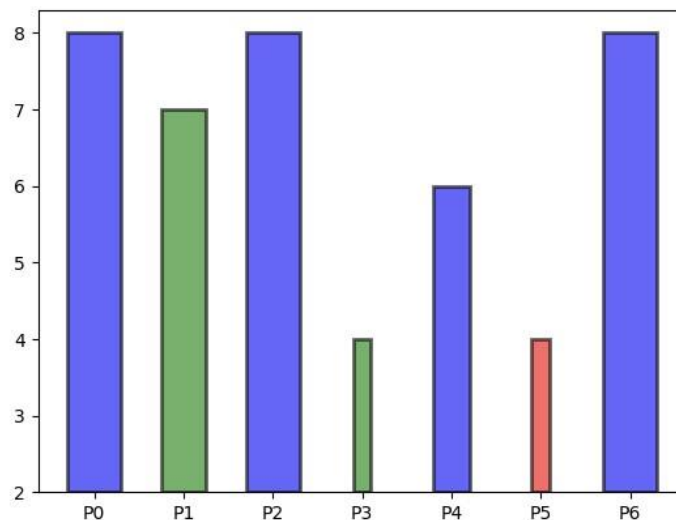


Рисунок 12 – Примеры лабораторной работы

```

In [50]: 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[50]: <matplotlib.legend.Legend at 0x1208c9700>

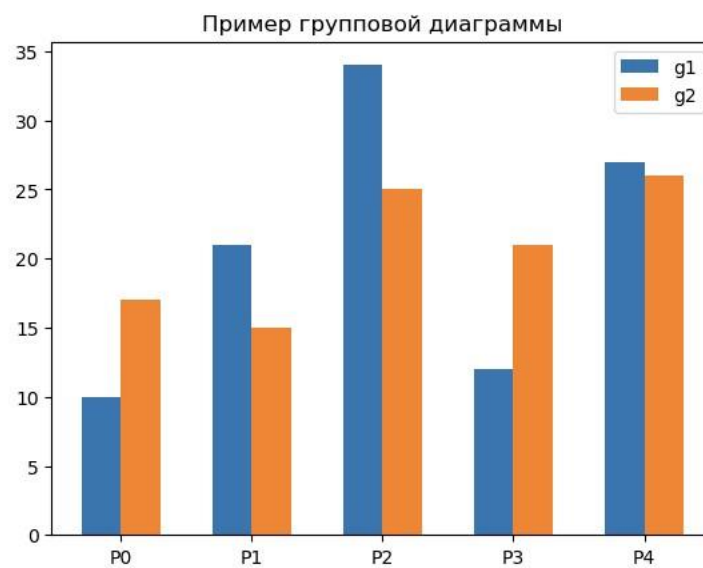


Рисунок 13 – Примеры лабораторной работы



```

In [51]: np.random.seed(123)

rnd = np.random.randint

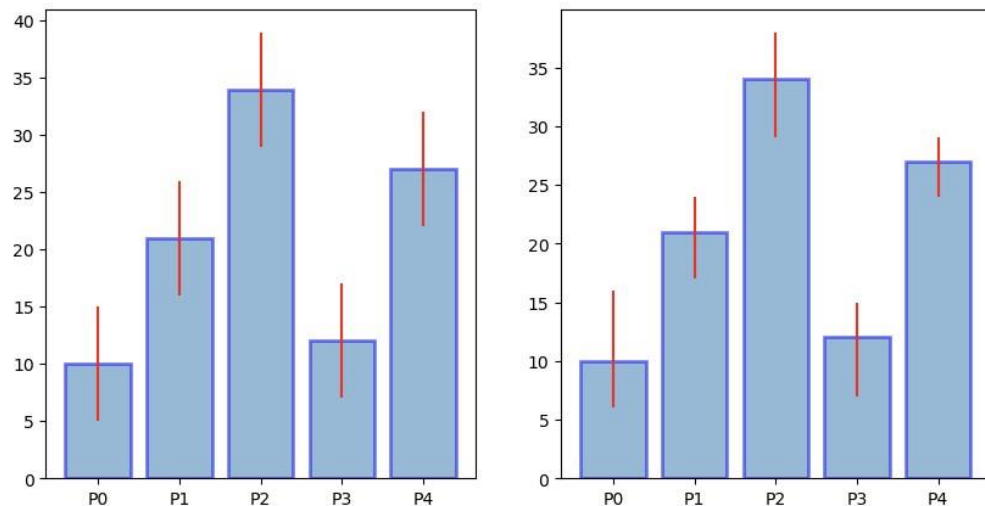
cat_par = [f"P{i}" for i in range(5)]
g1 = [10, 21, 34, 12, 27]

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[51]: <BarContainer object of 5 artists>



```

In [52]: 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[52]: (-1.1163226287452406,  
1.1007772680354877,  
-1.1107362350259515,  
1.1074836529113834)

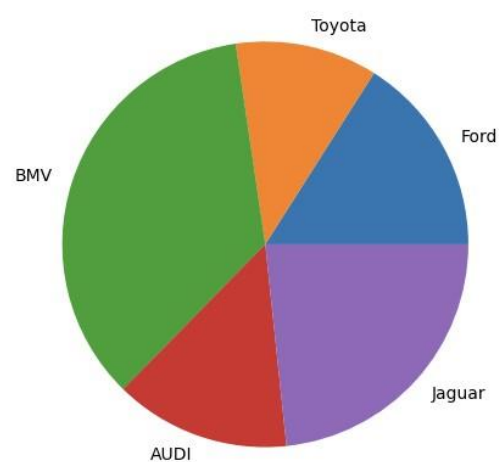


Рисунок 14 – Примеры лабораторной работы



```

In [54]: 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[54]: ([<matplotlib.patches.Wedge at 0x1205f0910>,
<matplotlib.patches.Wedge at 0x1205f0df0>,
<matplotlib.patches.Wedge at 0x1205e2310>,
<matplotlib.patches.Wedge at 0x1205e27f0>,
<matplotlib.patches.Wedge at 0x1205e2cd0>,
<matplotlib.patches.Wedge at 0x1205e91f0>,
<matplotlib.patches.Wedge at 0x1205e96d0>,
<matplotlib.patches.Wedge at 0x1205e9bb0>,
<matplotlib.patches.Wedge at 0x120812910>],
[Text(0.646314344414094, 0.13370777166859046, ''),
Text(0.4521935266177387, 0.4807504700829865, ''),
Text(0.040366679721656945, 0.6587643973138266, ''),
Text(-0.3454228878740909, 0.5623904591409097, ''),
Text(-0.6578039053946477, 0.05379611554331286, ''),
Text(-0.48987451889717687, -0.44229283934431896, ''),
Text(-0.12049606360635531, -0.6489073112975174, ''),
Text(0.39011356818311405, -0.532363976917521, ''),
Text(0.6332653697075483, -0.1859434632601054, '')]

```



Рисунок 15 – Примеры лабораторной работы

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

labels = ["Ford", "Toyota", "BMV", "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[53]: (-1.2704955621219602,
1.1999223938155328,
-1.1121847055183558,
1.1379015332518725)
```

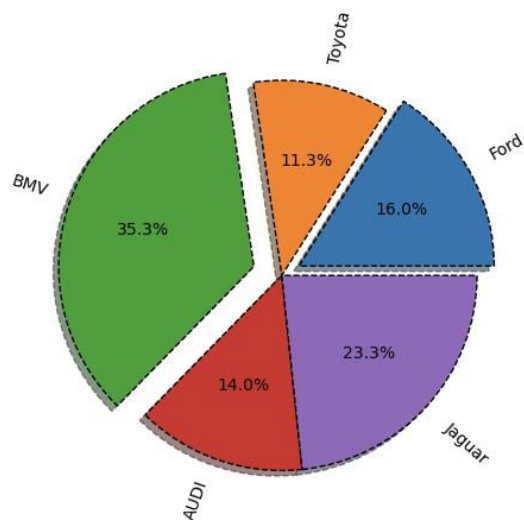
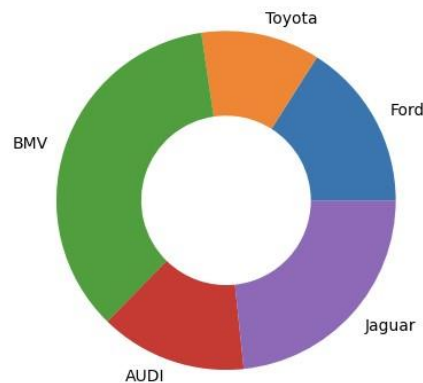


Рисунок 16 – Примеры лабораторной работы

```
In [55]: vals = [24, 17, 53, 21, 35]
labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]
fig, ax = plt.subplots()
ax.pie(vals, labels=labels, wedgeprops=dict(width=0.5))

Out[55]: ([<matplotlib.patches.Wedge at 0x1207eed30>,
<matplotlib.patches.Wedge at 0x1207f5250>,
<matplotlib.patches.Wedge at 0x1207f5730>,
<matplotlib.patches.Wedge at 0x1207f5bb0>,
<matplotlib.patches.Wedge at 0x1207ecd00>],
[Text(0.9639373540021144, 0.5299290306818475, 'Ford'),
Text(0.22870287165240302, 1.075962358309037, 'Toyota'),
Text(-1.046162158377023, 0.3399187231970734, 'BMW'),
Text(-0.3617533684721028, -1.0388139873909512, 'AUDI'),
Text(0.8174592712713288, -0.7360437078139777, 'Jaguar')])
```



```
In [56]: 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[56]: <matplotlib.image.AxesImage at 0x1220b1c40>
```

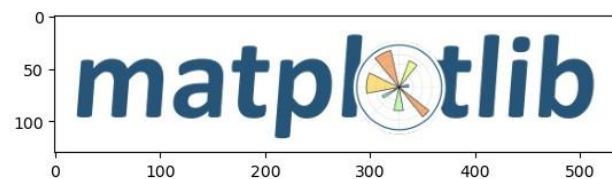
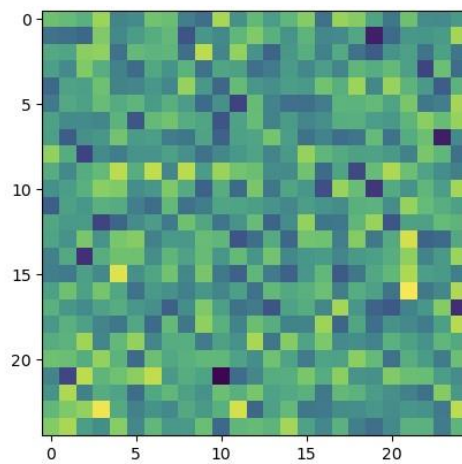


Рисунок 17 – Примеры лабораторной работы

```
In [57]: np.random.seed(19680801)
```

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

```
Out[57]: <matplotlib.image.AxesImage at 0x12211a700>
```



```
In [58]: fig, axes = plt.subplots(1, 2, figsize=(10,3), constrained_layout=True)  
  
p1 = axes[0].imshow(data, cmap='winter', aspect='equal', vmin=-1, vmax=1,  
origin="lower")  
fig.colorbar(p1, ax=axes[0])  
  
p2 = axes[1].imshow(data, cmap='plasma', aspect='equal',  
interpolation='gaussian', origin="lower", extent=(0, 30, 0, 30))  
fig.colorbar(p2, ax=axes[1])
```

```
Out[58]: <matplotlib.colorbar.Colorbar at 0x120642be0>
```

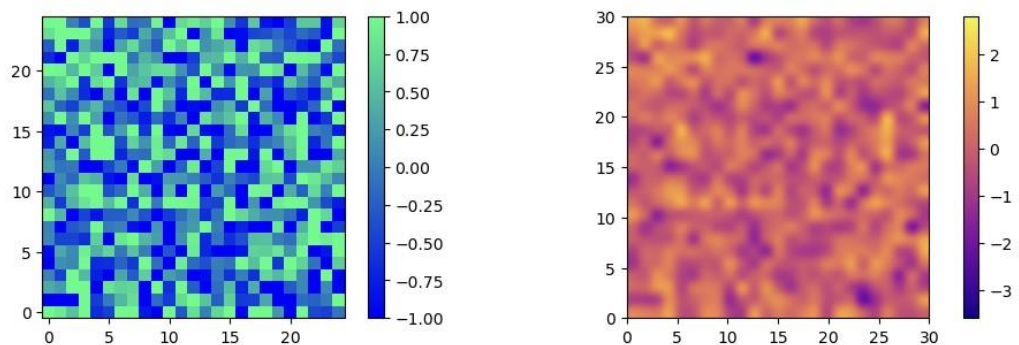


Рисунок 18 – Примеры лабораторной работы

```
In [59]: np.random.seed(123)
data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')

Out[59]: <matplotlib.collections.QuadMesh at 0x12230c4c0>
```

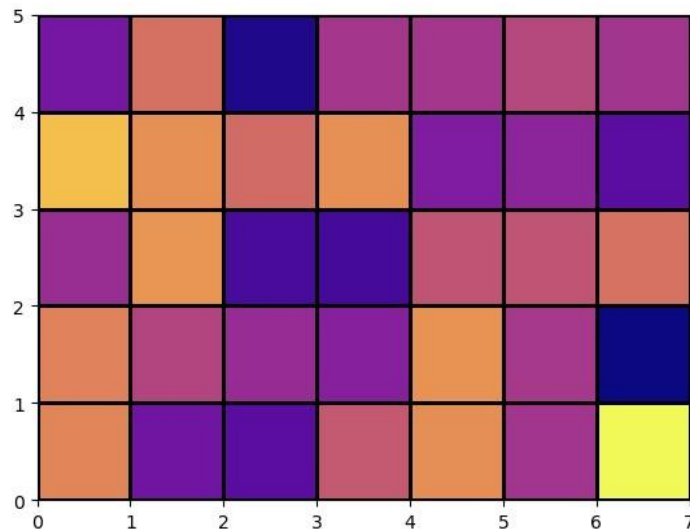


Рисунок 19 – Примеры лабораторной работы

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения линейного графика, условие которой предварительно необходимо согласовать с преподавателем.

**Условие задачи:**

У вас есть шарик массой 0,1 кг, который находится на высоте 2 метра над землей. Вы отпускаете шарик, и он начинает свободно падать под действием силы тяжести. Используя законы движения свободного падения, постройте график зависимости скорости шарика от времени за первые 4 секунды его падения. Помните, что скорость свободно падающего тела увеличивается со временем с постоянным ускорением, равным ускорению свободного падения, приблизительно равному  $9,8 \text{ м/с}^2$ .

**Решение:**

Для решения этой задачи мы можем использовать закон движения свободного падения, который гласит:

$$h = \frac{gt^2}{2}$$

где  $h$  - высота, с которой падает тело,  $g$  - ускорение свободного падения (приблизительно  $9,8 \text{ м/с}^2$ ),  $t$  - время падения.

Найдем скорость падающего тела:

$$v = gt$$

где  $v$  - скорость,  $g$  - ускорение свободного падения,  $t$  - время падения.

```
In [18]: import matplotlib.pyplot as plt
```

```
In [24]: h = 2
m = 0.1
g = 9.8
t_max = 4
```

```
In [25]: time = []
velocity = []
```

```
In [26]: for t in range(0, t_max+1):
        time.append(t)
        v = g*t
        velocity.append(v)
```

Построим график зависимости скорости шарика от времени за первые 4 секунды его падения: На оси X графика будет откладываться время в секундах, а на оси Y - скорость в метрах в секунду.

```
In [27]: # Строим график зависимости скорости от времени
plt.plot(time, velocity)
plt.title('Зависимость скорости шарика от времени')
plt.xlabel('Время, с')
plt.ylabel('Скорость, м/с')
plt.grid()
plt.show()
```

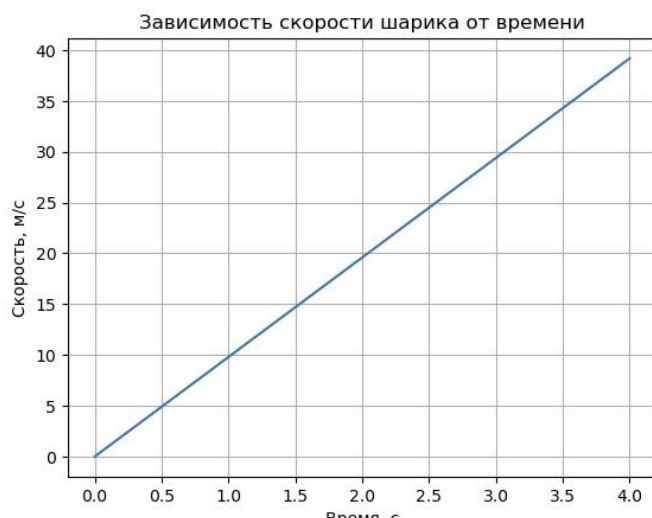


Рисунок 20 – Выполнение задания



8. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

**Условие задачи:**

Компания производит и продает три вида продукции: книги, музыкальные инструменты и спортивное оборудование. Проанализируйте, какой процент общего дохода компании приходится на каждый вид продукции, чтобы определить, на какую продукцию следует сосредоточить больше внимания в бизнес-стратегии.

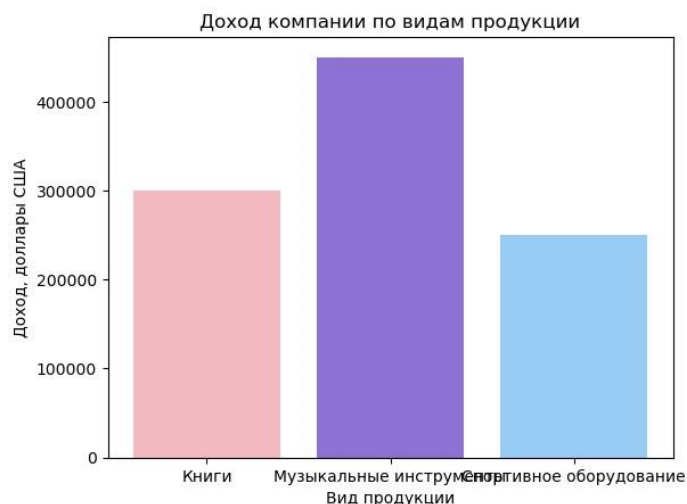
Общий доход компании за прошлый год составил 1 000 000 долларов США. Ниже приведена таблица, показывающая, какой доход был получен от каждого вида продукции:

Вид продукции:	Доход, доллары США
Книги	300 000
Музыкальные инструменты	450 000
Спортивное оборудование	250 000

```
In [3]: import matplotlib.pyplot as plt
```

```
In [4]: labels = ['Книги', 'Музыкальные инструменты', 'Спортивное оборудование']  
values = [300000, 450000, 250000]
```

```
In [6]: # Строим столбчатую диаграмму  
plt.bar(labels, values, color=['lightpink', 'MediumPurple', 'lightskyblue'])  
  
plt.title('Доход компании по видам продукции')  
plt.xlabel('Вид продукции')  
plt.ylabel('Доход, доллары США')  
  
plt.show()
```



Как видно из диаграммы, музыкальные инструменты приносят наибольший доход компании, тогда как спортивное оборудование приносит наименьший доход. Эта информация может помочь компании сосредоточить внимание на наиболее прибыльных продуктах при разработке бизнес-стратегии.

Рисунок 21 – Выполнение задания

9. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения круговой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.



## Самостоятельное задание №9.

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения круговой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

### Условие:

Компания производит и продает три вида продукции: книги, музыкальные инструменты и спортивное оборудование. Проанализируйте, какой процент общего дохода компании приходится на каждый вид продукции, чтобы определить, на какую продукцию следует сосредоточить больше внимания в бизнес-стратегии.

Общий доход компании за прошлый год составил 1 000 000 долларов США. Ниже приведена таблица, показывающая, какой доход был получен от каждого вида продукции:

Вид продукции:	Доход, доллары США
Книги	300 000
Музыкальные инструменты	450 000
Спортивное оборудование	250 000

```
In [4]: import matplotlib.pyplot as plt
```

```
In [7]: labels = ['Книги', 'Музыкальные инструменты', 'Спортивное оборудование']  
        sizes = [300000, 450000, 250000]
```

```
In [8]: colors = ['lightpink', 'MediumPurple', 'lightskyblue']  
  
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)  
plt.title('Доход компании по видам продукции')  
plt.axis('equal')  
plt.show()
```



Данная круговая диаграмма показывает какой процент общего дохода компании приходится на каждый вид продукции. Как видно из диаграммы, большую часть дохода компании (45%) приносят музыкальные инструменты, в то время как книги приносят лишь 30% от общего дохода.

Рисунок 22 – Выполнение задания

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

## Самостоятельное задание №10.

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

```
In [6]: import matplotlib.pyplot as plt  
%matplotlib inline
```

```
In [7]: from PIL import Image  
import requests
```

```
In [8]: from io import BytesIO
```

```
In [11]: plt.figure(figsize=(20, 20))  
  
url = requests.get('https://sun9-3.userapi.com/impgr/xHFuxvCLV2BHqaadBJxZfxWkEfSBdV6o_tB  
img = Image.open(BytesIO(url.content))  
  
plt.imshow(img)
```

```
Out[11]: <matplotlib.image.AxesImage at 0x1423a7790>
```

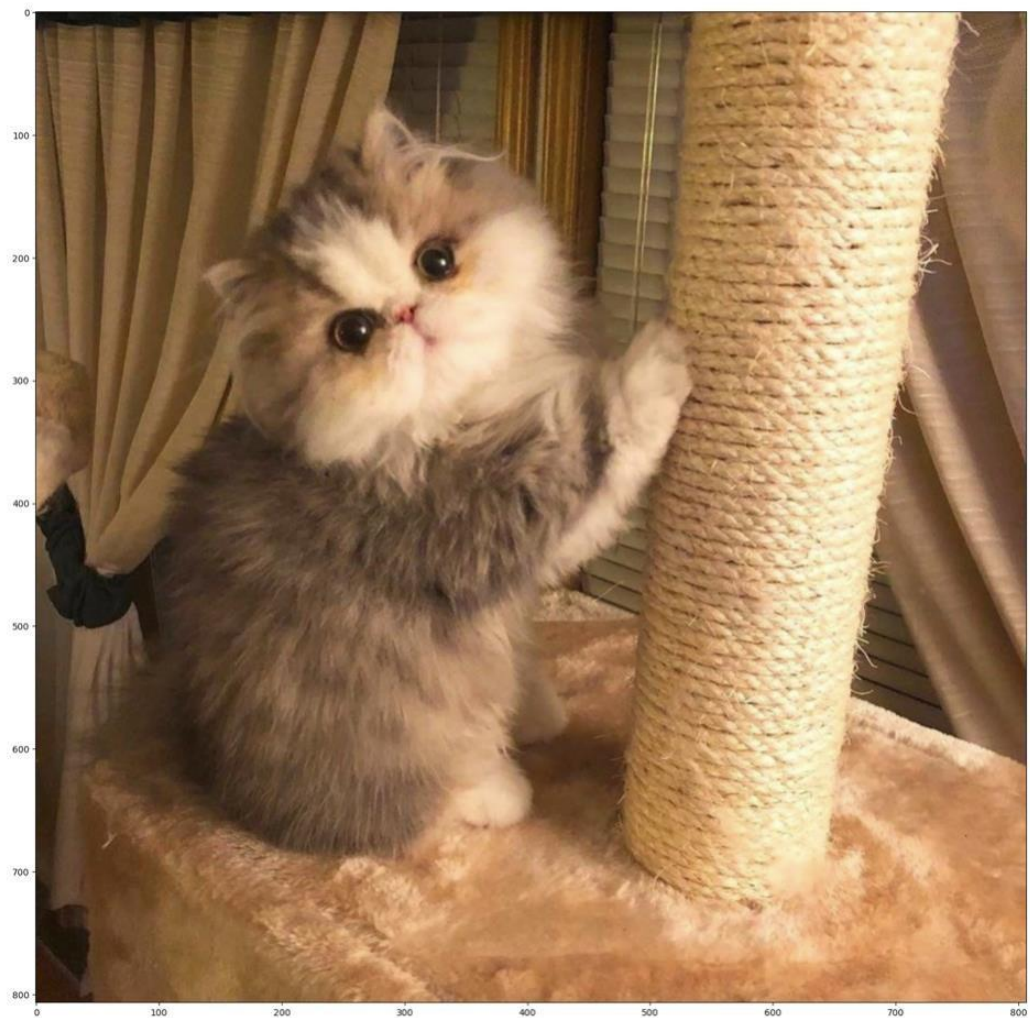


Рисунок 23 – Выполнение задания

11. Зафиксируйте сделанные изменения в репозитории.

12. Выполните слияние ветки для разработки с веткой main (master).
13. Отправьте сделанные изменения на сервер GitHub.

### Ответы на контрольные вопросы:

1. Как выполнить построение линейного графика с

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

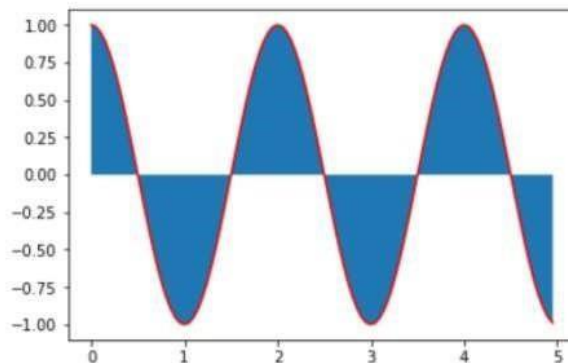
```
plot([x], y, [fmt], *, data=None, **kwargs)
plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)
```

помощью matplotlib?

2. Как выполнить заливку области между графиком и осью?

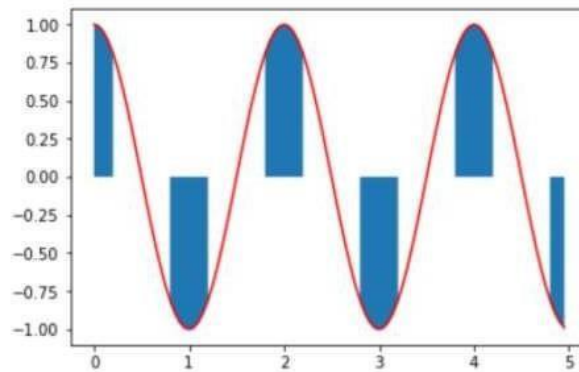
Между двумя графиками?

```
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))
```

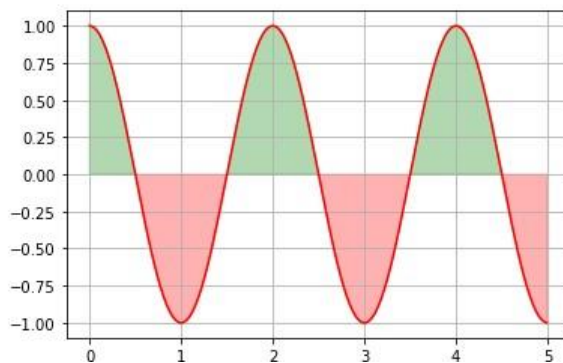


#### 4. Как выполнить двухцветную заливку?

```
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)
```

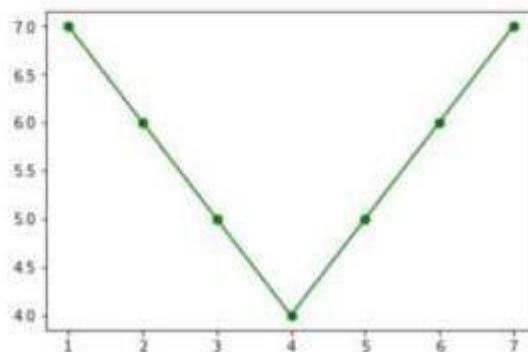
Out[14]: <matplotlib.collections.PolyCollection at 0x1b7583e04f0>



#### 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")
```

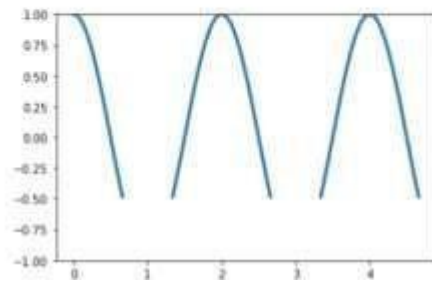


#### 6. Как выполнить обрезку графиков?

```
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)
```

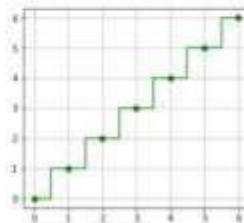
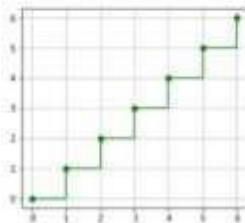
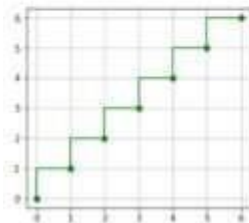


7. Как построить ступенчатый график? В чем особенность ступенчатого графика?

```
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.

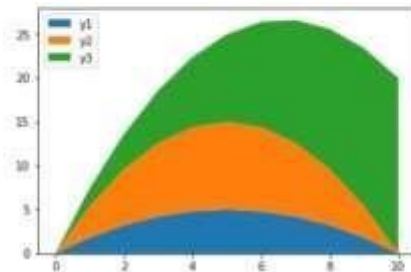
```
x = np.arange(0, 11, 1)

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')
```

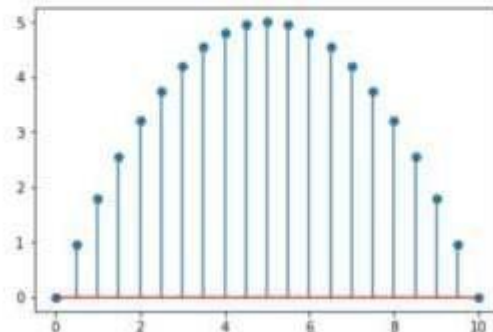


### 9. Как построить stem-график? В чем особенность stem-графика?

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

plt.stem(x, y)
```



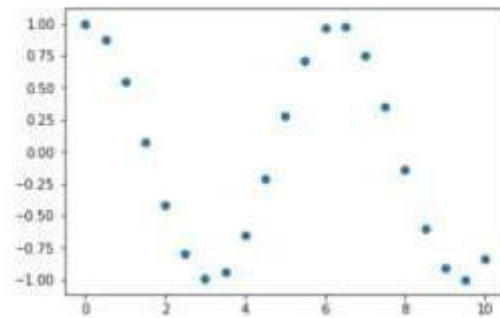
### 10. Как построить точечный график? В чем особенность точечного графика?

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)
```

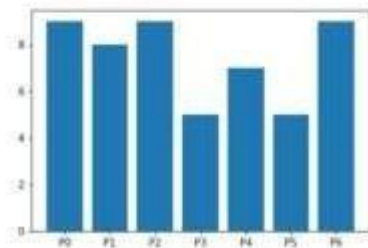


11. Как осуществляется построение столбчатых диаграмм с

```
np.random.seed(123)

groups = ['P1' for i in range(7)]
counts = np.random.randint(3, 10, len(groups))

plt.bar(groups, counts)
```



помощью matplotlib?

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]

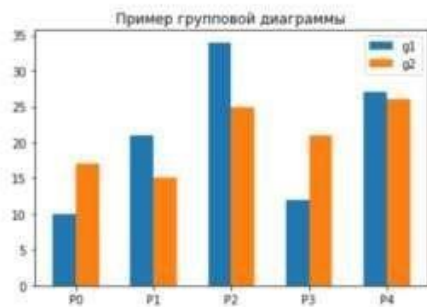
width = 0.3

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 хегг, уегг и еcolor (для задания цвета).

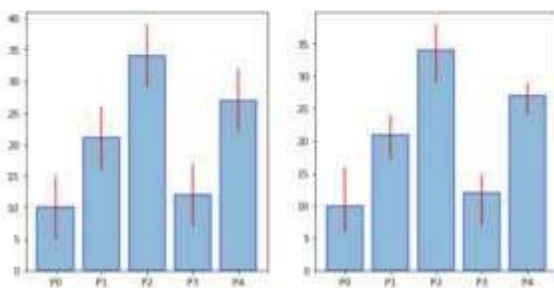
```
np.random.seed(123)

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)
```

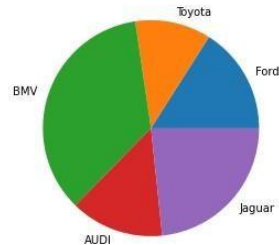


13. Как выполнить построение круговой диаграммы средствами matplotlib?



```
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))

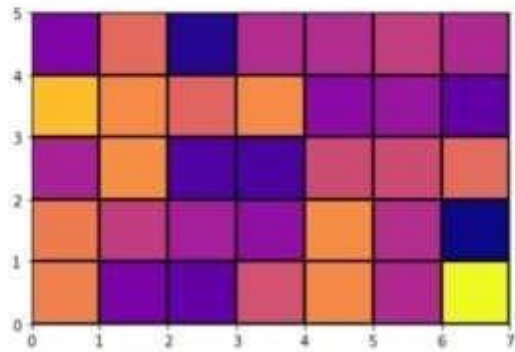
plt.imshow(img)
```

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')
```