

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

**Отчет по лабораторной работе №5 по дисциплине:
Технологии распознавания образов**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

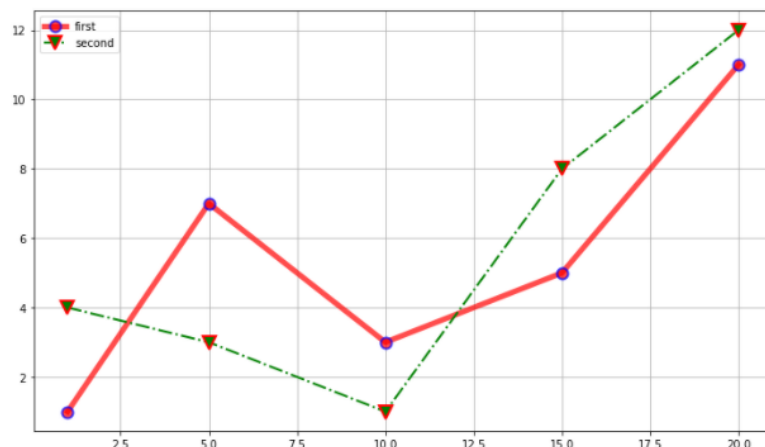
Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ:

1. Возможности команды plot()

```
In [1]: import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: 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)
```

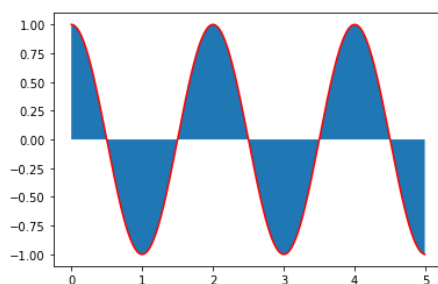


2. Выполнение заливки области между графиком и осью

```
In [4]: import numpy as np

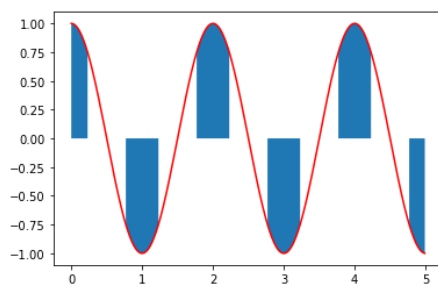
x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)
plt.plot(x, y, c="r")
plt.fill_between(x, y)
```

Out[4]: <matplotlib.collections.PolyCollection at 0x2278813ed30>



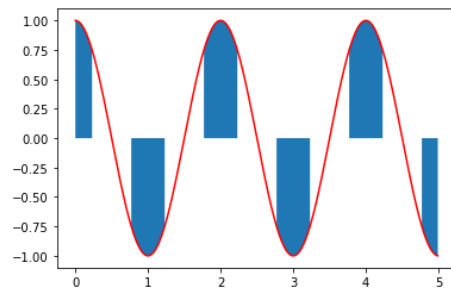
```
In [5]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))
```

Out[5]: <matplotlib.collections.PolyCollection at 0x227881b1c40>



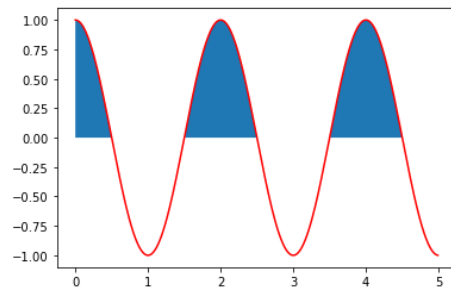
```
In [5]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))
```

Out[5]: <matplotlib.collections.PolyCollection at 0x227881b1c40>



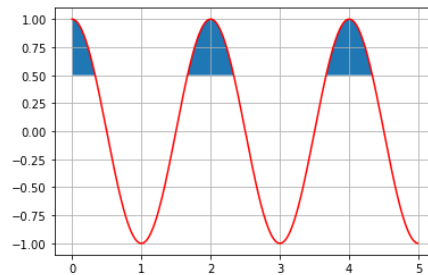
```
In [6]: plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0))
```

Out[6]: <matplotlib.collections.PolyCollection at 0x22788214850>



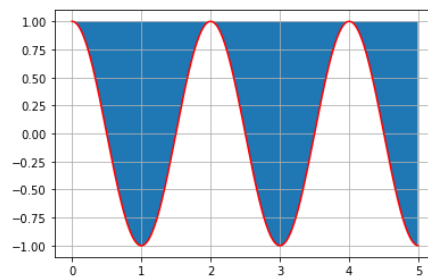
```
In [7]: plt.plot(x, y, c="r")
plt.grid()
plt.fill_between(x, 0.5, y, where=(y >= 0.5))
```

Out[7]: <matplotlib.collections.PolyCollection at 0x22788223580>



```
In [8]: plt.plot(x, y, c="r")
plt.grid()
plt.fill_between(x, y, 1)
```

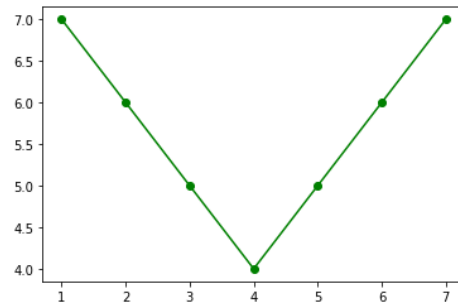
Out[8]: <matplotlib.collections.PolyCollection at 0x22788454220>



3. Настройка маркировки графиков

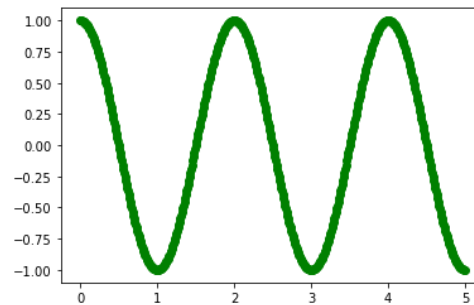
```
In [10]: x = [1, 2, 3, 4, 5, 6, 7]
y = [7, 6, 5, 4, 5, 6, 7]
plt.plot(x, y, marker="o", c="g")
```

```
Out[10]: [<matplotlib.lines.Line2D at 0x227885262b0>]
```

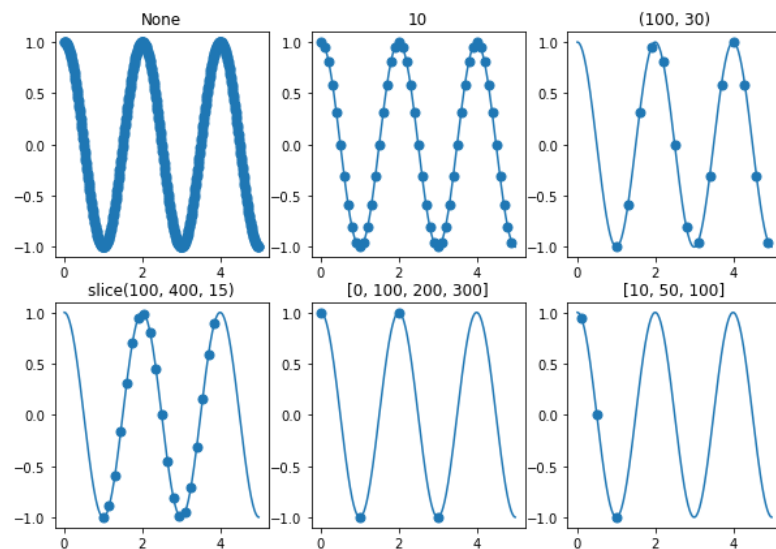


```
In [11]: import numpy as np
x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)
plt.plot(x, y, marker="o", c="g")
```

```
Out[11]: [<matplotlib.lines.Line2D at 0x227885f0640>]
```



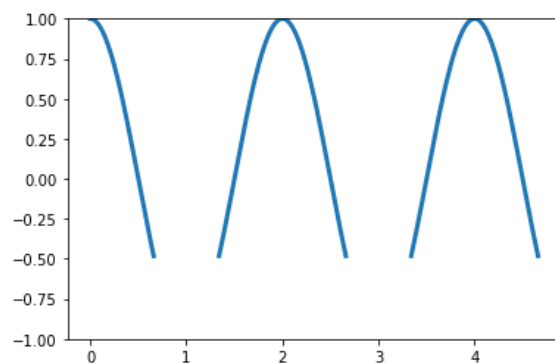
```
In [14]: x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)
m_ev_case = [None, 10, (100, 30), slice(100,400,15), [0, 100, 200, 300], [10, 50, 100]]
fig, ax = plt.subplots(2, 3, figsize=(10, 7))
ax = [ax[i, j] for i in range(2) for j in range(3)]
for i, case in enumerate(m_ev_case):
    ax[i].set_title(str(case))
    ax[i].plot(x, y, "o", ls='-', ms=7, markevery=case)
```



4. Обрезка графика

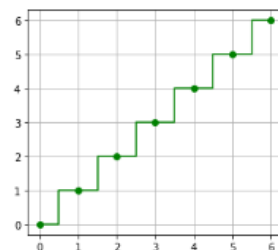
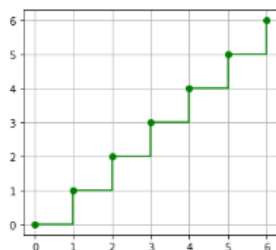
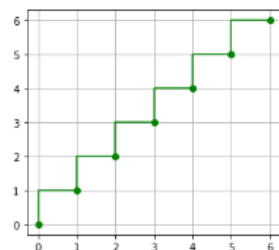
```
In [15]: x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)
y_masked = np.ma.masked_where(y < -0.5, y)
plt.ylim(-1, 1)
plt.plot(x, y_masked, linewidth=3)
```

```
Out[15]: [<matplotlib.lines.Line2D at 0x227882a61c0>]
```



5. Ступенчатый график

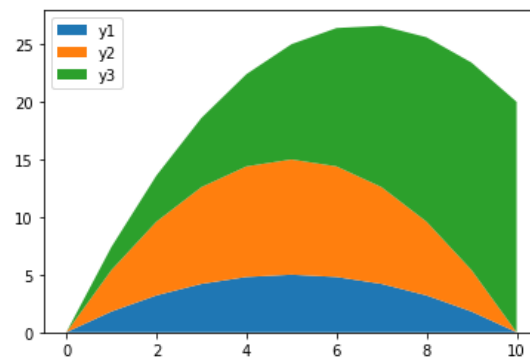
```
In [19]: x = np.arange(0, 7)
y = x
where_set = ['pre', 'post', 'mid']
fig, axs = plt.subplots(1, 3, figsize=(15, 4))
for i, ax in enumerate(axs):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()
```



6. Стековый график

```
In [20]: 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')
```

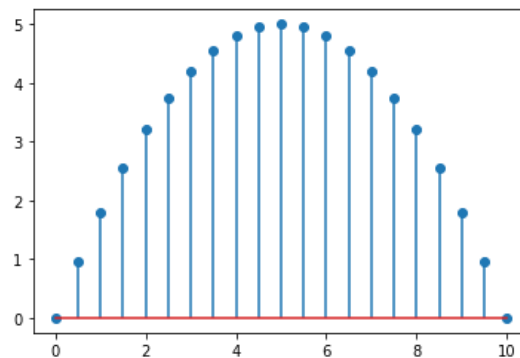
Out[20]: <matplotlib.legend.Legend at 0x22788177a00>



7. Stem-график

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

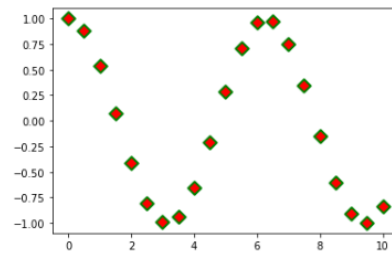
Out[21]: <StemContainer object of 3 artists>



8. Точечный график

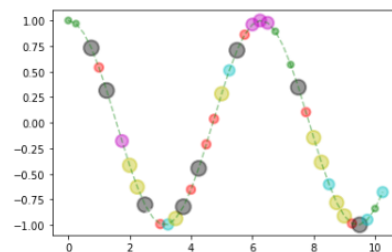
```
In [23]: x = np.arange(0, 10.5, 0.5)
y = np.cos(x)
plt.scatter(x, y, s=80, c="r", marker="D", linewidths=2, edgecolors="g")
```

Out[23]: <matplotlib.collections.PathCollection at 0x22788f21c0>



```
In [24]: 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)
```

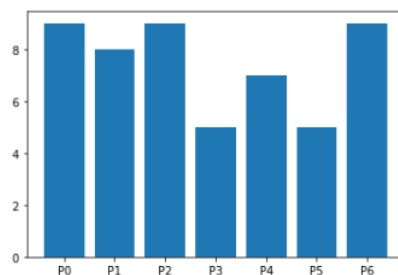
Out[24]: <matplotlib.lines.Line2D at 0x2278995f7c0>



9. Столбчатые диаграммы

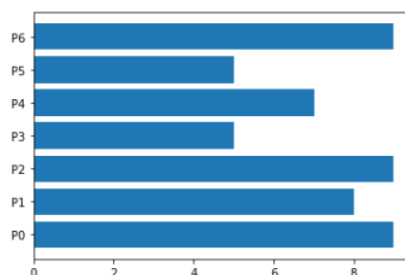
```
In [25]: np.random.seed(123)
groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(3, 10, len(groups))
plt.bar(groups, counts)
```

Out[25]: <BarContainer object of 7 artists>



```
In [26]: plt.barh(groups, counts)
```

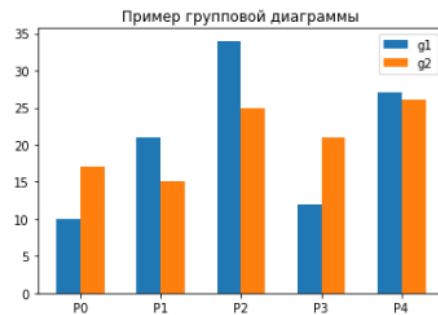
Out[26]: <BarContainer object of 7 artists>



10. Групповые столбчатые диаграммы

```
In [28]: 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()
```

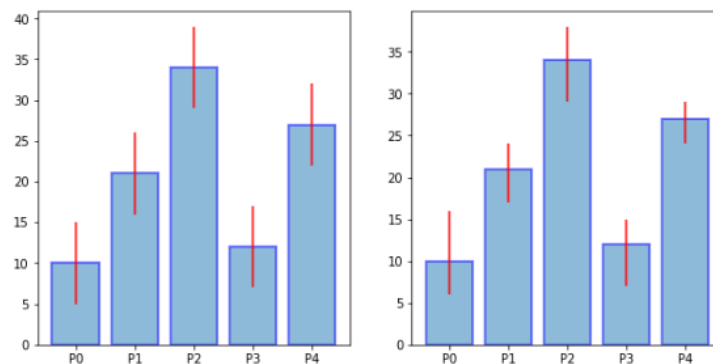
Out[28]: <matplotlib.legend.Legend at 0x227888c51f0>



11. Диаграмма с errorbar элементом

```
In [29]: 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)
```

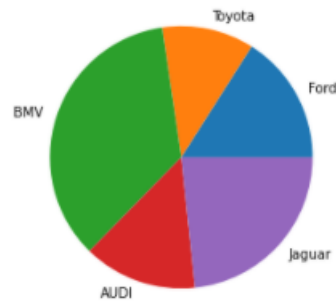
Out[29]: <BarContainer object of 5 artists>



12.Круговые диаграммы

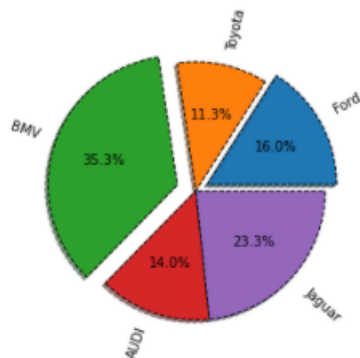
```
In [30]: vals = [24, 17, 53, 21, 35]
labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]
fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis("equal")
```

```
Out[30]: (-1.1163226287452406,
1.1007772680354877,
-1.1107362350259515,
1.1074836529113834)
```



```
In [31]: 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[31]: (-1.2704955621219602,
1.1999223938155328,
-1.1121847055183558,
1.1379015332518725)
```



13. Вложенные круговые диаграммы

```
In [32]: 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'))

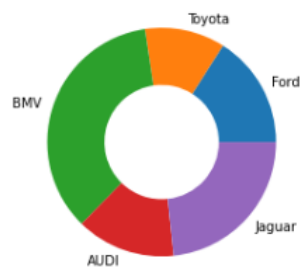
Out[32]: ([<matplotlib.patches.Wedge at 0x227886c8130>,
<matplotlib.patches.Wedge at 0x227886c8100>,
<matplotlib.patches.Wedge at 0x227886fea90>,
<matplotlib.patches.Wedge at 0x227886fe9a0>,
<matplotlib.patches.Wedge at 0x227886fed90>,
<matplotlib.patches.Wedge at 0x2278872fb20>,
<matplotlib.patches.Wedge at 0x2278872fd30>,
<matplotlib.patches.Wedge at 0x22789ada370>,
<matplotlib.patches.Wedge at 0x22789ada790>],
[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, -0.44229283934431896, ''),
Text(-0.12049606360635531, -0.6489073112975174, ''),
Text(0.39011356818311405, -0.532363976917521, ''),
Text(0.6332653697075483, -0.1859434632601054, '')])
```



14. Круговая диаграмма в виде бублика

```
In [33]: 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[33]: ([<matplotlib.patches.Wedge at 0x22789b8c3d0>,
<matplotlib.patches.Wedge at 0x22789b8c880>,
<matplotlib.patches.Wedge at 0x22789b8cca0>,
<matplotlib.patches.Wedge at 0x22789b98160>,
<matplotlib.patches.Wedge at 0x22789b98610>],
[Text(0.9639373540021144, 0.5299290306818474, 'Ford'),
Text(0.22870287165240302, 1.075962358309037, 'Toyota'),
Text(-1.046162158377023, 0.3399187231970734, 'BMW'),
Text(-0.3617533684721028, -1.0388139873909512, 'AUDI'),
Text(0.8174592712713289, -0.7360437078139777, 'Jaguar')])
```

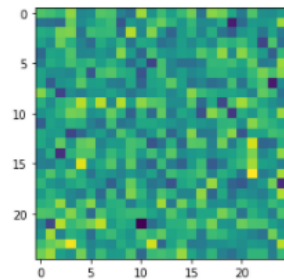


15. Отображение изображений



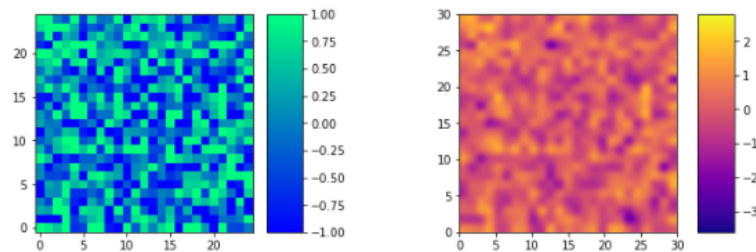
```
In [35]: np.random.seed(19680801)
data = np.random.randn(25, 25)
plt.imshow(data)
```

Out[35]: <matplotlib.image.AxesImage at 0x22789e64370>



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

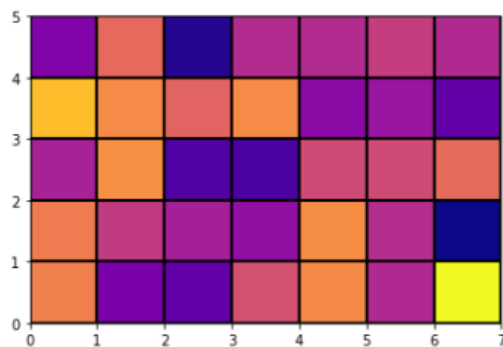
Out[36]: <matplotlib.colorbar.Colorbar at 0x22789f4aac0>



16. Отображение тепловой карты

```
In [37]: np.random.seed(123)
data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
```

Out[37]: <matplotlib.collections.QuadMesh at 0x22789f188e0>



Выполнение индивидуальных заданий:

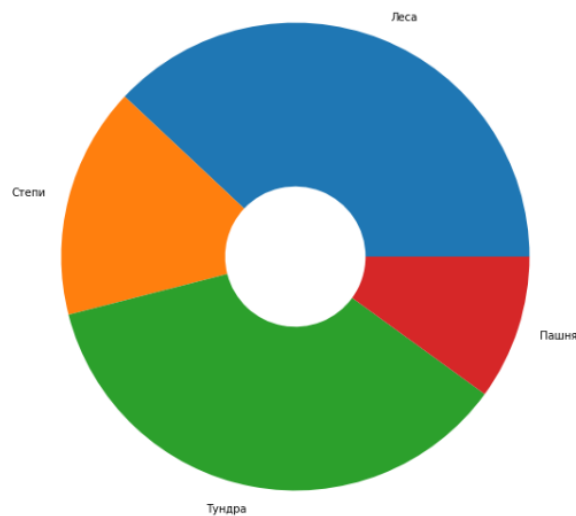
1. Построение диаграммы:

```
In [9]: import matplotlib.pyplot as plt
%matplotlib inline

Terrain = ['Леса', 'Степи', 'Тундра', 'Пашня']
Popular = [57, 24, 54, 15]

fig, ax = plt.subplots(figsize=(16, 10))
ax.pie(Popular, labels=Terrain, wedgeprops=dict(width=0.7))

Out[9]: ([<matplotlib.patches.Wedge at 0x1fab3f92580>,
<matplotlib.patches.Wedge at 0x1fab3ff0e50>,
<matplotlib.patches.Wedge at 0x1fab3fd6f40>,
<matplotlib.patches.Wedge at 0x1fab3fda8b0>],
[Text(0.4049370232742902, 1.0227541284110062, 'Леса'),
Text(-1.065441465972024, 0.2735589197730253, 'Степи'),
Text(-0.20611945413751406, -1.080515974257694, 'Тундра'),
Text(1.0461621822461364, -0.3399186497354948, 'Пашня')])
```



2. Построение столбчатой диаграммы:

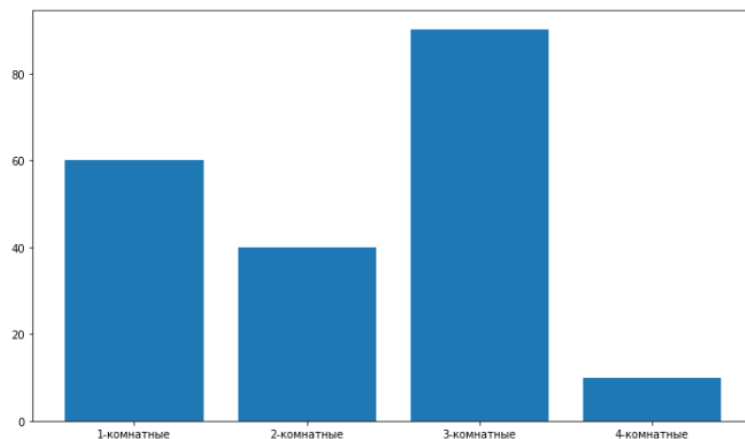
Постройте столбчатую диаграмму по следующим данным: в доме 200 квартир, из них 30% однокомнатные, 20% - двухкомнатные, 45% - трехкомнатные, а остальные четырехкомнатные.

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

plt.figure(figsize=(12, 7))
Country = ['1-комнатные', '2-комнатные', '3-комнатные', '4-комнатные']
Popular = [60, 40]

plt.bar(Apartments, Quantity)

Out[3]: <BarContainer object of 4 artists>
```

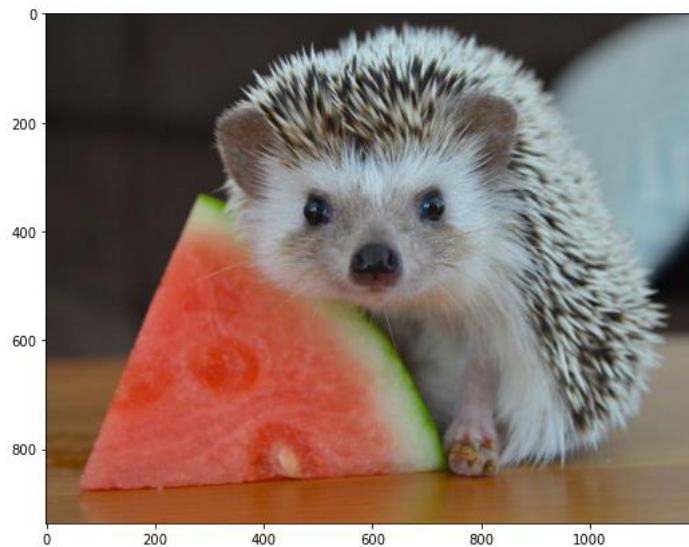


3. Работа с изображениями:

```
In [10]: import matplotlib.pyplot as plt
%matplotlib inline
from PIL import Image
import requests
from io import BytesIO
plt.figure(figsize=(12, 7))

response = requests.get('https://placepic.ru/wp-content/uploads/2018/10/s1200-2-2.jpg')
img = Image.open(BytesIO(response.content))
plt.imshow(img)

Out[10]: <matplotlib.image.AxesImage at 0x1b3e799aca0>
```

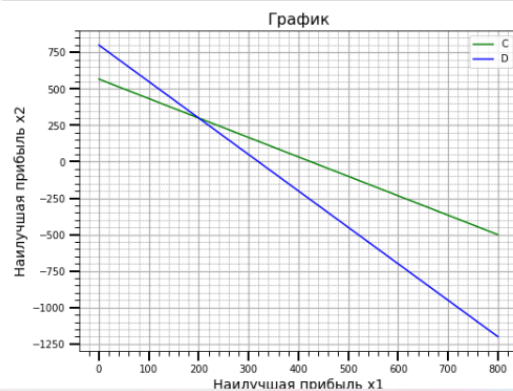


4. Работа с линейными графиками

```
In [4]: import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
                               AutoMinorLocator)
import numpy as np

x1 = np.linspace(0, 800, 10)
x2 = np.linspace(0, 800, 10)
y1 = (1700 - 4*x1)/3
y2 = (1600 - 5*x2)/2
fig, ax = plt.subplots(figsize=(8, 6))
ax.set_title("График", fontsize=16)
ax.set_xlabel("Наилучшая прибыль x1", fontsize=14)
ax.set_ylabel("Наилучшая прибыль x2", fontsize=14)
ax.grid(which="major", linewidth=1.2)
ax.grid(which="minor", linestyle="--", color="gray", linewidth=0.5)

ax.plot(x1, y1, label="C", c="green")
ax.plot(x2, y2, label="D", c="blue")
ax.legend()
ax.xaxis.set_minor_locator(AutoMinorLocator())
ax.yaxis.set_minor_locator(AutoMinorLocator())
ax.tick_params(which='major', length=10, width=2)
ax.tick_params(which='minor', length=5, width=1)
plt.show()
```



Вопросы для защиты:

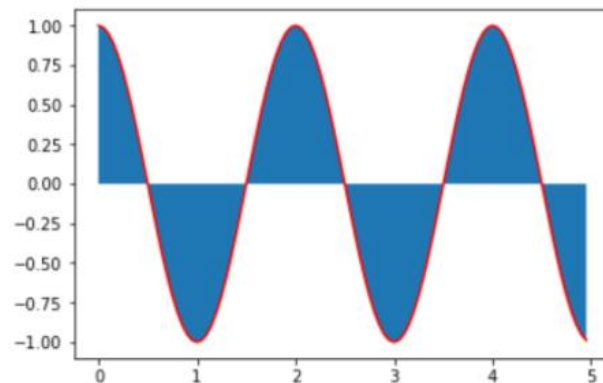
1. Как выполнить построение линейного графика с помощью matplotlib?

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)
```

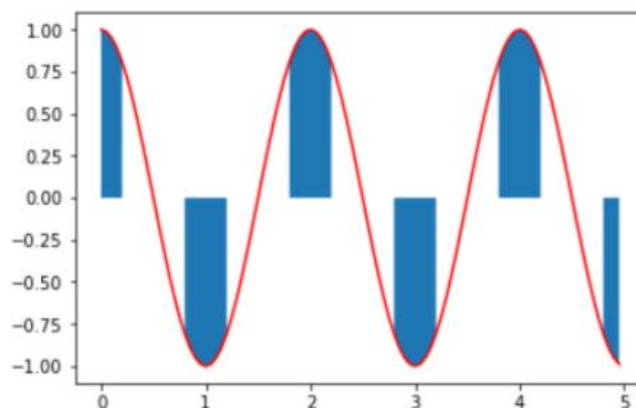
2. Как выполнить заливку области между графиком и осью? Между двумя графиками?

```
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))
```

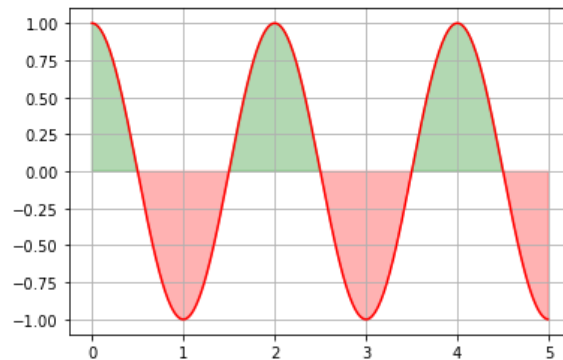


4. Как выполнить двухцветную заливку?

```
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)
```

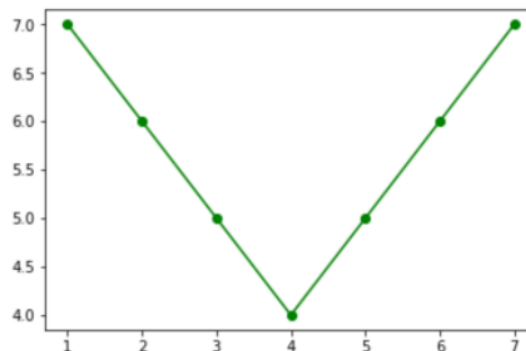
Out[14]: <matplotlib.collections.PolyCollection at 0x1b7583e04f0>



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")
```

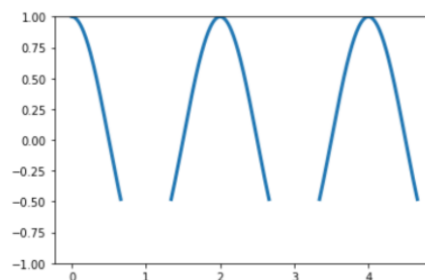


6. Как выполнить обрезку графиков?

```
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)
```

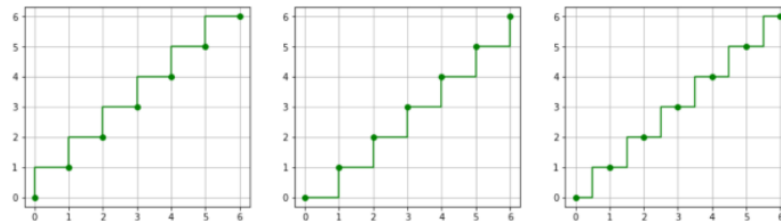


7. Как построить ступенчатый график? В чем особенность ступенчатого графика?

```
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.

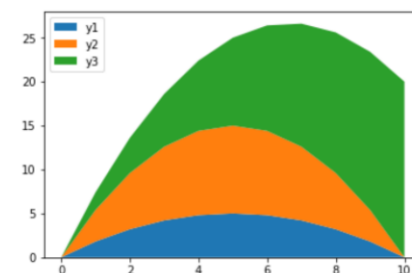
```
x = np.arange(0, 11, 1)

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

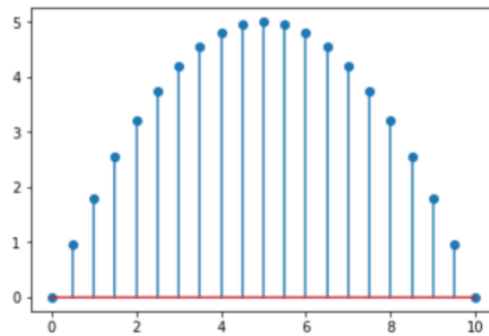


9. Как построить stem-график? В чем особенность stem-графика?

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)
```

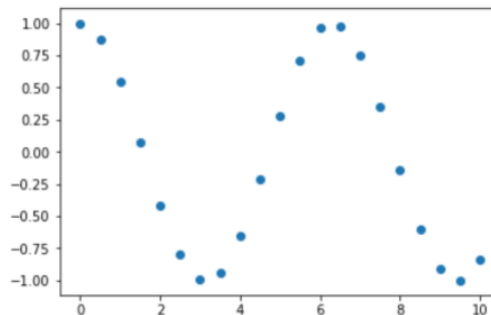


10. Как построить точечный график? В чем особенность точечного графика?

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)
```

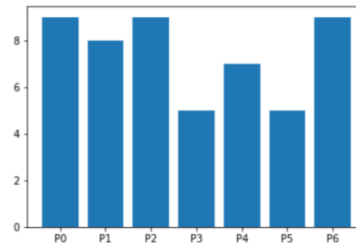


11. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

```
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]

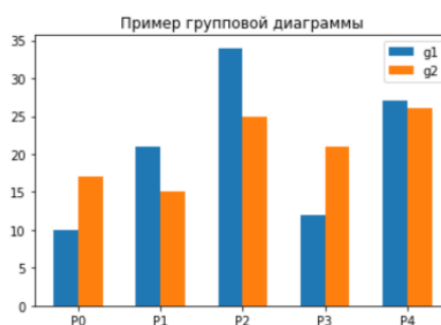
width = 0.3

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

```

np.random.seed(123)

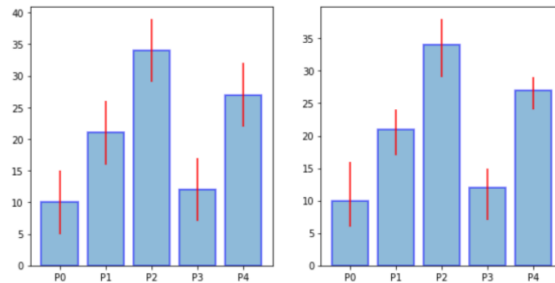
rnd = np.random.randint

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?

```

In [34]: vals = [24, 17, 53, 21, 35]

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

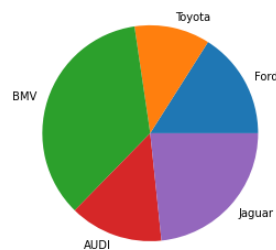
fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis("equal")

```

```

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

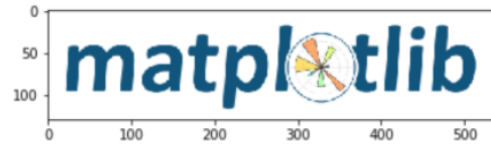
from io import BytesIO

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

plt.imshow(img)

```

В результате получим изображение логотипа *Matplotlib*.

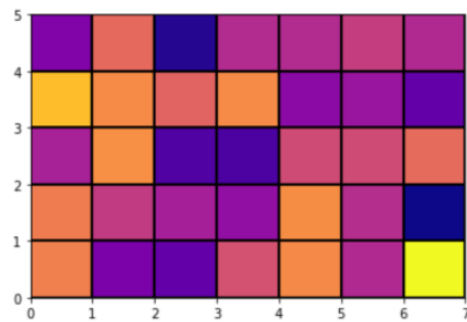


```

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')

```