

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №3.5

Визуализация данных с помощью matplotlib.

по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-21-1

Лысенко И.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2023

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

Ход работы:

1. Создал репозиторий на GitHub:
https://github.com/IsSveshuD/lab_3.5.git .

2. Проработал примеры:

Линейный график

Построение графика

```
In [17]: from matplotlib import pylab as plt
%matplotlib inline

In [18]: 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)
```

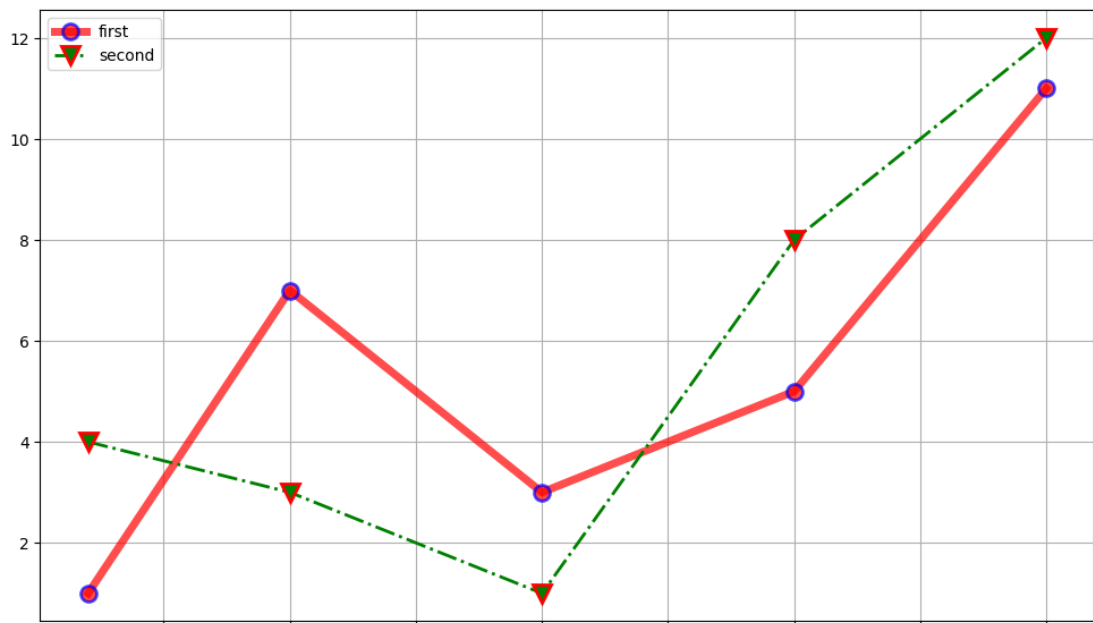


Рисунок 1 – Примеры

3. Выполнил индивидуальное задание.

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.

Задача:

Постройте график функции и с его помощью укажите нули функции, интервалы законопостоянства, промежутки монотонности, точки экстремума, экстремумы функции, наибольшее и наименьшее значения функции, область значений функции:

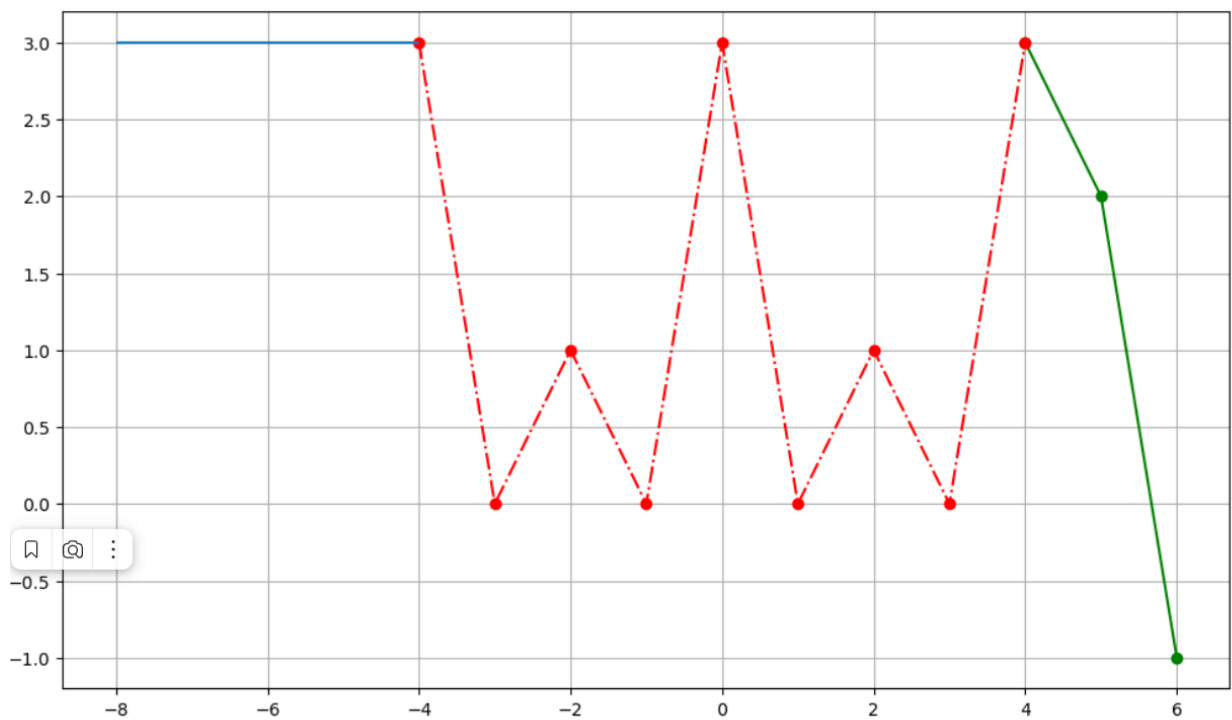
$$y = \begin{cases} 3, & \text{если } x \leq -4 \\ x^2 - 4|x| + 3, & \text{если } -4 < x \leq 4 \\ 3 - (x - 4)^2, & \text{если } x > 4 \end{cases}$$

```
In [106]: import numpy as np
import math

x = np.arange(-8, 7)
x1 = np.arange(-4, 5)
y1 = 3
y2 = abs(np.power(x1, 2) - 4 * abs(x1) + 3)
y3 = (3 - np.power(x - 4, 2))

y1_masked = np.ma.masked_where(x1 > 4, y2)
y2_masked = np.ma.masked_where(x1 > 4, y2)
y3_masked = np.ma.masked_where(x < 4, y3)

plt.figure(figsize=(12, 7))
plt.plot(x, y3_masked, marker="o", c="g")
plt.plot(x1, y2_masked, 'o-.r')
plt.hlines(y = 3, xmin = -8, xmax = -4)
plt.grid(True)
```



Ответ:

$$y(x) = 0 \text{ при } x \in \{-3; -1; 1; 3; 4 + \sqrt{3}\}$$

$$y(x) < 0 \text{ при } x \in (4 + \sqrt{3}; +\infty)$$

$$y(x) > 0 \text{ при } x \in (-\infty; -3), (-3; -1), (-1; 1), (1; 3), (3; 4 + \sqrt{3})$$

$$y(x) \text{ возрастает при } x \in [-3; -2], [-1; 0], [1; 2], [3; 4]$$

$$y(x) \text{ убывает при } x \in [-4; -3], [-2; -1], [0; 1], [2; 3], [4; +\infty)$$

точки максимума, максимум функции: $(x; 3), x \in (-\infty; -4]; (-2; 1); (0; 3); (2; 1); (4; 3)$

точки минимума, минимум функции: $(-3; 0); (-1; 0); (1; 0); (3; 0)$

$\max y(x) = 3$; $\min y(x)$ не существует

$$E(y) = (-\infty; 3]$$

Рисунок 2 – Индивидуальное задание 1

4. Выполнил индивидуальное задание 2:

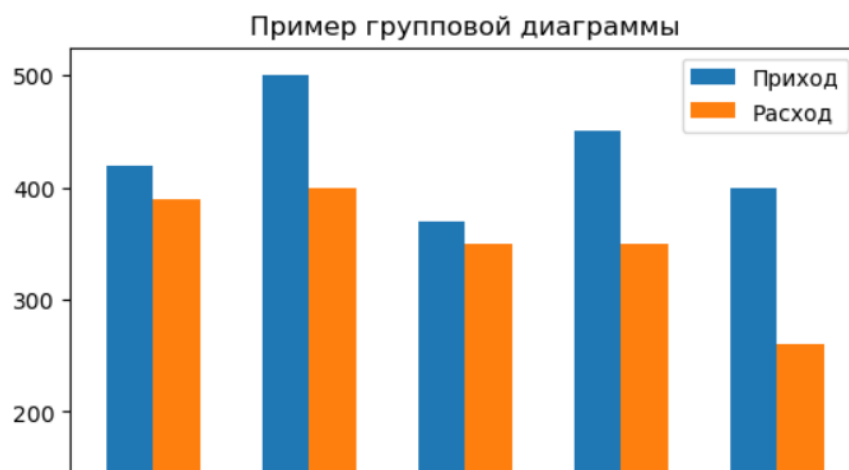
Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем

Постройте гистограмму (столбчатую диаграмму) прихода и расхода в зависимости от года. Определите год с наибольшей разницей.

```
In [108]: cat_par = [2000, 2001, 2002, 2003, 2004]
g1 = [420, 500, 370, 450, 400]
g2 = [390, 400, 350, 350, 260]

width = 0.3
x = np.arange(len(cat_par))
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, g1, width, label='Приход')
rects2 = ax.bar(x + width/2, g2, width, label='Расход')
ax.set_title('Пример групповой диаграммы')
ax.set_xticks(x)
ax.set_xticklabels(cat_par)
ax.legend()
```

Out[108]: <matplotlib.legend.Legend at 0x1e50d0f3190>



Ответ: 2004

Рисунок 3 – Индивидуальное задание 2.

5. Выполнил индивидуальное задание 3:

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

Постройте круговую диаграмму распределения суши на Земле по следующим данным:

Вид суши	Занимаемая площадь, млн.кв.км.	Сектор диаграммы, градусы
Леса	57	--
Степи	24	--
Тундры, пустыни и болота	54	--
Пашня	15	--
Всего:	--	360°

```
In [128]: vals = np.array([57, 24, 54, 16])
          valss = (vals/360) * 100

          labels = ["Леса", "Степи", "Тундры, пустыни и болота", "Пашня"]
          explode = (0.1, 0, 0.15, 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[128]: (-1.1086334175999055,
           1.1397454116712409,
           -1.258387261083339,
           1.2046352287018385)
```

```
Out[128]: (-1.1086334175999055,
           1.1397454116712409,
           -1.258387261083339,
           1.2046352287018385)
```

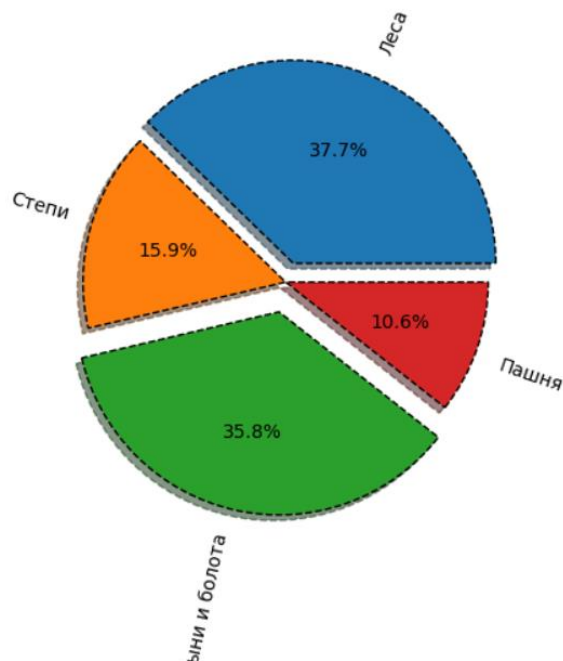


Рисунок 3 – Индивидуальное задание 3.

6. Выполнил индивидуальное задание 4:

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

```
In [129]: from PIL import Image
import requests

from io import BytesIO

response = requests.get('https://тревеларгент.рф/wp-content/uploads/2023/04/skfu-scaled.jpg')
img = Image.open(BytesIO(response.content))
plt.imshow(img)

Out[129]: <matplotlib.image.AxesImage at 0x1e50da0df60>
```



Рисунок 3 – Индивидуальное задание 4.

Ответы на вопросы:

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)
```

Если вызвать функцию `plot()` с одним аргументом – вот так: `plot(y)`, то мы получим график, у которого по оси ординат (ось y) будут отложены значения из переданного списка, по по оси абсцисс (ось x) – индексы элементов массива.

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

Для заливки областей используется функция `fill_between()`. Сигнатура функции:

Основные параметры функции:

– x : массив длины N - набор данных для оси абсцисс.

- `y1` : массив длины `N` или скалярное значение - набор данных для оси ординат – первая кривая.
- `y2` : массив длины `N` или скалярное значение - набор данных для оси ординат – вторая кривая.
- `where` : массив `bool` элементов (длины `N`), optional, значение по умолчанию: `None` - задает
- заливаемый цветом регион, который определяется координатами `x[where]`: интервал будет
- залит между `x[i]` и `x[i+1]`, если `where[i]` и `where[i+1]` равны `True`.
- `step` : `{‘pre’, ‘post’, ‘mid’}`, optional - определяет шаг, если используется `step` функция для отображения графика.
- `**kwargs` - свойства класса `Polygon`

3. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию?

Используя параметры `y1` и `y2` можно формировать более сложные решения.

Заливка области между 0 и `y`, при условии, что `y >= 0`:

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

```
plt.fill_between(x, y, where=(y > 0))
```

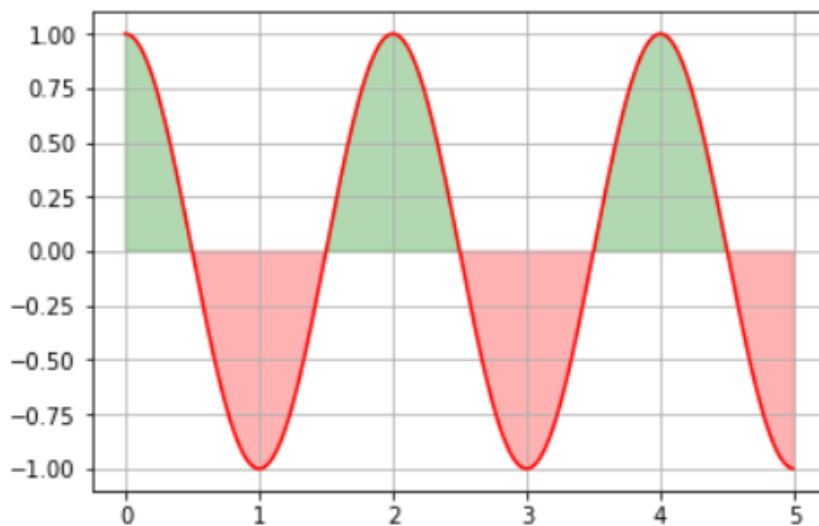
4. Как выполнить двухцветную заливку?

```
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)
```



5. Как выполнить маркировку графиков?

marker: str – определяет тип маркера.

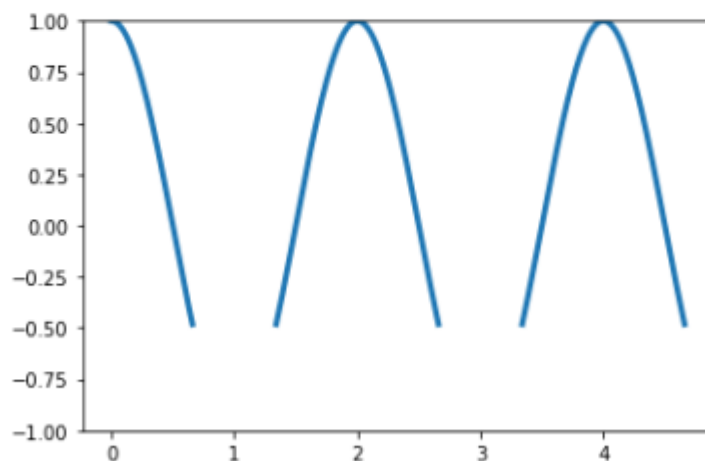
6. Как выполнить обрезку графиков?

Для того, чтобы отобразить только часть графика, которая отвечает определенному условию используйте предварительное маскирование данных с помощью функции `masked_where` из пакета `numpy`.

```
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?

Такой график строится с помощью функции `step()`, которая принимает следующий набор параметров:

- `x`: `array_like` - набор данных для оси абсцисс
- `y`: `array_like` - набор данных для оси ординат
- `fmt`: `str`, optional - задает отображение линии (см. функцию `plot()`).
- `data`: `indexable object`, optional - метки.
- `where` : `{'pre', 'post', 'mid'}`, optional , по умолчанию `'pre'` - определяет место, гдебудет установлен шаг;

`'pre'`: значение `y` ставится слева от значения `x`, т.е. значение `y[i]` определяется для интервала `(x[i-1]; x[i])`.

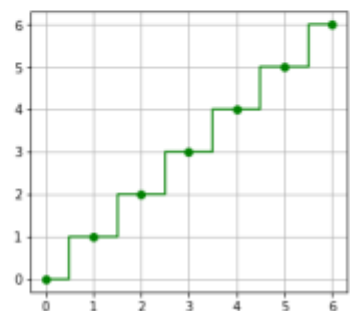
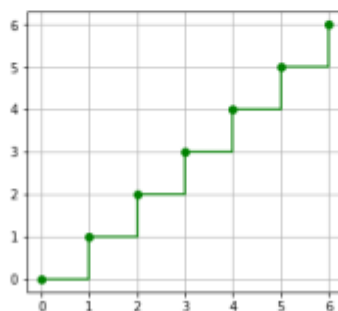
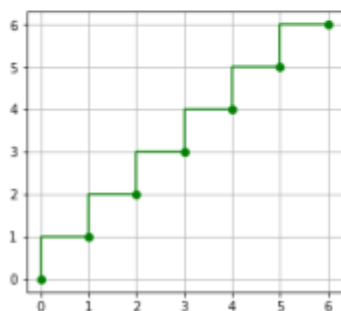
`'post'`: значение `y` ставится справа от значения `x`, т.е. значение `y[i]` определяется для интервала `(x[i]; x[i+1])`.

`'mid'`: значение `y` ставится в середине интервала.

```
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:

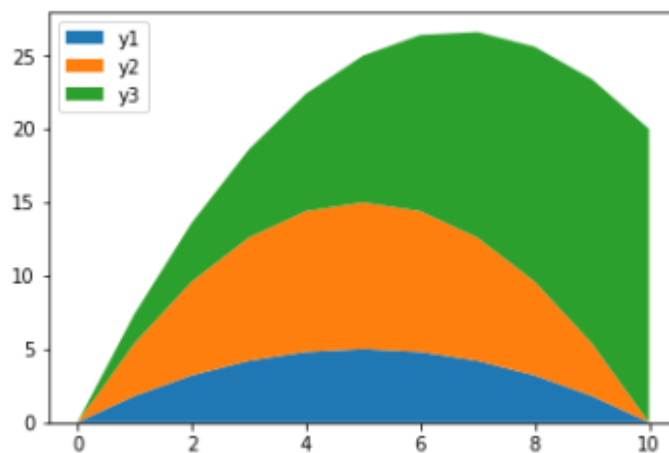
```
x = np.arange(0, 11, 1)

y1 = np.array([(-0.2)*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')
```

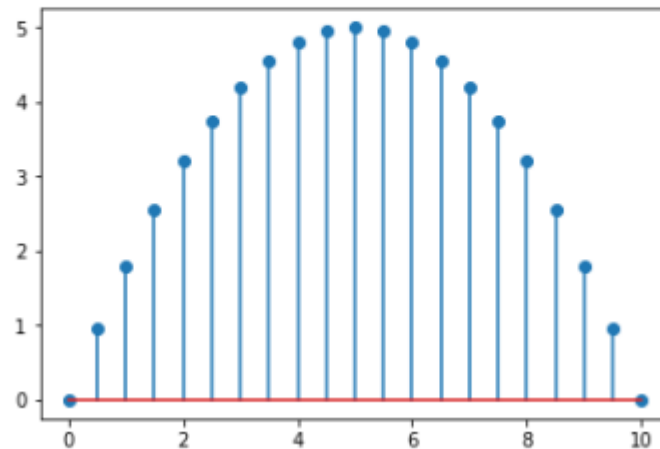


9. Как построить stem-график? В чем особенность stem-графика?

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)**2+2*i for i in x])

plt.stem(x, y)
```

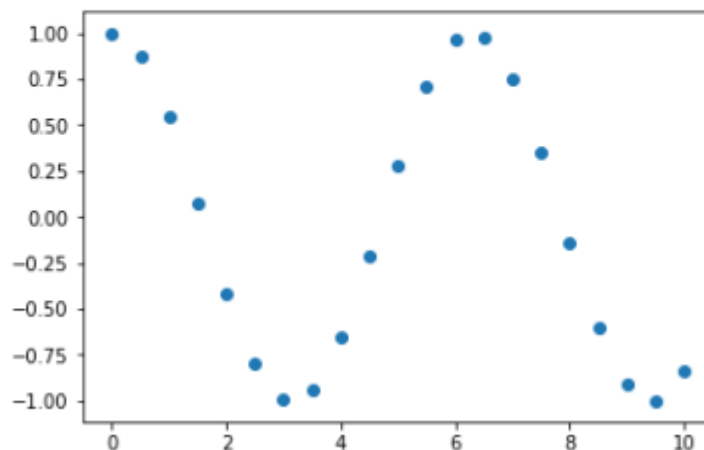


10. Как построить точечный график? В чем особенность точечного графика?

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

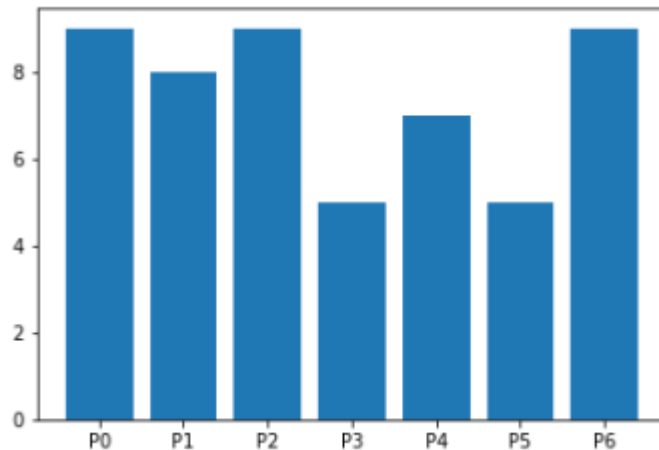
`barh()` – для построения горизонтальной диаграммы.

```
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?

Используя определенным образом подготовленные данные можно строить групповые диаграммы:

```
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')
```

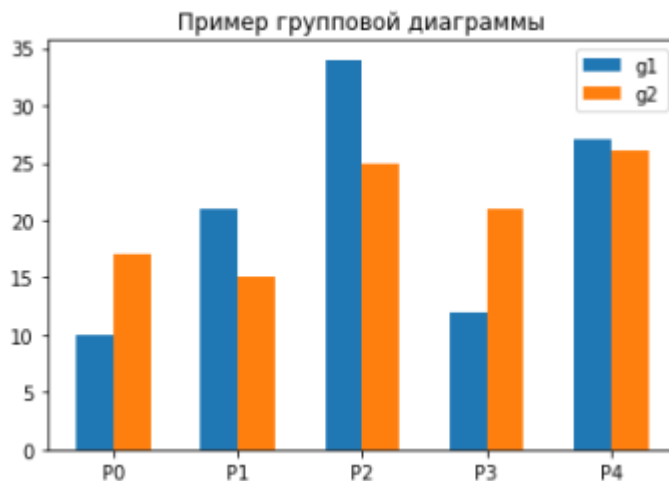
```
rects2 = ax.bar(x + width/2, g2, width, label='g2')
```

```
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)]
```

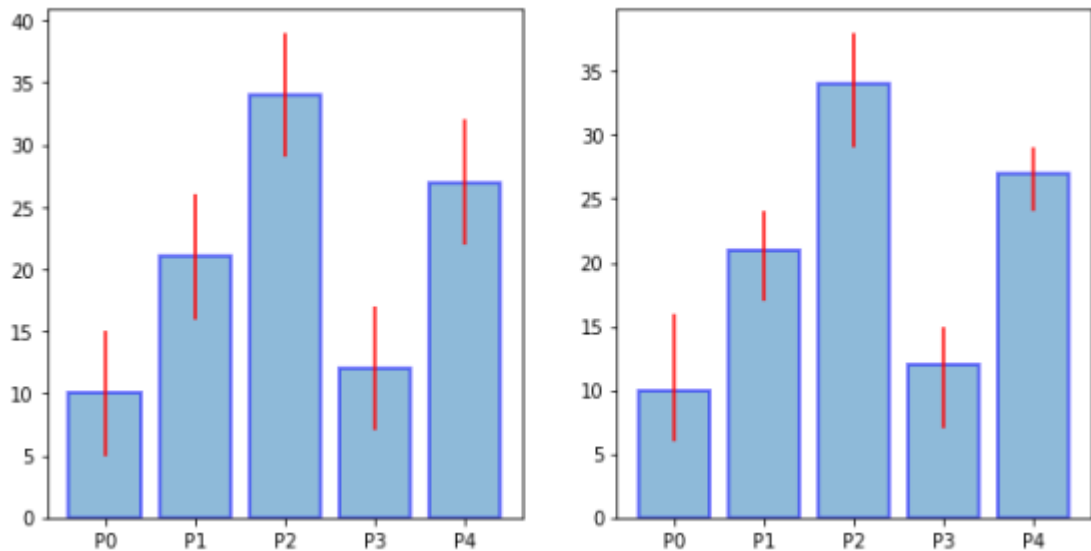
```
g1 = [10, 21, 34, 12, 27]
```

```
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()`.

Пример построения диаграммы:

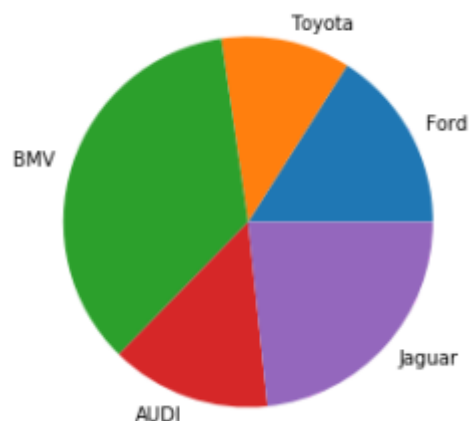
```
vals = [24, 17, 53, 21, 35]
```

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

```
fig, ax = plt.subplots()
```

```
ax.pie(vals, labels=labels)
```

```
ax.axis("equal")
```



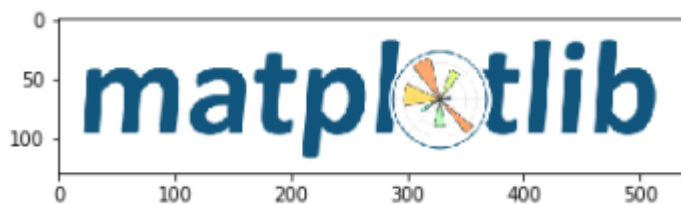
14. Что такое цветовая карта? Как осуществляется работа с цветовыми картами в matplotlib?

Цветовая карта представляет собой подготовленный набор цветов, который хорошо подходит для визуализации того или иного набора данных.

15. Как отобразить изображение средствами matplotlib?

Основное назначение функции `imshow()` состоит в представлении 2d растров. Это могут быть картинки, двумерные массивы данных, матрицы и т.п.

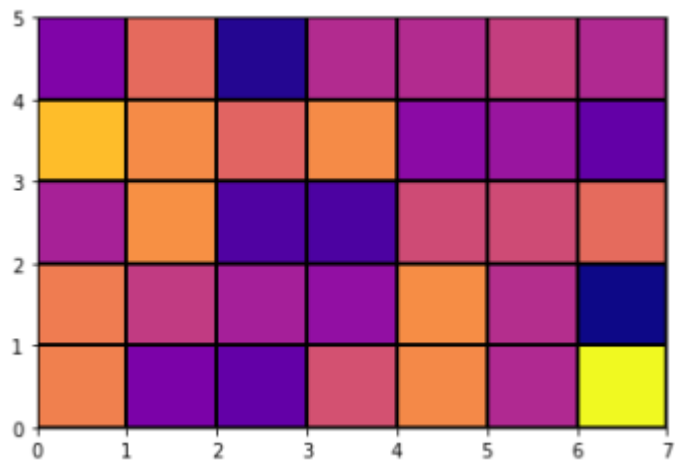
```
from PIL import Image
import requests
from io import BytesIO
response = requests.get('https://matplotlib.org/_static/logo2.png')
img = Image.open(BytesIO(response.content))
plt.imshow(img)
```



16. Как отобразить тепловую карту средствами matplotlib?

С помощью функции `pcolormesh()`.

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



Вывод: были исследованы базовые возможности визуализации данных на плоскости средствами библиотеки `matplotlib` языка программирования Python.