Лекция 10. Создание графиков. Визуализация данных

Matplotlib

- **Matplotlib —** библиотека на языке программирования Python для визуализации данных двумерной 2D и трехмерной графики 3D.

- https://matplotlib.org/ - Официальный сайт библиотеки Matplotlib
- https://matplotlib.org/stable/contents.html - Руководство пользователя Matplotlib
- https://matplotlib.org/stable/gallery/index.html - Примеры графиков Matplotlib
- https://github.com/matplotlib/cheatsheets#cheatsheets – Шпаргалки по Matplotlib

Matplotlib

- Пакет поддерживает многие виды графиков и диаграмм:
  - Графики (line plot)
  - Диаграммы разброса (scatter plot)
  - Столбчатые диаграммы (bar chart) и гистограммы (histogram)
  - Круговые диаграммы (pie chart)
  - Ствол-лист диаграммы (stem plot)
  - Контурные графики (contour plot)
  - Поля градиентов (quiver)
  - Спектральные диаграммы (spectrogram)

Визуализация данных. Библиотека Matplotlib

- **Библиотека `Matplotlib` является одним из самых популярных средств визуализации данных на Python.**
- Она отлично подходит как для создания статичных изображений, так и анимированных, и интерактивных решений.
- Matplotlib является частью Scientific Python — набора библиотек для научных вычислений и визуализации данных, куда также входят NumPy, SciPy, Pandas, SymPy и ещё ряд других инструментов.

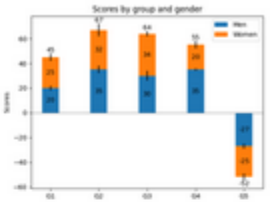
Matplotlib - модуль Pyplot

- Для построения графиков из библиотеки Matplotlib нужно импортировать модуль **Pyplot**.
- **Pyplot** это набор команд, созданных для построения графиков функций и уравнений.
- Для удобного построения графиков так же можно использовать библиотеку **NumPy**.

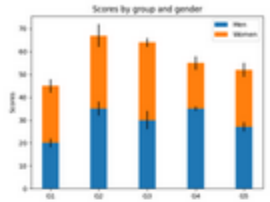
Примеры применения

Matplotlib – Gallery

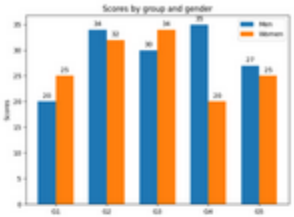
Matplotlib – Gallery - Lines, bars and markers



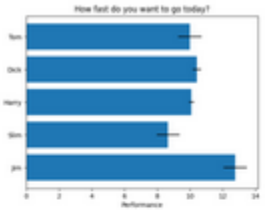
Bar Label Demo



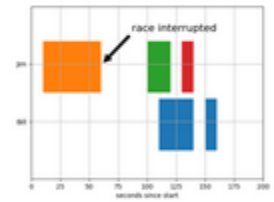
Stacked bar chart



Grouped bar chart with labels

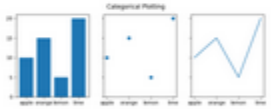


Horizontal bar chart

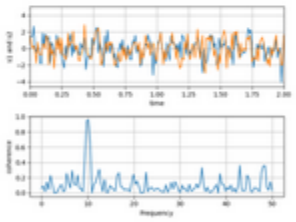


Broken Barh

CapStyle



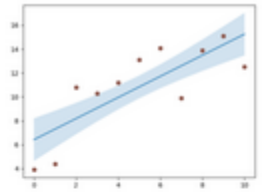
Plotting categorical variables



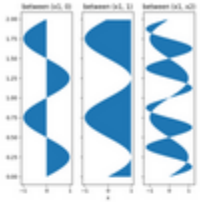
Plotting the coherence of two signals

https://matplotlib.org/stable/gallery/index.html

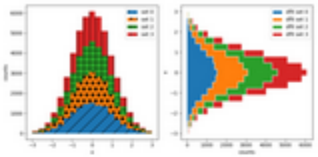
Matplotlib – Gallery - Lines, bars and markers



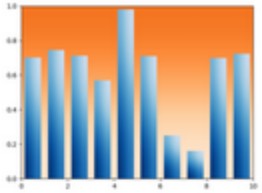
Filling the area between lines



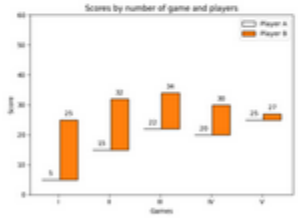
Fill Betweenx Demo



Hatch-filled histograms

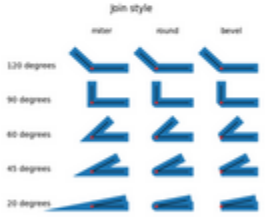


Bar chart with gradients

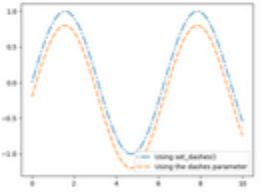


Hat graph

Discrete distribution as horizontal bar chart



JoinStyle



Customizing dashed line styles

Matplotlib

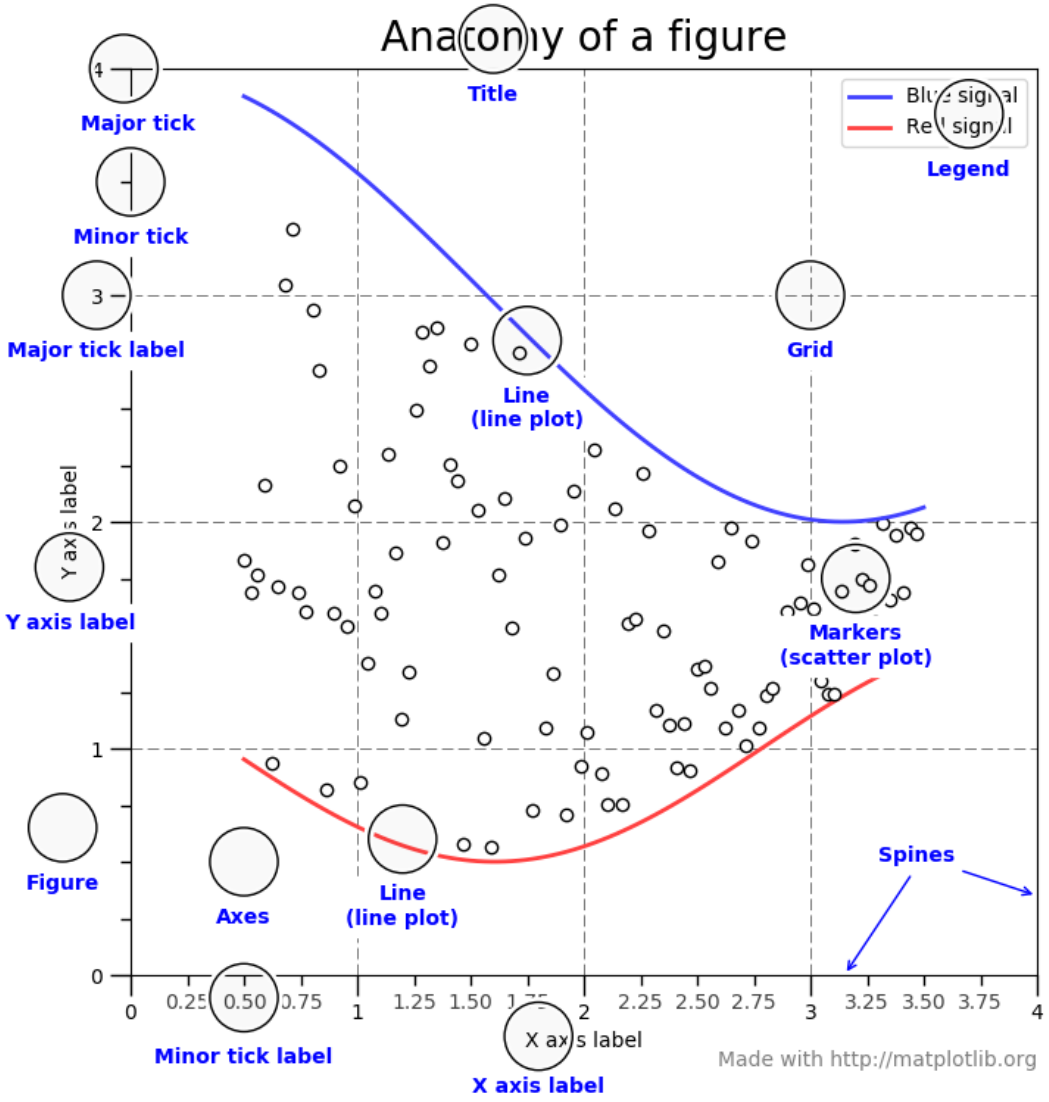
Создание графиков

Matplotlib - Подключение

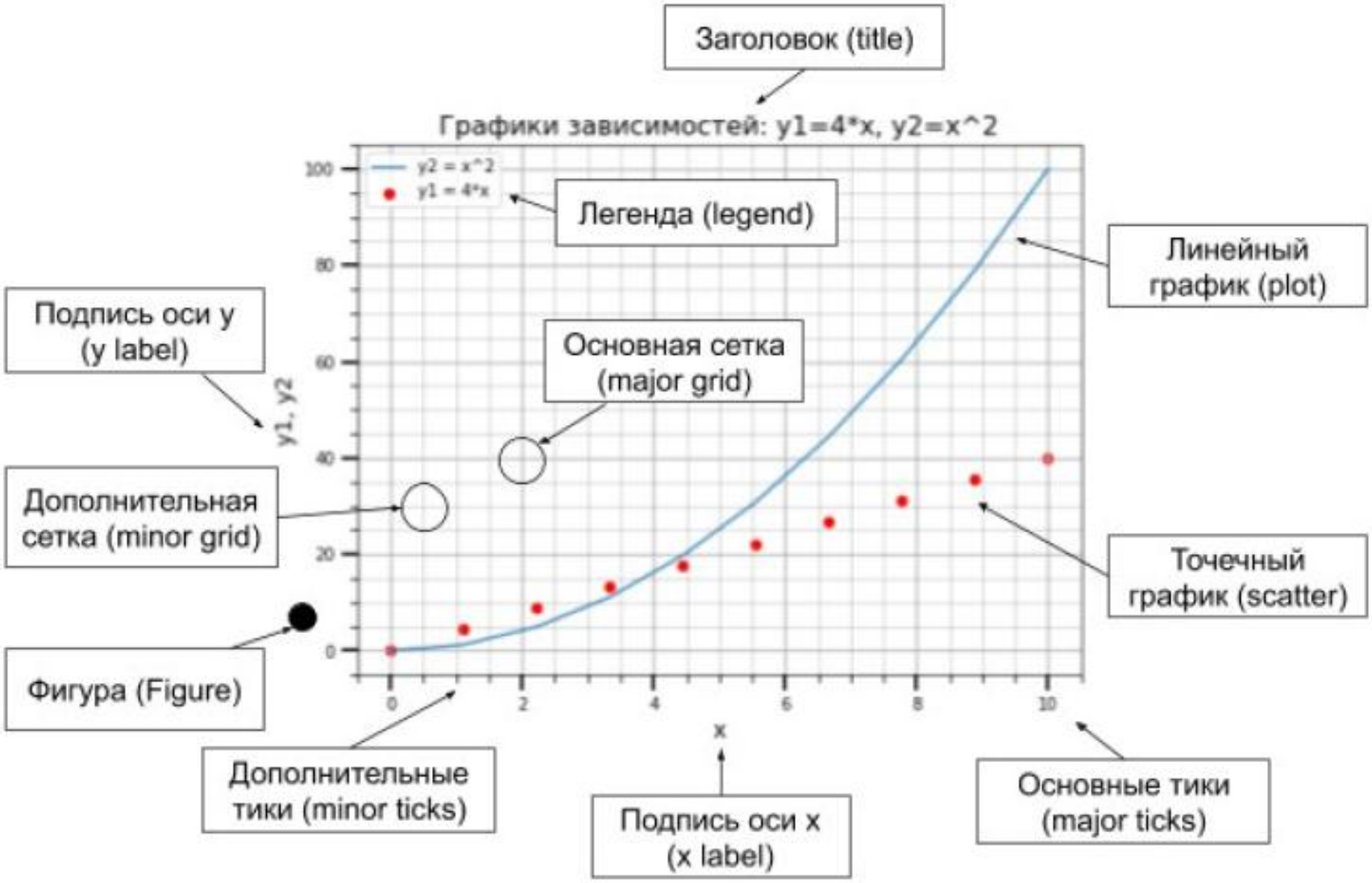
```
# подключение набор команд для работы с графиками из библиотеки matplotlib
import matplotlib.pyplot as plt

# подключение библиотеки numpy для выполнения математических расчетов и работы с матрицами (массивами, списками)
import numpy as np
```

Matplotlib - Основные элементы графика



Matplotlib - Основные элементы графика



Matplotlib Некоторые функции отрисовки

- **`plt.scatter(x, y, params)`** — нарисовать точки с координатами из x по горизонтальной оси и из y по вертикальной оси;
- **`plt.plot(x, y, params)`** — нарисовать график по точкам с координатами из x по горизонтальной оси и из y по вертикальной оси. Точки будут соединятся в том порядке, в котором они указаны в этих массивах;
- **`plt.fill_between(x, y1, y2, params)`** — закрасить пространство между y1 и y2 по координатам из x;
- **`plt.pcolormesh(x1, x1, y, params)`** — закрасить пространство в соответствии с интенсивностью y;
- **`plt.contour(x1, x1, y, lines)`** — нарисовать линии уровня. Затем нужно применить `plt.clabel`.

Matplotlib Вспомогательные функции

- **`plt.figure(figsize=(x, y))`** — создать график размера (x,y);
- **`plt.show()`** — показать график;
- **`plt.subplot(...)`** — добавить подграфик;
- **`plt.xlim(x_min, x_max)`** — установить пределы графика по горизонтальной оси;
- **`plt.ylim(y_min, y_max)`** — установить пределы графика по вертикальной оси;
- **`plt.title(name)`** — установить имя графика;
- **`plt.xlabel(name)`** — установить название горизонтальной оси;
- **`plt.ylabel(name)`** — установить название вертикальной оси;
- **`plt.legend(loc=...)`** — сделать легенду в позиции loc;
- **`plt.grid()`** — добавить сетку на график;
- **`plt.savefig(filename)`** — сохранить график в файл.

plt.plot

- **`plt.plot(x, y, params)`** — нарисовать график по точкам с координатами из **`x`** по горизонтальной оси и из **`y`** по вертикальной оси.
- Точки будут соединятся в том порядке, в котором они указаны в этих массивах;

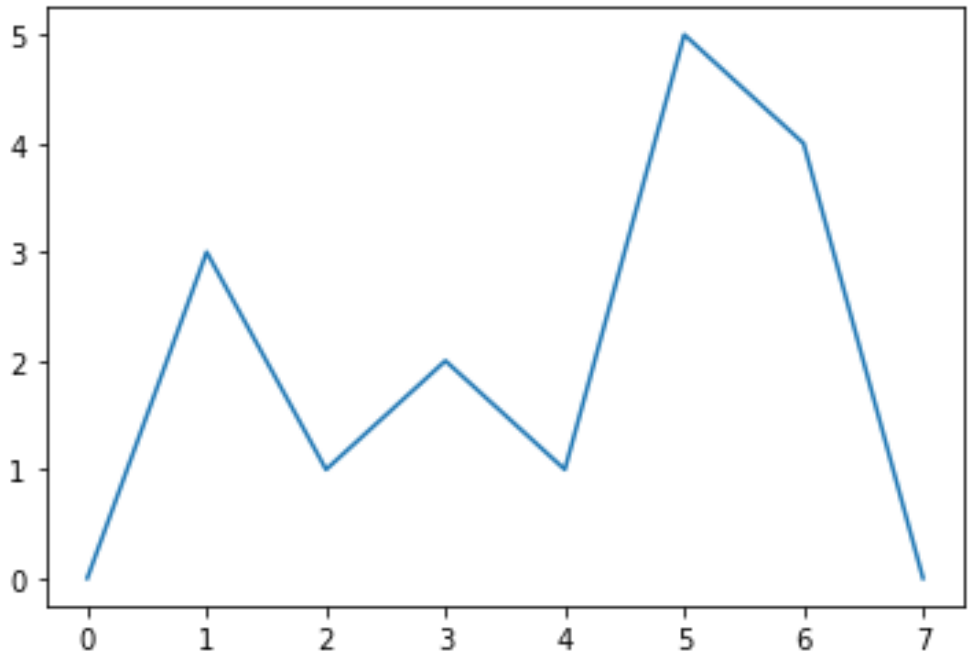
Matplotlib - График линии

- **Метод построения линии очень прост:**
  - есть массив абсцис (x);
  - есть массив ординат (y);
  - элементы с одинаковым индексом в этих массивах - это координаты точек на плоскости;
  - последовательные точки соединяются линией.

- Под массивами, подразумеваются списки, кортежи или массивы NumPy.

Matplotlib - График линии

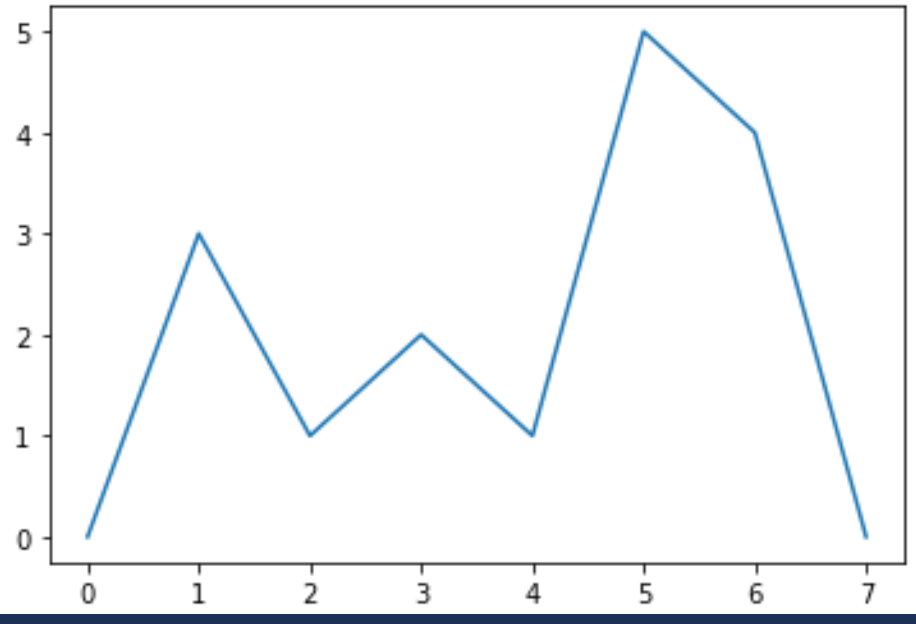
```
import matplotlib.pyplot as plt
plt.plot((0, 1, 2, 3, 4, 5, 6, 7),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



Matplotlib - График линии

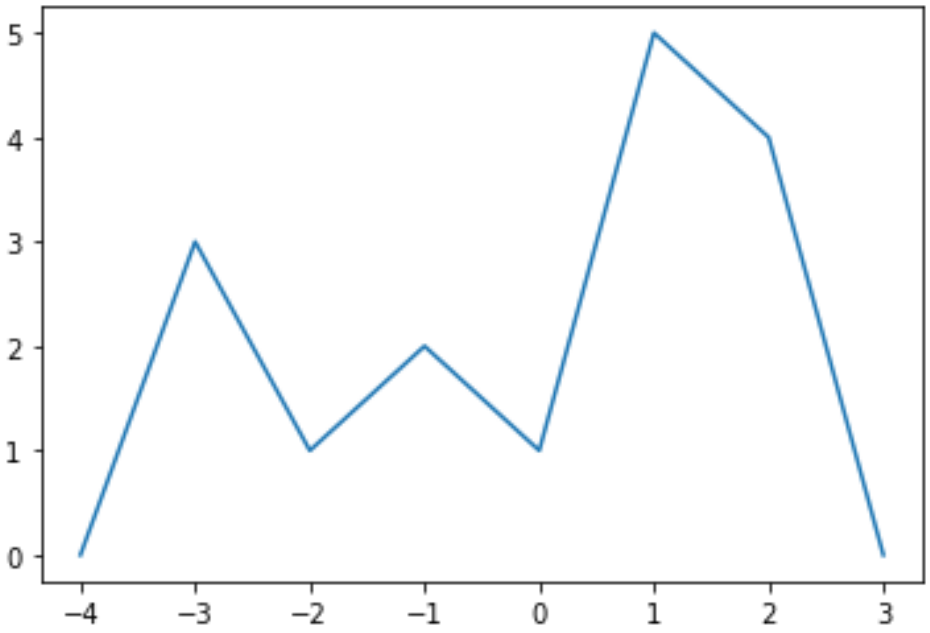
- Метод **plt.plot()**, в простейшем случае, принимает один аргумент - последовательность чисел, которая соответствует оси ординат (y), ось абсцис (x) строится автоматически от 0 до n, где n - это длинна массива ординат.

```
import matplotlib.pyplot as plt
plt.plot((0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



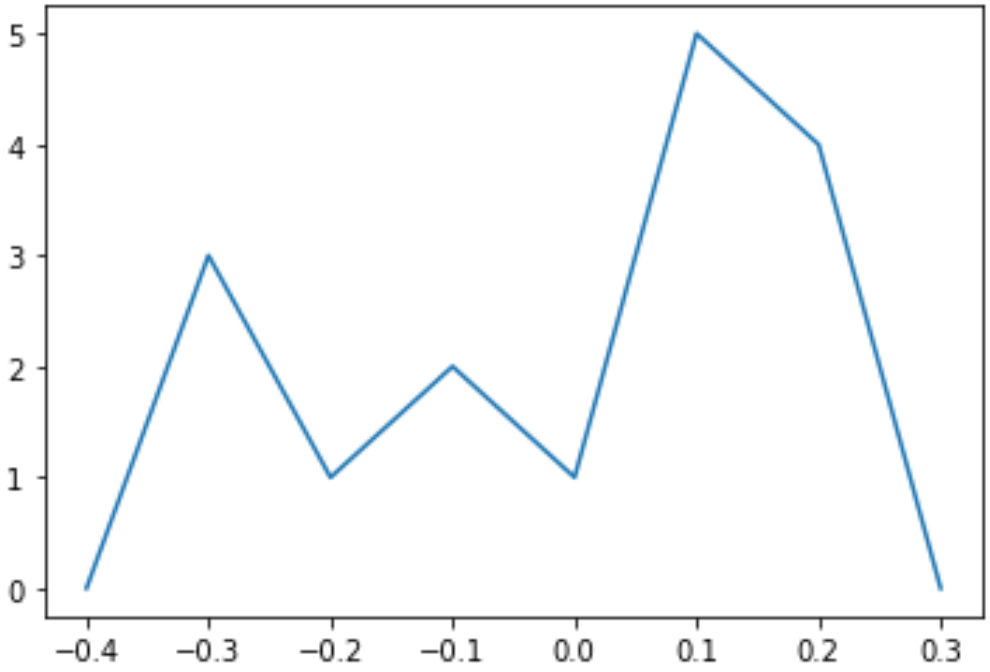
Matplotlib - График линии

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((-4, -3, -2, -1, 0, 1, 2, 3),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



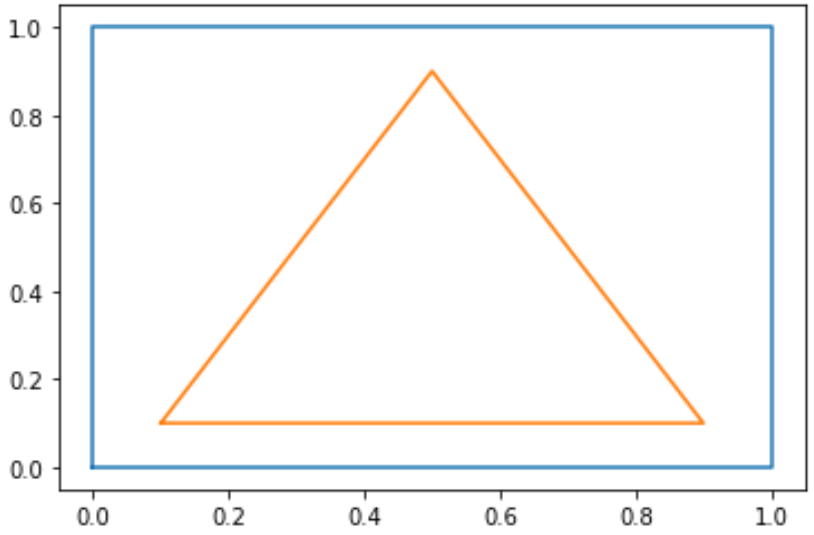
Matplotlib - График линии

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((-0.4, -0.3, -0.2, -0.1, 0., 0.1, 0.2, 0.3),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



Matplotlib – Рисуем фигуры линиями

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((0, 0, 1, 1, 0),
         (0, 1, 1, 0, 0))
plt.plot((0.1, 0.5, 0.9, 0.1),
         (0.1, 0.9, 0.1, 0.1))
plt.show()
```



Цвет и стиль графиков

Стили линии линейного графика

Значение параметра	Описание
'-' или 'solid'	Непрерывная линия.
'--' или 'dashed'	Штриховая линия.
'-.' или 'dashdot'	Штрихпунктирная линия.
':' или 'dotted'	Пунктирная линия.
'None' или ' ' или ''	Не отображать линию.

- **Цвет линии графика** задаётся через параметр color (или c, если использовать сокращённый вариант). Значение может быть представлено в одном из следующих форматов:
  - RGB или RGBA: кортеж значений с плавающей точкой в диапазоне [0, 1] (пример: (0.1, 0.2, 0.3);
  - RGB или RGBA: значение в hex формате (пример: '#0a0a0a');
  - строковое представление числа с плавающей точкой в диапазоне [0, 1] (определяет цвет в шкале серого) (пример: '0.7');
  - символ из набора: {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'};
  - имя цвета из палитры X11/CSS4;
  - цвет из палитры xkcd (https://xkcd.com/color/rgb/), должен начинаться с префикса 'xkcd:';
  - цвет из набора Tableau Color (палитра T10), должен начинаться с префикса 'tab:'.
- Если цвет задаётся с помощью символа из набора {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'}, то он может быть совмещён со стилем линии в рамках параметра fmt функции plot(). Например: штриховая красная линия будет задаваться так: '--r', а штрихпунктирная зелёная так '-.g':

Столбчатые и круговые диаграммы

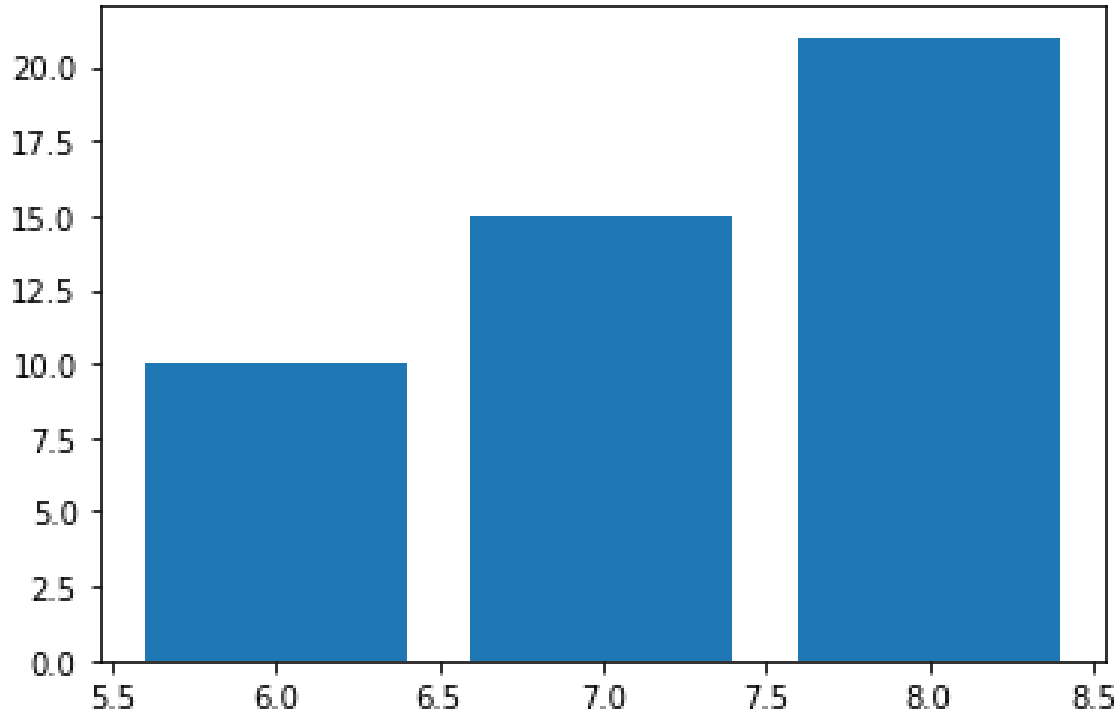
Столбчатые диаграммы

- Для визуализации категориальных данных хорошо подходят **столбчатые диаграммы**.

- Для их построения используются функции:
  - **`bar()`** — вертикальная столбчатая диаграмма;
  - **`barh()`** — горизонтальная столбчатая диаграмма.

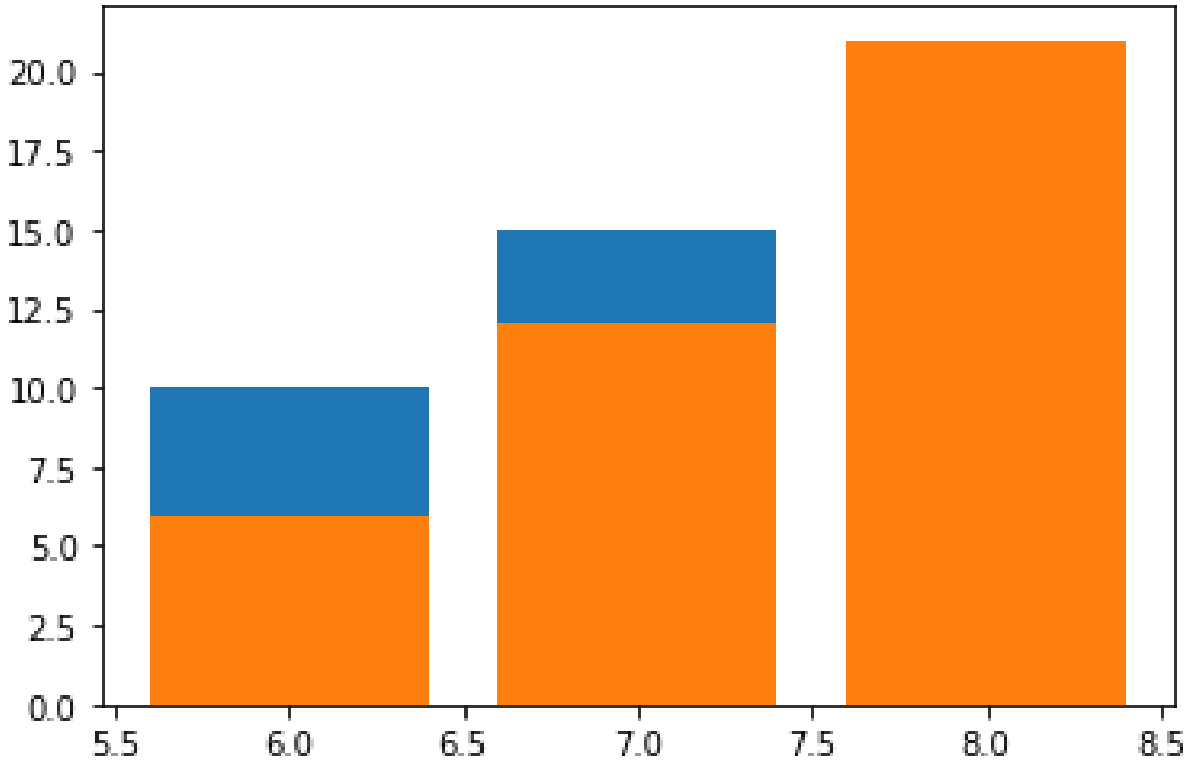
Гистограммы

```
import matplotlib.pyplot as plt
plt.bar([6, 7, 8],
        [10, 15, 21])
plt.show()
```



Гистограммы с несколькими наборами данных

```
import matplotlib.pyplot as plt
plt.bar([6, 7, 8], [10, 15, 21])
plt.bar([6, 7, 8], [6, 12, 21])
plt.show()
```

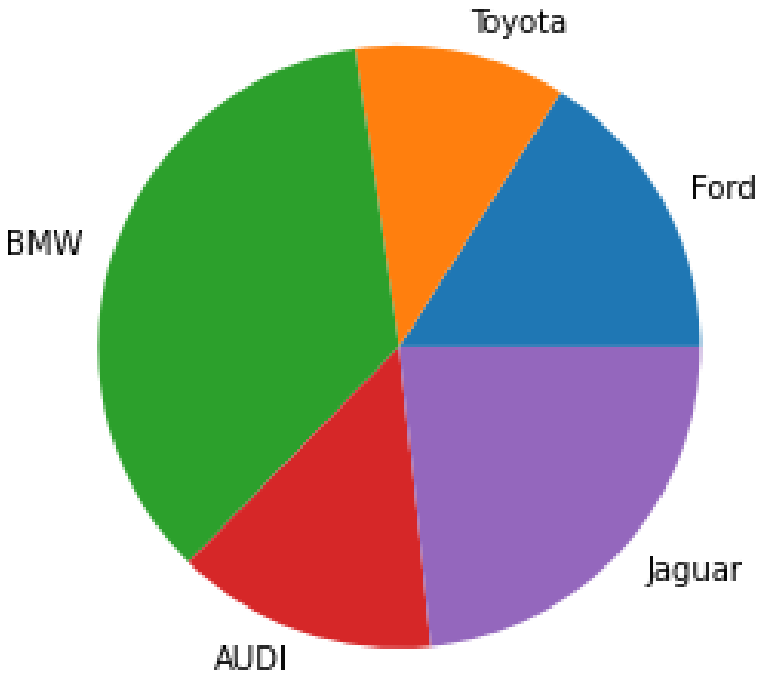


Круговые диаграммы

- **Круговые диаграммы** — это наглядный способ показать доли компонентов в наборе.
- Они идеально подходят для отчётов, презентаций и т.п.
- Для построения круговых диаграмм в Matplotlib используется функция `pie()`.

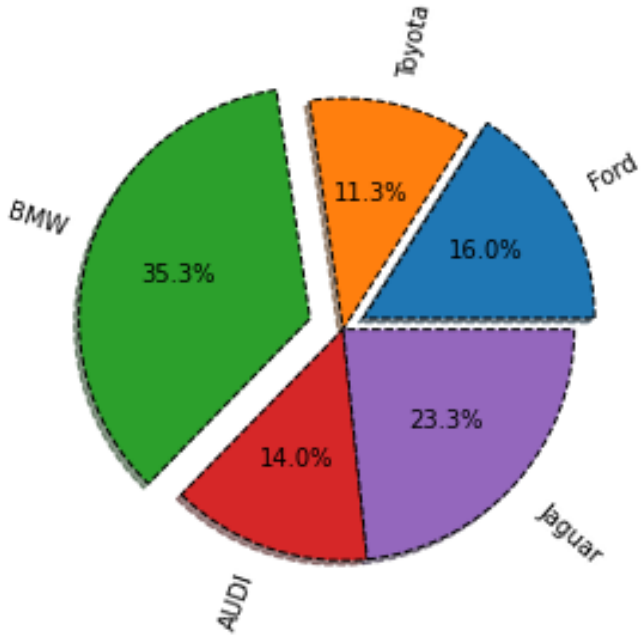
Круговая диаграмма

```
import matplotlib.pyplot as plt
vals = [24, 17, 53, 21, 35]
labels = ['Ford', 'Toyota', 'BMW', 'AUDI', 'Jaguar']
fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis('equal')
```



Модифицированная круговая диаграмма

```
import matplotlib.pyplot as plt
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')
```



Список использованных источников

1. Python
   https://www.python.org/
2. Google Colaboratory
   https://colab.research.google.com/
3. Matplotlib: Visualization with Python
   https://matplotlib.org/
4. Matplotlib User's Guide
   https://matplotlib.org/stable/Matplotlib.pdf
5. Библиотека matplotlib
   https://mipt-stats.gitlab.io/courses/python/06_matplotlib.html
6. Matplotlib Я новичок. Можно попроще? | NumPy
   https://pyprog.pro/mpl/mpl_types_of_graphs.html
7. Matplotlib Gallery
   https://matplotlib.org/stable/gallery/index.html
8. Python в научных вычислениях
   https://inp.nsk.su/~grozin/python/
9. matplotlib: пакет для построения графиков
   https://inp.nsk.su/~grozin/python/b22_matplotlib.html