

Лабораторная работа 2. Основы библиотеки `matplotlib`

1. Введение в `matplotlib`

`Matplotlib` – это библиотека на Python для **визуализации данных**. Она используется для построения графиков, диаграмм и других визуальных представлений числовой информации.

Библиотека используется в:

- Data Science
- Machine Learning
- Научных исследованиях
- Аналитике данных
- Инженерных расчётах и пр.

Установка и импорт

Установка

```
pip install matplotlib
```

Также нам пригодится библиотека `numpy`

```
import matplotlib.pyplot as plt
import numpy as np
```

2. Архитектура

```
Figure (вся картинка)
├─ Axes (область графика)
│   └─ Plot (данные)
```

Figure – это весь холст целиком. Он содержит глобальные настройки.

Axes – непосредственно сам график (не оси). Может быть построено несколько графиков на одном холсте

Пример 1

Построим график `y = sin(x)` на отрезке `[0, 10]`

```

x = np.linspace(0, 10, 100)
y = np.sin(x)

fig, ax = plt.subplots() # fig – холст, ax – график
ax.plot(x, y)           # На ax рисуем график по точкам (x, y)

plt.show()              # Отображаем все построения

```

Задание 1

Постройте графики следующих функций

1. `cos(x)` на отрезке `[-2*pi, 2*pi]`
2. `x^2` на отрезке `[-10, 10]`
3. `sqrt(x)` на отрезке `[0, 10]`

3. Холст

Параметры холста figure

Параметр	Назначение
<code>figsize</code>	Размер картинки. Измеряется в дюймах (inches)
<code>dpi</code>	Разрешение
<code>facecolor</code>	Цвет фона
<code>edgecolor</code>	Цвет рамки
<code>num</code>	Имя / номер холста

Пример 2

```

fig, ax = plt.subplots(
    figsize=(8, 6),
    dpi=100,
    facecolor='lightgray',
    edgecolor='black',
    num='My Plot'
)

```

Цвета

Однобуквенные цвета (старый стиль)

Код	Цвет
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Имена цветов, RGBA и HEX

Название	RGBA	HEX
red	(1.0, 0.0, 0.0, 1.0)	#FF0000
green	(0.0, 1.0, 0.0, 1.0)	#00FF00
blue	(0.0, 0.0, 1.0, 1.0)	#0000FF
black	(0.0, 0.0, 0.0, 1.0)	#000000
white	(1.0, 1.0, 1.0, 1.0)	#FFFFFF
yellow	(1.0, 1.0, 0.0, 1.0)	#FFFF00
cyan	(0.0, 1.0, 1.0, 1.0)	#00FFFF
magenta	(1.0, 0.0, 1.0, 1.0)	#FF00FF
gray	(0.5, 0.5, 0.5, 1.0)	#808080
orange	(1.0, 0.65, 0.0, 1.0)	#FFA500
purple	(0.5, 0.0, 0.5, 1.0)	#800080
pink	(1.0, 0.75, 0.8, 1.0)	#FFC0CB
brown	(0.6, 0.3, 0.0, 1.0)	#993300
lime	(0.0, 1.0, 0.0, 1.0)	#00FF00
navy	(0.0, 0.0, 0.5, 1.0)	#000080
teal	(0.0, 0.5, 0.5, 1.0)	#008080

RGBA – Red, Green, Blue, Alpha. Alpha – это прозрачность цвета

HEX – это способ записи цвета в виде шестнадцатеричного числа (часто используется в

вебе)

Несколько графиков на одном холсте

```
fig, axs = plt.subplots(2, 2)    # Создаем сетку 2x2

axs[0, 0].plot(x1, y1)
axs[0, 1].plot(x2, y2)
axs[1, 0].plot(x3, y3)
axs[1, 1].plot(x4, y4)

plt.show()
```

Задание 2

1. Создайте холст размера (10, 8)
2. Назовите его *Math plots*
3. Измените цвет фона на ваше усмотрение
4. На одном холсте изобразите четыре графика: `sin(x)`, `cos(x)`, `x^2` и `sqrt(x)`

4. Графики

Заголовок и оси

Добавление заголовка и подписей к осям

```
ax.set_title('График')
ax.set_xlabel('X')
ax.set_ylabel('Y')
```

Ограничения на осях

```
ax.set_xlim(0, 10)
ax.set_ylim(-1, 1)
```

Задание 3

1. Добавьте подписи осей и названия всех графиков на вашем холсте
2. Также на ваше усмотрение добавьте ограничения на оси. Как изменились графики?

Дополнительные параметры графиков

Сетка

- `ax.grid(True)`
- `ax.grid(color='gray', linestyle='--', alpha=0.5)`

Легенда

```
ax.plot(x, y, label="sin(x)")
ax.legend()
```

Фон графика

- `ax.set_facecolor('lightgray')`

Масштаб

- `ax.set_aspect('equal')` – делает оси одинакового масштаба

5. Линейный график `plot()`

Настройка внешнего вида

```
ax.plot(x, y,
        color='red',
        linestyle='--',
        marker='o',
        linewidth=2
    )
plt.show()
```

- `color` – цвет
- `linestyle` – стили линий
- `marker` – маркер на отдельных точках
- `linewidth` – толщина линии

Стиль линии	Название	Пример
<code>'-'</code>	сплошная линия	————
<code>'--'</code>	пунктир	- - -
<code>'-.'</code>	штрих-пунктир	— · — ·
<code>':'</code>	точечная

Маркер	Вид	Описание
<code>'o'</code>	●	круг
<code>'.'</code>	•	маленькая точка

Маркер	Вид	Описание
'.'	,	пиксель
'x'	×	крест
'+'	+	плюс
's'	■	квадрат
'^'	▲	треугольник вверх
'v'	▼	вниз
'D'	◆	ромб
'h'	●	шестиугольник

Построение нескольких линий

```
ax.plot(x1, y1)
ax.plot(x2, y2)
```

Задание 4

1. Создайте новый холст с одним графиком
2. На графике нарисуйте две линии:
 - Первая – синусоида, синего цвета маркер в виде ромба
 - Вторая – синусоида с добавленным шумом, оранжевая пунктирная линия
Добавление шума:

```
noise = np.random.normal(0, 0.2, size=100)
y_noisy = y + noise
```

6. Точечный график scatter()

Scatter – это метод для построения **точечного графика**, где каждая точка отображается отдельно и не соединяется линиями.

Он используется, когда важно увидеть **распределение данных и связь между переменными**

```
x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y)
plt.show()
```

Параметры

- `color` – цвет
- `s` – размер точек
- `alpha` – прозрачность
- `marker` – маркер

Задание 5

1. Измените созданный холст так, чтобы можно было рисовать три графика. Первый график оставьте неизменным
2. На втором графике постройте синусоиду и шум, но на этот раз шум должен быть в виде точек
3. Создайте три `numpy` массива. Первый – координаты x , второй – координаты y , третий – размеры точек. Постройте точечный график с различными размерами точек

7. Столбчатая диаграмма

`bar()` – вертикальные столбцы

```
categories = ['A', 'B', 'C']
values = [10, 20, 15]

ax.bar(categories, values)
plt.show()
```

`barh()` – горизонтальные столбцы

```
ax.barh(categories, values)
plt.show()
```

Параметры

- `ax.bar(categories, values, width=0.5)` – ширина для вертикальных столбцов
- `ax.barh(categories, values, height=0.5)` – высота для горизонтальных столбцов
- `color` – цвет столбцов
- `edgecolor` – цвет обводки столбцов

Слоеные столбчатые диаграммы

- Параметр `bottom` задает с какого уровня по оси Y начнется столбец

```
categories = ['A', 'B', 'C']

team1 = [3, 5, 2]
team2 = [4, 2, 6]

ax.bar(categories, team1, label='Team 1')
ax.bar(categories, team2, bottom=team1, label='Team 2')
ax.legend()

plt.show()
```

Задание 6

Построить горизонтальную диаграмму для `products = ['A', 'B', 'C']` и `sales = [5, 12, 9]`

Задание 7

Построить слоеную диаграмму "Распределение рабочего дня". Воспользуемся данными, сколько каждый студент тратит времени на разные занятия

```
students = ['Alex', 'Maria', 'John', 'Elena']

study = [5, 3, 4, 6]
work = [2, 4, 3, 1]
rest = [7, 5, 7, 7]
```

Построить **stacked bar chart**, где:

- каждый студент — это один столбец
- внутри столбца:
 - нижний слой → учёба
 - средний → работа
 - верхний → отдых

8. 3D графики

Чтобы подключить 3D-визуализацию в графике необходимо добавить параметр `projection=3d`

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
```

- 111 означает, что 1 строка, 1 столбец и 1-ый график

Необходимо сгенерировать 100 случайных точек:

- $x, y, z \in [0, 1]$

И построить:

- 3D scatter plot (`scatter3D`)
- Прозрачность (`alpha=0.5`)
- Точки должны быть разного (случайного размера)

Задание 9

Построить трехмерную функцию (`plot_surface()`)

$$z = \sin(x^2 + y^2)$$