

# Задание 2. Работа с цветами в облаках точек с использованием Python, NumPy, Matplotlib и Plotly

Тема: Работа с цветами в облаках точек с использованием Python, NumPy, Matplotlib и Plotly

---

## Цель работы:

Изучить методы визуализации облаков точек с цветовой информацией с помощью **Matplotlib** и **Plotly** в языке **Python**, научиться использовать координаты и другие параметры точек для задания цвета, а также сравнить 2D и 3D визуализацию.

---

## Оборудование и ПО:

- Python 3.8+
- Установленные библиотеки:

```
pip install numpy matplotlib plotly
```

---

## ЗАДАНИЕ

### 1. Генерация облака точек

Создайте облако точек размером  $N = 1000$ , равномерно распределённое в пределах куба  $[0, 1]^3$ :

```
import numpy as np

N = 1000
points = np.random.rand(N, 3) # X, Y, Z
```

---

## 2. Назначение цветов

Задайте цвета точкам по следующему принципу:

- Красный (R) компонент = координата X
- Зелёный (G) компонент = координата Y
- Синий (B) компонент = координата Z

```
colors = points.copy() # цвета нормализованы в диапазоне [0, 1]
```

---

## 3. Визуализация в 2D (Matplotlib)

Отобразите облако точек в 2D-проекции (XY), используя цвета точек:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 6))
plt.scatter(points[:, 0], points[:, 1], c=colors, s=10)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("2D визуализация облака точек (Matplotlib)")
plt.grid(True)
plt.show()
```

---

## 4. Визуализация в 3D (Plotly)

Сделайте интерактивную 3D-визуализацию с помощью Plotly:

```
import plotly.graph_objects as go

fig = go.Figure(data=[go.Scatter3d(
    x=points[:, 0],
    y=points[:, 1],
    z=points[:, 2],
    mode='markers',
    marker=dict(
        size=3,
        color=['rgb({}, {}, {})'.format(int(r*255), int(g*255), int(b*255))
        for r, g, b in colors],
    )
)])
fig.update_layout(title='3D визуализация облака точек (Plotly)',
```

```
scene=dict(  
    xaxis_title='X',  
    yaxis_title='Y',  
    zaxis_title='Z'))  
  
fig.show()
```

---

## 5. Дополнительное задание (по желанию)

Создайте альтернативную схему раскраски:

- Используйте расстояние от центра куба (0.5, 0.5, 0.5) для задания цвета
- Пример:

```
center = np.array([0.5, 0.5, 0.5])  
dist = np.linalg.norm(points - center, axis=1)  
norm_dist = (dist - dist.min()) / (dist.max() - dist.min()) #  
нормализация  
colors_alt = plt.cm.viridis(norm_dist)[: , :3] # RGB из colormap
```

---

## ТРЕБОВАНИЯ К ОТЧЁТУ

Отчёт оформляется в текстовом редакторе (MS Word, LibreOffice, PDF и др.) и должен содержать:

### 1. Титульный лист:

- Название работы
- ФИО, группа, дисциплина
- Преподаватель
- Дата выполнения

### 2. Цель и задачи работы (в формулировке студента)

### 3. Теоретическая часть:

- Что такое облако точек?
- Как представляется цвет в RGB?
- Отличие визуализации в Matplotlib и Plotly

### 4. Ход работы:

- Описание шагов (пункты 1–5)
- **Исходный код** с комментариями
- **Скриншоты:**
  - 2D-графика Matplotlib
  - 3D-графика Plotly (с указанием камеры/ракурса)
  - (опционально) альтернативной цветовой схемы

## 5. Ответы на вопросы:

## 6. Выводы:

- Что нового узнали?
  - В чём преимущества интерактивной визуализации?
  - Где можно использовать такие методы на практике?
- 

## Контрольные вопросы:

1. Почему для RGB значения используются числа от 0 до 1 или от 0 до 255?
  2. Как можно задать цвет точек на основе скалярной величины?
  3. Чем отличается статичная визуализация (Matplotlib) от интерактивной (Plotly)?
  4. Можно ли использовать другую цветовую карту (colormap) в Matplotlib? Как?
  5. Что произойдёт, если не нормализовать значения при генерации цветов?
- 

Вот подборка **вопросов для самоподготовки** по теме "**Цвета (Colors) в облаках точек на Python с использованием NumPy, Matplotlib и Plotly**". Эти вопросы помогут закрепить теорию и практику по работе с визуализацией точек с использованием цветовой информации.

---

## Вопросы для самоподготовки

### Общие понятия

1. Что такое облако точек и какие данные оно обычно содержит?
2. Как представляется цвет в модели RGB?
3. В каком диапазоне должны находиться значения RGB для визуализации?
4. Что такое нормализация значений и зачем она нужна при работе с цветами?

5. Чем отличается абсолютный цвет от вычисляемого (например, от координаты или расстояния)?
- 

## Работа с NumPy и цветами

6. Как с помощью NumPy можно задать цвет каждой точке на основе её координаты?
  7. Как вычислить расстояние от точки до центра пространства с помощью NumPy?
  8. Как задать градиент цвета по оси X, Y или Z?
  9. Как преобразовать массив `float` значений в нормализованный массив для `colormap`?
- 

## Matplotlib и цветовая визуализация

10. Как используется параметр `c=` в функции `plt.scatter()` ?
  11. Что такое `colormap` в Matplotlib и как его применить к облаку точек?
  12. Какие бывают цветовые карты ( `colormaps` ) и как выбрать подходящую?
  13. Как отобразить цветовую шкалу ( `legend` ) на графике?
- 

## Plotly и интерактивность

14. Как создаётся 3D-график с помощью `plotly.graph_objects` ?
  15. Как задать цвет каждой точки вручную в формате `'rgb(R,G,B)'` в Plotly?
  16. В чём преимущества интерактивной визуализации по сравнению со статичной?
  17. Какие параметры можно задать в Plotly для изменения внешнего вида маркеров?
- 

## Дополнительно — сохранение и экспорт

18. Как можно сохранить облако точек с цветами в файл (например, `.ply` , `.csv` )?
  19. Как можно использовать цвет как способ отображения дополнительных данных (температура, интенсивность, категория и т.д.)?
  20. Что произойдёт, если задать некорректные значения RGB (например, больше 1.0 или отрицательные)?
-

## Развивающие (применение на практике)

21. Как можно применить цветовую визуализацию при анализе:

- Ландшафта
- Сканов зданий
- 3D-моделей
- Научных данных (например, температуры или давления)?

22. Как можно визуализировать разные классы точек (например, "земля", "растительность", "здания") с помощью цветов?

23. Что лучше выбрать для презентации результатов: Matplotlib или Plotly? Почему?