

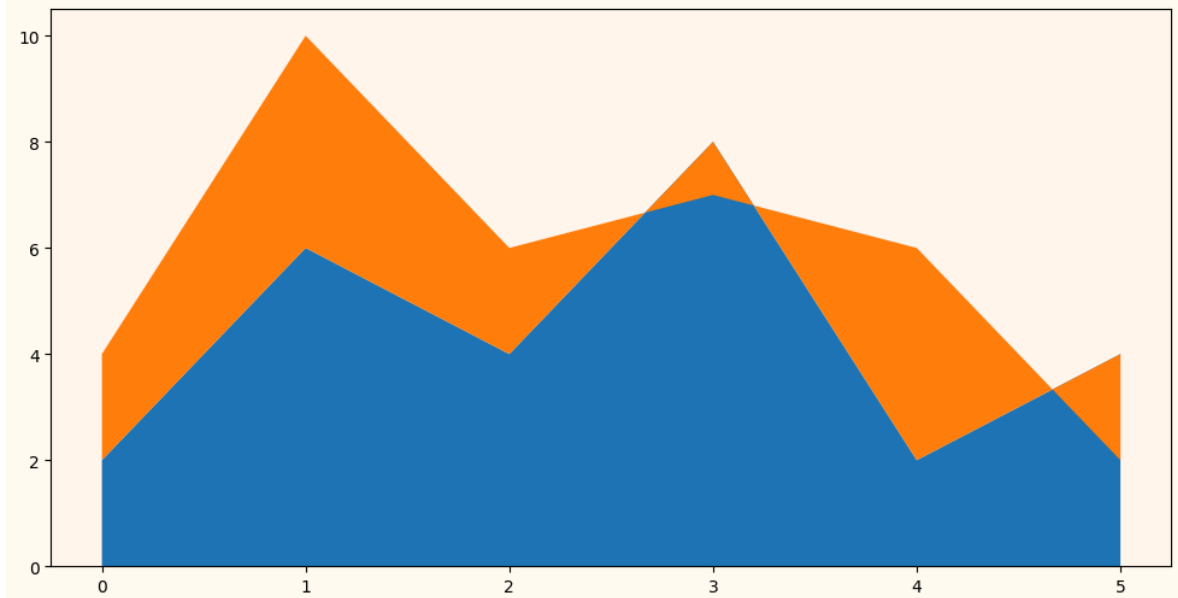
Задание 1: Stacked Area Chart

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

x = [0, 1, 2, 3, 4, 5]

y1 = [2, 6, 4, 8, 2, 4]
y2 = [2, 4, 2, -1, 4, -2]

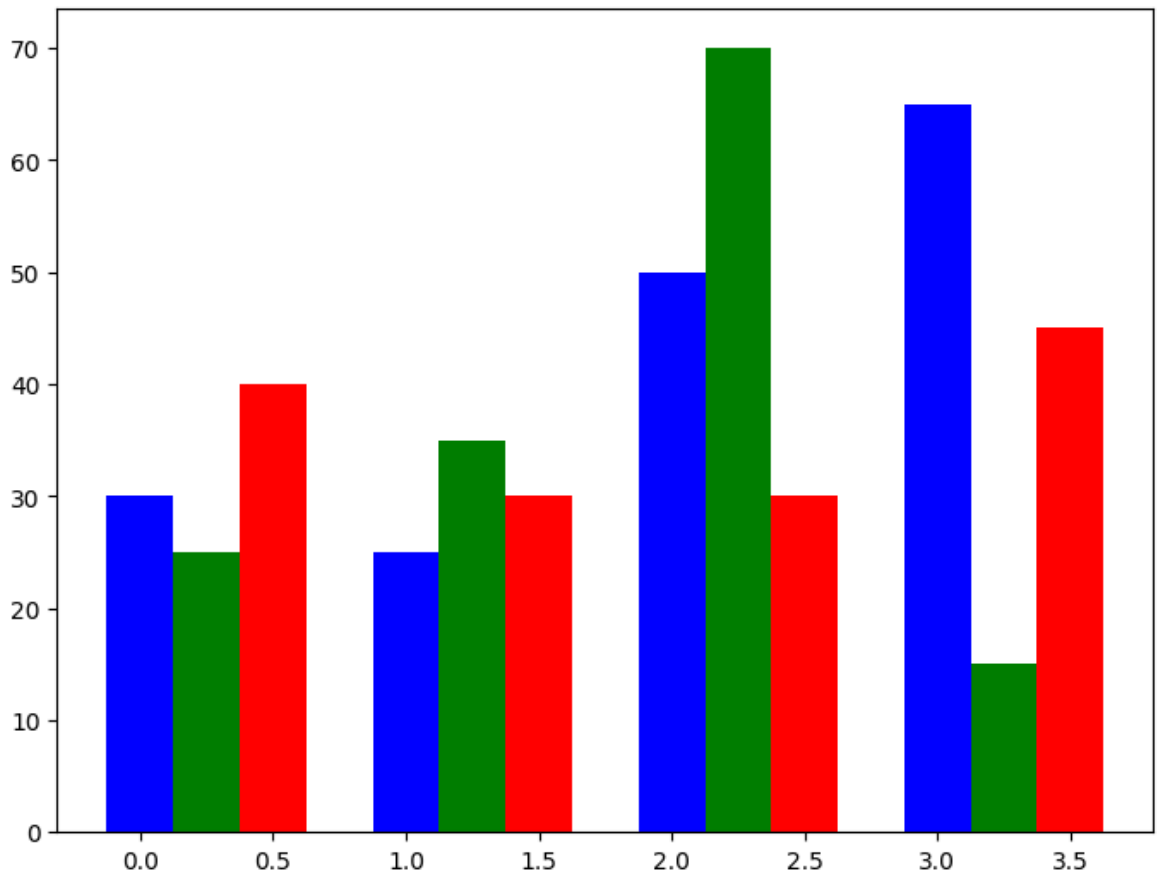
fig, ax = plt.subplots()
# Создаем наложенные области на основе созданных нами массивов:
ax.stackplot(x, [y1, y2])
# С помощью разных свойств мы можем изменить ширину, высоту, цвет и многое другое
fig.set_figwidth(12) # ширина и
fig.set_figheight(6) # высота "Figure"
fig.set_facecolor('floralwhite')
ax.set_facecolor('seashell')
# Демонстрируем полученную Figure.
plt.show()
```



2. Задание 2: Bar Chart

```
In [ ]: # Создаем массив для дальнейшей работы с ним
data = [[30, 25, 50, 65], [25, 35, 70, 15], [40, 30, 30, 45]]
# Создаем одномерный массив, используя библиотеку numpy
X = np.arange(4)
# Создаем область Figure, а после добавляем область Axes:
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])

ax.bar(X + 0.00, data[0], color = 'b', width = 0.25)
ax.bar(X + 0.25, data[1], color = 'g', width = 0.25)
ax.bar(X + 0.50, data[2], color = 'r', width = 0.25)
# Демонстрируем полученный Figure.
plt.show()
```



Задание 3: Histogram

```
In [ ]: import matplotlib

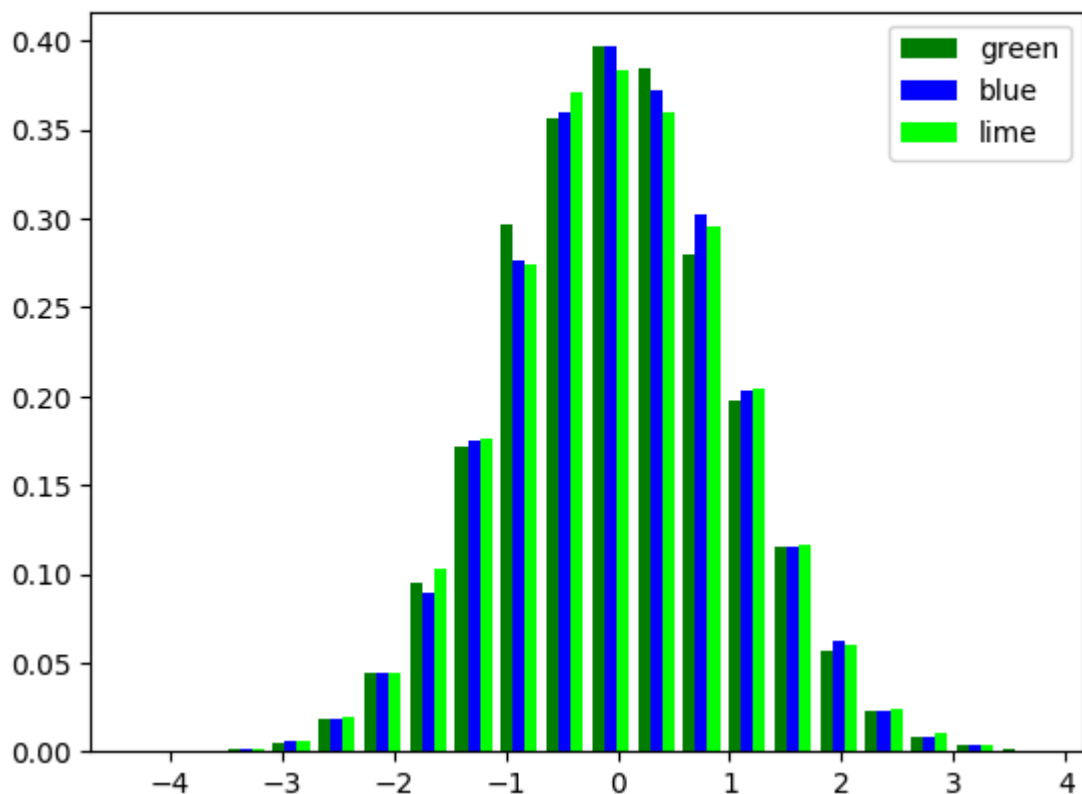
# Создаем данные для дальнейшей работы, используя библиотеку numpy
np.random.seed(10**7)
n_bins = 20
x = np.random.randn(10000, 3)

colors = ['green', 'blue', 'lime']

# Создаем гистограмму, сразу применяя к ней свойства лейбла и цвета
plt.hist(x, n_bins, density = True,
         histtype = 'bar',
         color = colors,
         label = colors)
# Изменяем размер легенд и название заголовка
plt.legend(prop = {'size': 10})

plt.title('matplotlib.pyplot.hist() function Example\n\n', fontweight = "bold")
# Демонстрируем полученный Figure.
plt.show()
```

matplotlib.pyplot.hist() function Example



4. Создайте гистограмму, которая будет отражать показатели посещаемости студентов вашей группы на лабораторных работах за последние 2 недели

Создайте гистограмму, которая будет отражать показатели bmi (минимальный, средний и максимальный) больных диабетом относительно их smoking_history

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Загрузка данных
data = pd.read_csv('diabetes_dataset.csv')

# Группировка данных по smoking_history
grouped = data.groupby('smoking_history')['blood_glucose_level']

# Вычисление средних и медианных значений
stats = grouped.agg(mean='mean', median='median').reset_index()

# Параметры диапазона оси Y
y_min = 120
y_max = 150

# Подготовка данных для гистограммы
categories = stats['smoking_history']
x = np.arange(len(categories)) # индексы для категорий
width = 0.35 # ширина столбцов

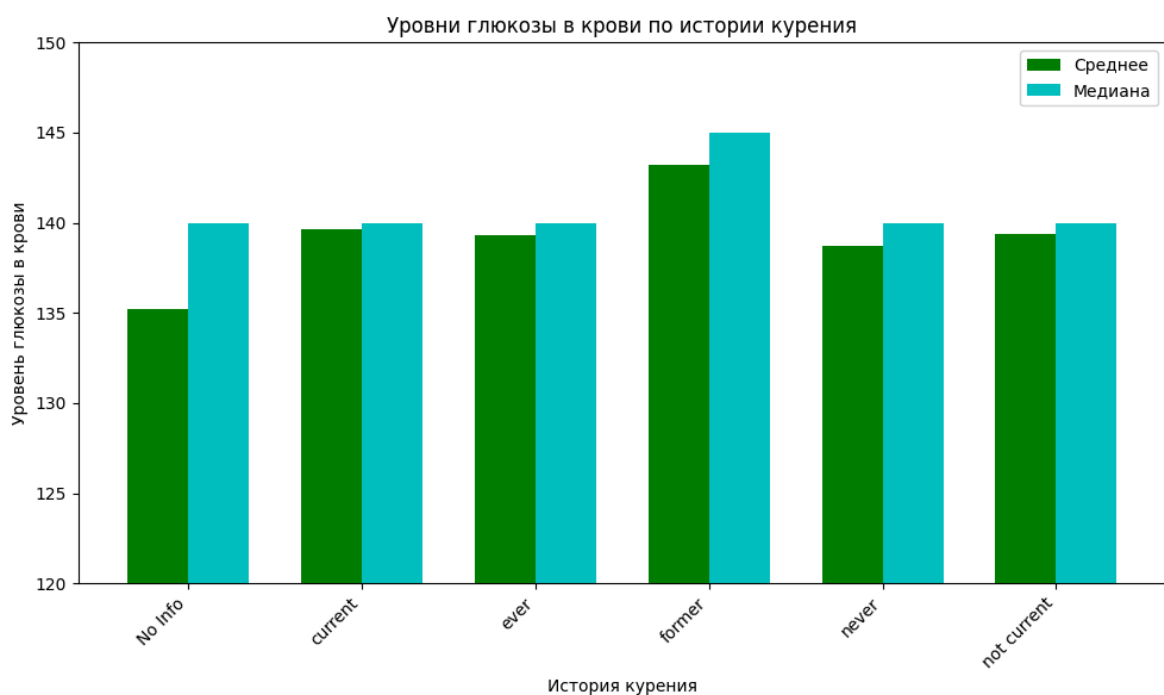
# Построение гистограммы
```

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x - width/2, stats['mean'], width, label='Среднее', color='g')
ax.bar(x + width/2, stats['median'], width, label='Медиана', color='c')

# Настройка осей и легенды
ax.set_xticks(x)
ax.set_xticklabels(categories, rotation=45, ha='right')
ax.set_xlabel('История курения')
ax.set_ylabel('Уровень глюкозы в крови')
ax.set_title('Уровни глюкозы в крови по истории курения')
ax.legend()

# Установка диапазона по оси Y
ax.set_ylim(y_min, y_max)

# Отображение графика
plt.tight_layout()
plt.show()
```



```
In [ ]: pd.read_csv('diabetes_dataset.csv')
```

Out[]:

	year	gender	age	location	race:AfricanAmerican	race:Asian	race:Caucasian
0	2020	Female	32.0	Alabama	0	0	0
1	2015	Female	29.0	Alabama	0	1	0
2	2015	Male	18.0	Alabama	0	0	0
3	2015	Male	41.0	Alabama	0	0	1
4	2016	Female	52.0	Alabama	1	0	0
...
99995	2018	Female	33.0	Wyoming	0	0	0
99996	2016	Female	80.0	Wyoming	0	1	0
99997	2018	Male	46.0	Wyoming	0	1	0
99998	2018	Female	51.0	Wyoming	1	0	0
99999	2016	Male	13.0	Wyoming	0	0	0

100000 rows × 16 columns



5. Создайте Bar Chart, которая будет отражать показатели посещаемости студентов вашей группы на лабораторных работах за последние 2 недели, добавьте в вывод что наиболее удобно для работы и почему

Создайте Bar Chart, которая будет отражать показатели bmi (минимальный, средний и максимальный) больных диабетом относительно их smoking_history

```
In [6]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Загрузка данных
data = pd.read_csv('diabetes_dataset.csv')

# Фильтрация только больных диабетом
diabetes_data = data[data['diabetes'] == 1]

# Группировка данных по smoking_history и расчет статистик для blood_glucose_level
grouped = diabetes_data.groupby('smoking_history')['bmi']
stats = grouped.agg(mean='mean', median='median').reset_index()

# Параметры для диапазона оси Y
y_min = 25 # Минимум для оси Y
y_max = 34 # Максимум для оси Y

# Подготовка данных для Bar Chart
categories = stats['smoking_history']
x = np.arange(len(categories)) # индексы для категорий
width = 0.4 # ширина столбцов

# Построение Bar Chart
```

```

fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x - width/2, stats['mean'], width, label='Среднее', color='limegreen')
ax.bar(x + width/2, stats['median'], width, label='Медиана', color='cyan')

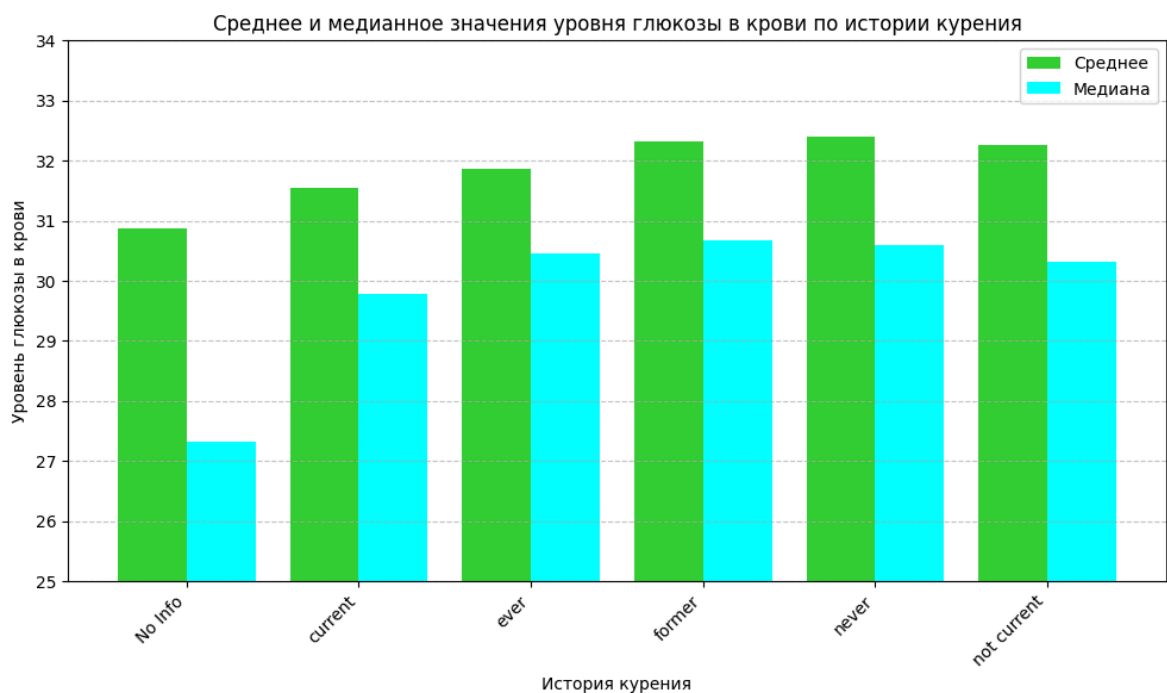
# Настройка осей и легенды
ax.set_xticks(x)
ax.set_xticklabels(categories, rotation=45, ha='right')
ax.set_xlabel('История курения')
ax.set_ylabel('Уровень глюкозы в крови')
ax.set_title('Среднее и медианное значения уровня глюкозы в крови по истории курения')
ax.legend()

# Установка диапазона оси Y
ax.set_ylim(y_min, y_max)

# Добавление сетки для удобства анализа
ax.grid(axis='y', linestyle='--', alpha=0.7)

# Отображение графика
plt.tight_layout()
plt.show()

```



Bar Chart — более удобный вариант для задачи, потому что: Он изначально предназначен для категориальных данных. Легко сравнивает показатели (минимум, среднее, максимум) по каждой категории. Визуально проще для восприятия при анализе данных. Гистограмма пригодится, если бы задача заключалась в анализе распределения уровней глюкозы в одной из категорий.

6. Нарисуйте 3 уровня лестницы с помощью Stacked Area Chart различными цветами, добавьте лейблы и название графика

```

In [ ]: import matplotlib.pyplot as plt
import numpy as np

# Определяем области по оси X и их высоте

```

```

x1 = np.arange(0, 2, 0.01) # Первая область (ширина от 0 до 2)
x2 = np.arange(2, 4, 0.01) # Вторая область (ширина от 2 до 4)
x3 = np.arange(4, 6, 0.01) # Третья область (ширина от 4 до 6)

# Определяем одинаковую высоту для всех уровней
y1 = np.ones_like(x1) * 1 # Высота 1 для первой области
y2 = np.ones_like(x2) * 2 # Высота 2 для второй области
y3 = np.ones_like(x3) * 3 # Высота 3 для третьей области

# Построение Stacked Area Chart
plt.fill_between(x1, 0, y1, color='skyblue', label='Уровень 1') # Первый уровень
plt.fill_between(x2, 0, y2, color='limegreen', label='Уровень 2') # Второй уровень
plt.fill_between(x3, 0, y3, color='salmon', label='Уровень 3') # Третий уровень

# Настройки графика
plt.xlabel('Ось X')
plt.ylabel('Ось Y')
plt.title('Stacked Area Chart в виде лестницы')
plt.legend(loc='upper left')

# Отображение графика
plt.tight_layout()
plt.show()

```

