

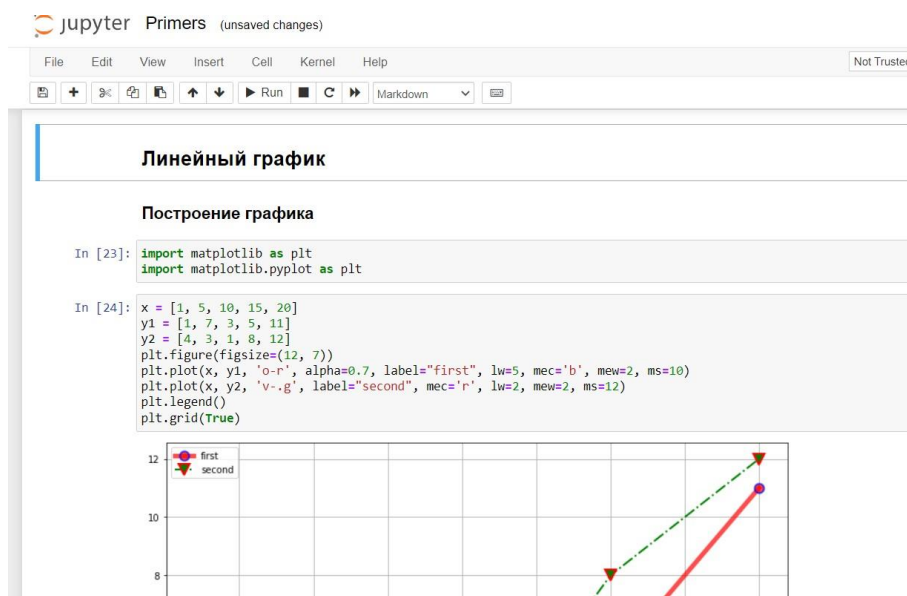
Ставрополь 2023

Цель работы: исследовать базовые возможности библиотеки matplotlib языка программирования Python.

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).
3. Выполните клонирование созданного репозитория на рабочий компьютер.
4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

6. Проработать примеры лабораторной работы в отдельном ноутбуке.



7. 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.

Условие. Даны два поезда, которые одновременно отправляются из пунктов А и В и движутся навстречу друг другу по одной железной дороге. Расстояние между пунктами А и В составляет 200 км. Скорость первого поезда составляет 80 км/ч, а скорость второго поезда - 60 км/ч. Определить расстояние от пункта А, на котором произойдет встреча поездов, и время этой встречи.

```

Ввод [5]: # Время и координаты для графика
time = np.linspace(0, 3, 100) # Промежуток времени (в часах)
position_train_A = speed_train_A * time # Координаты первого поезда (от пункта A)
position_train_B = distance_AB - speed_train_B * time # Координаты второго поезда (от пункта A)

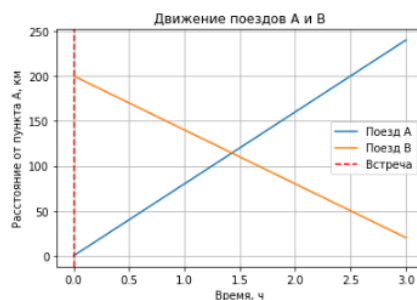
Ввод [6]: # Расчет расстояния и времени встречи
meeting_distance = (speed_train_A + speed_train_B) * time[0] # Расстояние на начальный момент времени
meeting_time = meeting_distance / (speed_train_A + speed_train_B) # Время встречи

Ввод [7]: # Вывод результатов
print("Расстояние от пункта A до места встречи:", meeting_distance, "км")
print("Время встречи:", meeting_time, "ч")

Расстояние от пункта A до места встречи: 0.0 км
Время встречи: 0.0 ч

Ввод [8]: # Построение графика
plt.plot(time, position_train_A, label='Поезд A')
plt.plot(time, position_train_B, label='Поезд B')
plt.axvline(x=meeting_time, color='r', linestyle='--', label='Встреча')
plt.xlabel('Время, ч')
plt.ylabel('Расстояние от пункта A, км')
plt.title('Движение поездов A и B')
plt.legend()
plt.grid(True)
plt.show()

```

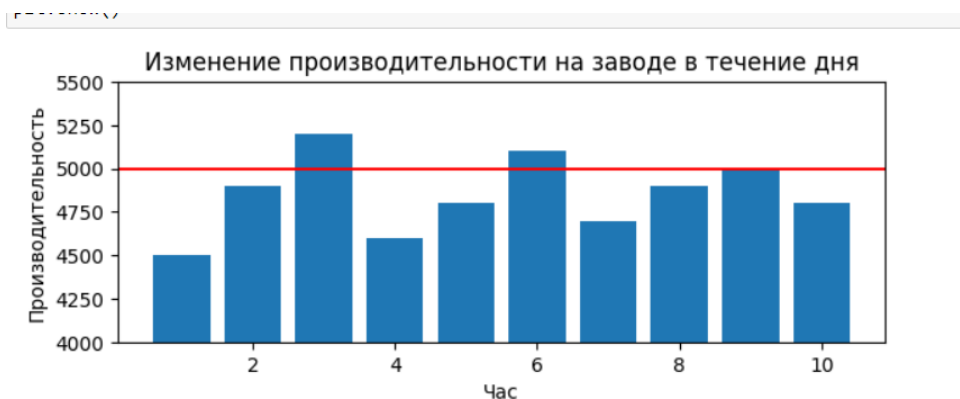


бук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

Условие. В среднем, в течение дня на заводе производится 5000 единиц продукции. Однако, каждый час происходят некоторые отклонения в производительности. Измерения производительности в 10 случайно выбранных часов за день представлены в таблице:

Часы	Производительность
1	4500
2	4900
3	5200
4	4600
5	4800
6	5100
7	4700
8	4900
9	5000
10	4800

Построим столбчатую диаграмму, отображающую изменение производительности в течение дня. Также построим линейную диаграмму, отображающую разброс значений производительности за каждый час.



9. 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.

Условие. В студенческой группе состоят студенты двух специальностей: "Информатика" и "Экономика". Студентов на специальности "Информатика" - 45%, а на специальности "Экономика" - 55%. Необходимо проанализировать соотношение студентов по специальностям и представить результаты в виде круговой диаграммы.

```
# Создание круговой диаграммы
colors = ['#1f77b4', '#ff7f0e']
plt.pie(students_count, labels=specialties, colors=colors, autopct='%1.1f%%',
plt.axis('equal')
plt.title('Соотношение студентов по специальностям')
plt.show()
```



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

Задание

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

```
Ввод [1]: import matplotlib.pyplot as plt

from PIL import Image
import requests

from io import BytesIO

response = requests.get('https://oir.mobi/uploads/posts/2021-05/1621520869_20-oir_mobi-p-koshki-milota-zhivotnie-krasivo-foto-20')
img = Image.open(BytesIO(response.content))

plt.imshow(img)
```

Out[1]: <matplotlib.image.AxesImage at 0x1fcc8439a30>



12. Выполните слияние ветки для разработки с веткой main (master).

13. Отправьте сделанные изменения на сервер GitHub.

Контрольные вопросы

1. Как выполнить построение линейного графика с помощью matplotlib?

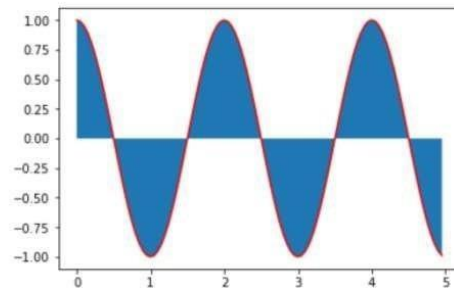
Для построения линейного графика используется функция `plot()`, со следующей сигнатурой:

```
plot([x], y, [fmt], *, data=None, **kwargs)
plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)
```

11. Э
афик
сиру
йте
дел
анн
ые
изме
нени
я в
репо
зито
рии.

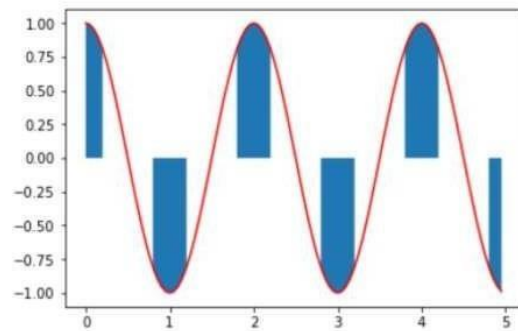
2. Как выполнить заливку области между графиком и осью? Между двумя графиками?

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



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

```
plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))
```

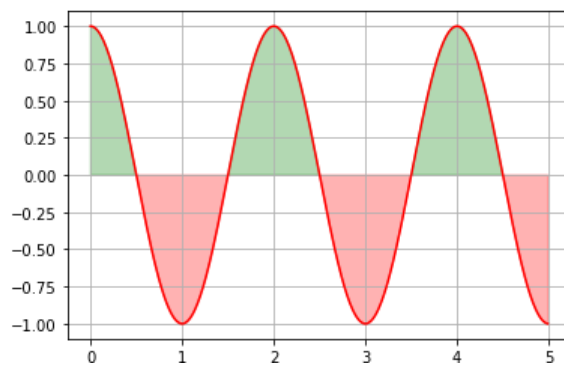


4. Как выполнить двухцветную заливку?

```
In [14]: 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)
```

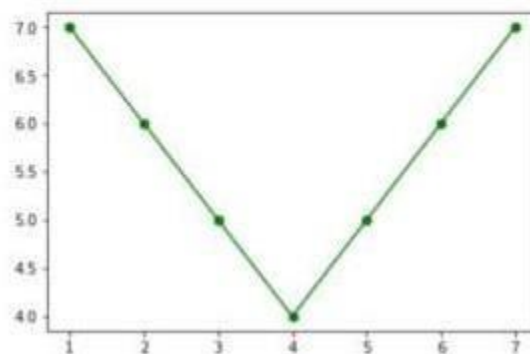
Out[14]: <matplotlib.collections.PolyCollection at 0x1b7583e04f0>



5. Как выполнить маркировку графиков?

```
x = [1, 2, 3, 4, 5, 6, 7]
y = [7, 6, 5, 4, 5, 6, 7]

plt.plot(x, y, marker="o", c="g")
```

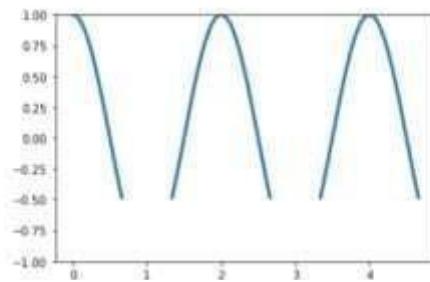


1. Как выполнить обрезку графиков?


```
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)
```

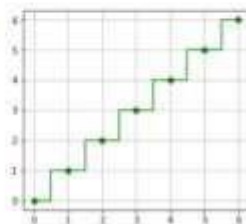
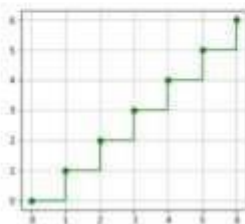
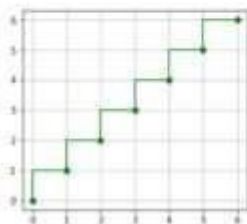


2. Как построить ступенчатый график? В чем особенность ступенчатого графика?

```
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()
```



3. 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)

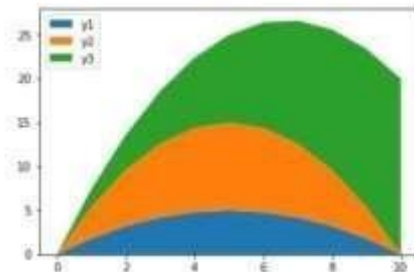
y1 = np.array([(i-0.2)**2+2*i for i in x])
y2 = np.array([(i-0.4)**2+4*i for i in x])
y3 = np.array([2*i for i in x])

labels = ["y1", "y2", "y3"]

fig, ax = plt.subplots()

ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')

```



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

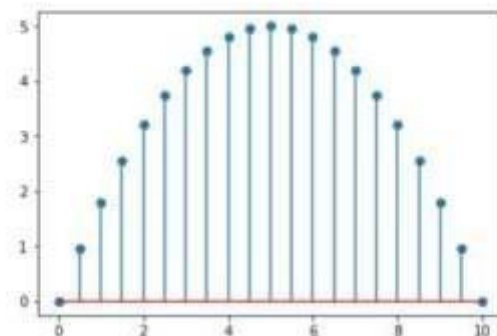
Визуально этот график выглядит как набор линий от точки с координатами (x,y) до базовой линии, в верхней точке ставится маркер.

```

x = np.arange(0, 10, 0.5, 0.5)
y = np.array([(i-0.2)**2+2*i for i in x])

plt.stem(x, y)

```

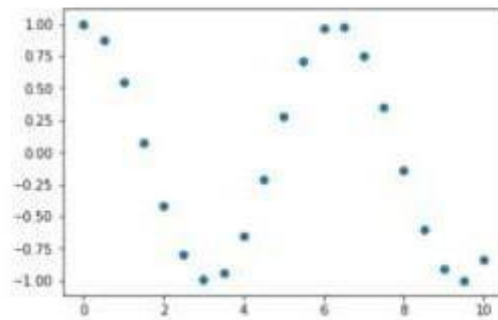


5. Как построить точечный график? В чем особенность точечного графика?

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)
```

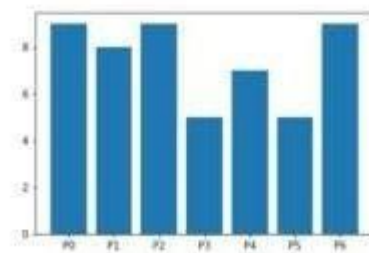


6. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

```
np.random.seed(123)

groups = ['P' * i for i in range(7)]
counts = np.random.randint(3, 10, len(groups))

plt.bar(groups, counts)
```



7. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с error bar элементом?

```

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

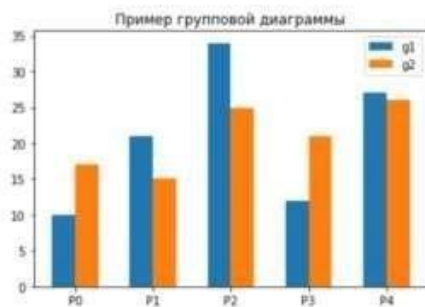
x = np.arange(len(cat_par))

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 хегг, уегг и еcolor (для задания цвета).

```

np.random.seed(123)

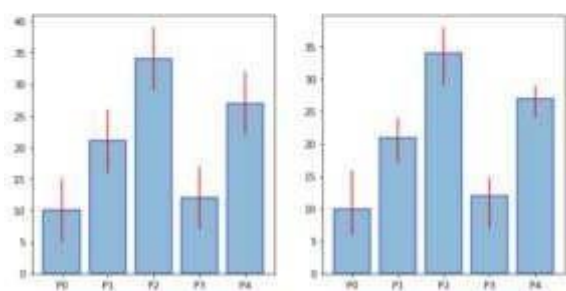
rnd = np.random.randint

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, axes = plt.subplots(1, 2, figsize=(10, 5))

axes[0].bar(cat_par, g1, yerr=5, ecolor="r", alpha=0.5, edgecolor="k",
linewidth=2)
axes[1].bar(cat_par, g1, yerr=error, ecolor="r", alpha=0.5, edgecolor="k",
linewidth=2)

```



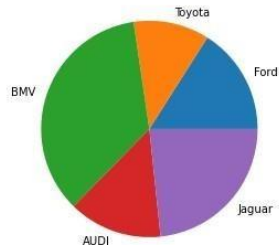
8. Как выполнить построение круговой диаграммы средствами matplotlib?

```
In [34]: vals = [24, 17, 53, 21, 35]

labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]

fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis("equal")
```

```
Out[34]: (-1.1163226287452406,
1.1007772680354877,
-1.1107362350259515,
1.1074836529113834)
```



9. 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?

Рассмотрим две функции для построения цветовой сетки: `imshow()` и `pcolormesh()`.

```
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))

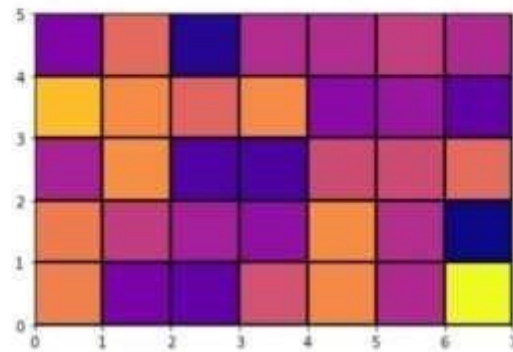
plt.imshow(img)
```

В результате получим изображение логотипа Matplotlib.



```
np.random.seed(123)

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



16. Как отобразить тепловую карту средствами matplotlib?

```
np.random.seed(123)

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