

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

Практическая часть:

1. 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).

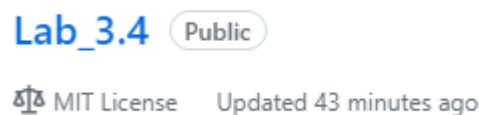


Рисунок 1. Создание репозитория

2. Выполните клонирование созданного репозитория на рабочий компьютер.

```
C:\Users\student-09-330\Downloads>git clone https://github.com/DinaDichenko/Lab_3.4.git
Cloning into 'Lab_3.4'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
Receiving objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
C:\Users\student-09-330\Downloads>
```

Рисунок 2. Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\student-09-330\Downloads\Lab_3.4>git checkout develop
Switched to branch 'develop'

C:\Users\student-09-330\Downloads\Lab_3.4>git flow init

Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] main

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fr
Bugfix branches? [bugfix/] bf
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] vet
Hooks and filters directory? [C:/Users/student-09-330/Downloads/Lab_3.4/.git/hooks] haf
```

Рисунок 3. Организация репозитория в соответствии с git-flow

4. Дополните файл `.gitignore` необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

```
.gitignore — Блокнот
Файл  Правка  Формат  Вид  Справка

.roperproject

# mkdocs documentation
/site

# mypy
.mypy_cache/
.dmyru.json
dmyru.json

# Pyre type checker
.pyre/

# pytype static type analyzer
.pytype/

# Cython debug symbols
cython_debug/

# PyCharm
# JetBrains specific template is maintained in a separate JetBrains.gitignore that can
# be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
# and can be added to the global gitignore or merged into this file.  For a more nuclear
# option (not recommended) you can uncomment the following to ignore the entire idea folder.
#.idea/
```

Рисунок 4. Изменение файла `.gitignore`

5. Проработать примеры лабораторной работы в отдельном ноутбуке.

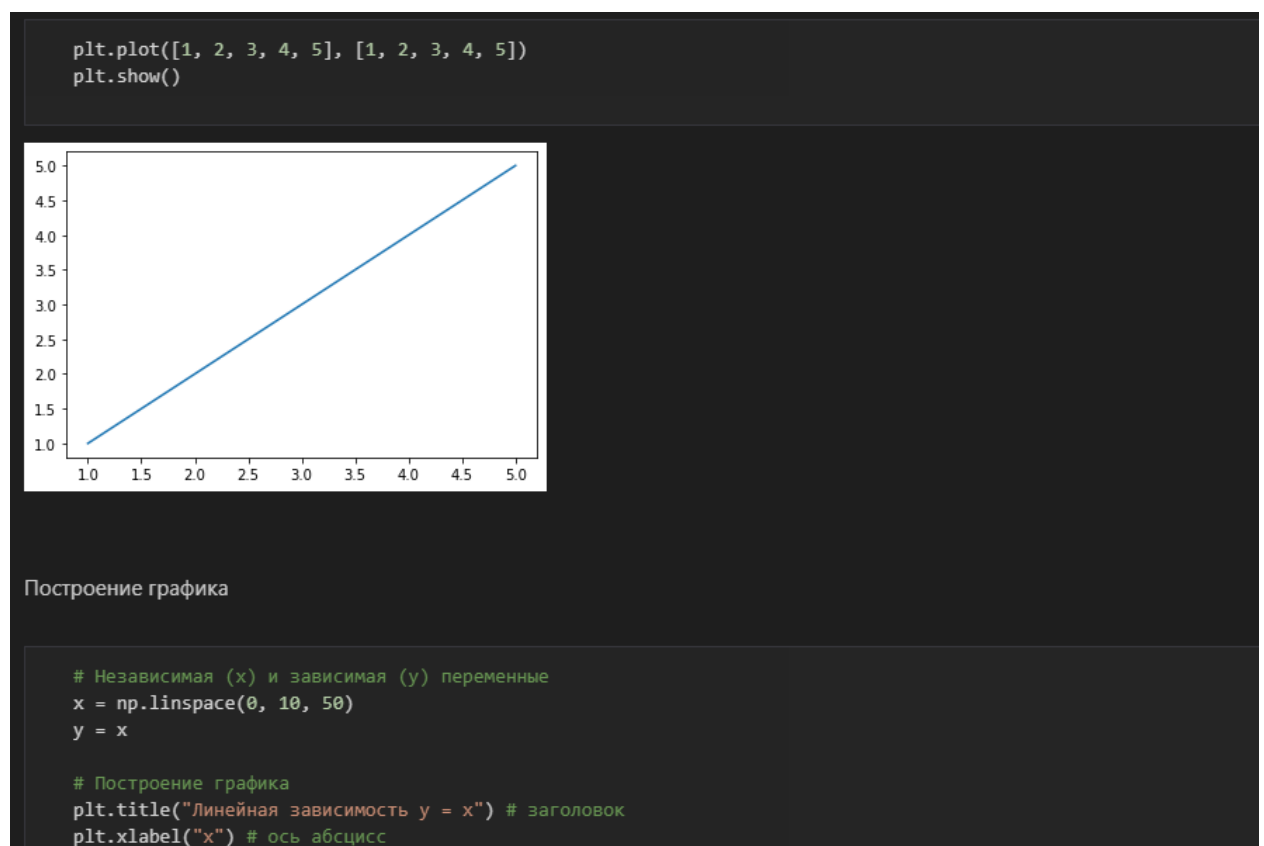


Рисунок 5. Проработка примеров

Контрольные вопросы:

1. Как осуществляется установка пакета `matplotlib`?

Существует два основных варианта установки этой библиотеки: в первом случае вы устанавливаете пакет `Anaconda`, в состав которого входит большое количество различных инструментов для работы в области машинного обучения и анализа данных (и не только); во втором – установить `Matplotlib` самостоятельно, используя менеджер пакетов.

2. Какая "магическая" команда должна присутствовать в ноутбуках `Jupyter` для корректного отображения графиков `matplotlib`?

```
%matplotlib inline
```

3. Как отобразить график с помощью функции `plot`?

```
plt.plot(x, y)
```

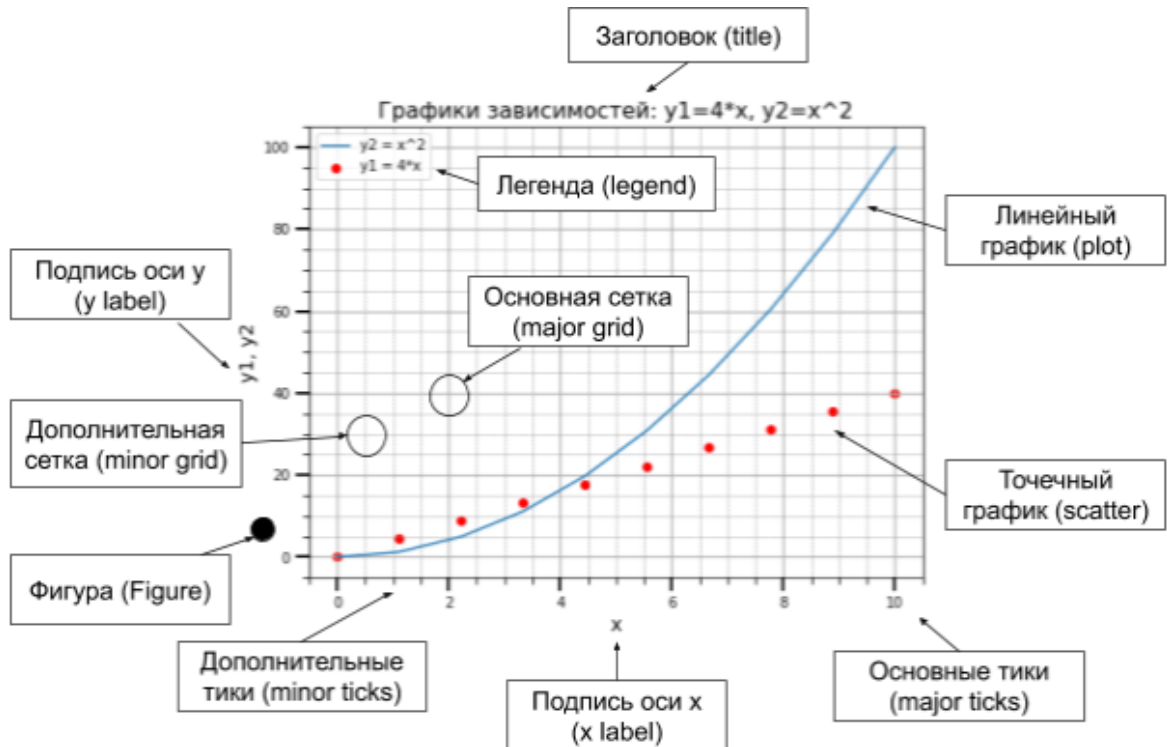
4. Как отобразить несколько графиков на одном поле?

В приведенном примере в функцию `plot()` последовательно передаются два массива для построения первого графика и два массива для построения второго, при этом, как вы можете заметить, для обоих графиков массив значений независимой переменной `x` один и тот же.

5. Какой метод Вам известен для построения диаграмм категориальных данных.

```
bar()
```

6. Какие основные элементы графика Вам известны?



7. Как осуществляется управление текстовыми надписями на графике?

xlabel, ylabel, title

8. Как осуществляется управление легендой графика?

legend()

9. Как задать цвет и стиль линий графика?

Задание цвета линии графика производится через параметр color (или c, если использовать сокращенный 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\(https://xkcd.com/color/rgb/\)](https://xkcd.com/color/rgb/), должен начинаться с префикса 'xkcd:'

- цвет из набора Tableau Color (палитра T10), должен начинаться с префикса 'tab:'

10. Как выполнить размещение графика в разных полях?

```
plt.subplot(2, 1, 1)
```

```
plt.plot(x, y1) # построение графика
```

```
plt.title("Зависимости:  $y_1 = x$ ,  $y_2 = x^2$ ") # заголовок
```

```
plt.ylabel("y1", fontsize=14) # ось ординат
```

```
plt.grid(True) # включение отображение сетки
```

```
plt.subplot(2, 1, 2)
```

```
plt.plot(x, y2) # построение графика
```

```
plt.xlabel("x", fontsize=14) # ось абсцисс
```

```
plt.ylabel("y2", fontsize=14)
```

Вывод: в результате выполнения работы были исследованы базовые возможности библиотеки matplotlib языка программирования Python.