

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение высшего образования**
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ

по лабораторной работе №3.5

Дисциплина: «Программирование на Python»

Тема: «Визуализация данных с помощью matplotlib»

Выполнила: студентка 2 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2023

Цель работы: исследовать базовые возможности визуализации данных на плоскости средствами 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).

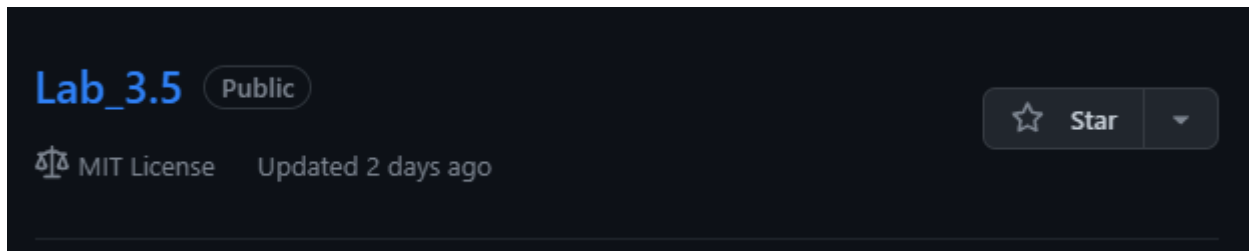


Рисунок 1. Создание репозитория

2. Выполните клонирование созданного репозитория на рабочий компьютер.

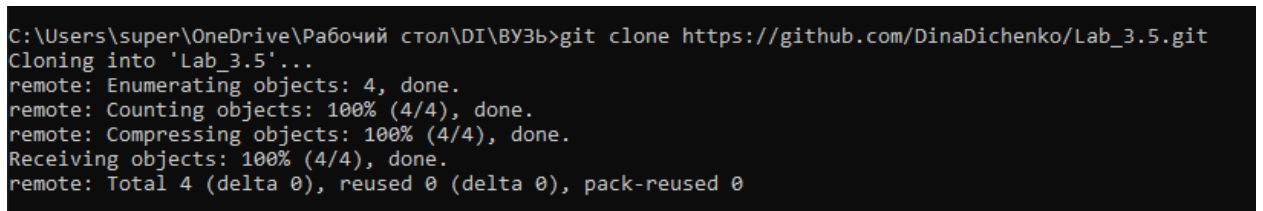


Рисунок 2. Клонирование репозитория

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

```

C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗ\Lab_3.5>git branch develop
C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗ\Lab_3.5>git checkout develop
Switched to branch 'develop'

C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗ\Lab_3.5>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/] f
Bugfix branches? [bugfix/] b
Release branches? [release/] r
Hotfix branches? [hotfix/] h
Support branches? [support/] s
Version tag prefix? [] v
Hooks and filters directory? [C:/Users/super/OneDrive/Рабочий стол/DI/ВУЗ/Lab_3.5/.git/hooks] hf

```

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
# mypy
.mypy_cache/
.dmypy.json
dmypy.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/

### Python Patch ###
# Poetry local configuration file - https://python-poetry.org/docs/configuration/#local-configuration
poetry.toml

```

Рисунок 4. Изменение файла .gitignore

5. Проработать примеры лабораторной работы в отдельном ноутбуке.

Линейный график

```
In [25]: import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import numpy as np
```

```
In [26]: x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [4, 3, 1, 8, 12]

plt.figure(figsize=(12, 7))
plt.plot(x, y1, 'o-r', alpha=0.7, label="first", lw=5, mec='b', mew=2, ms=10)
plt.plot(x, y2, 'v-.g', label="second", mec='r', lw=2, mew=2, ms=12)

plt.legend()
plt.grid(True)
```

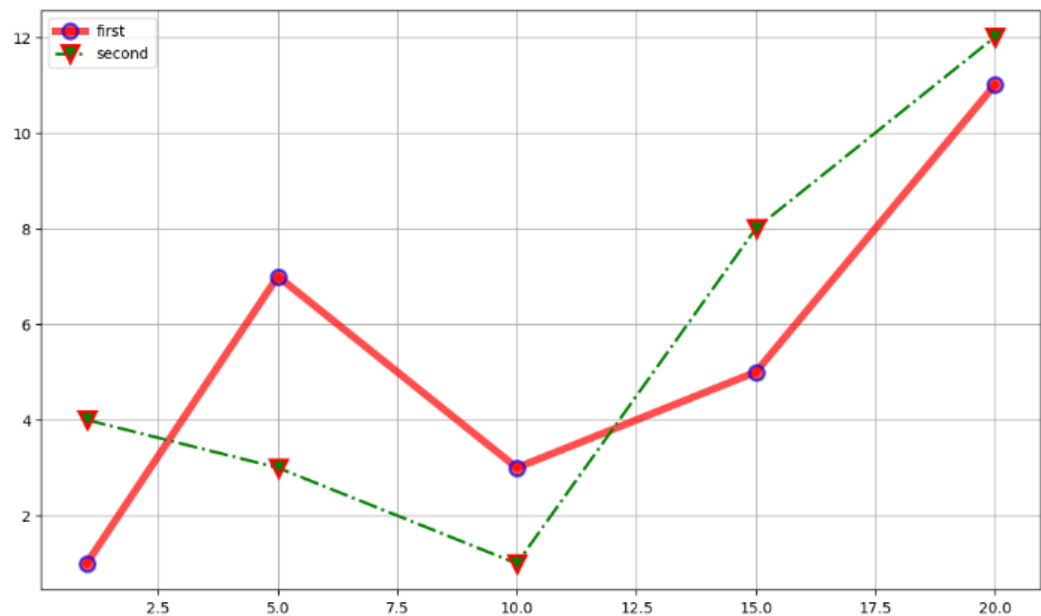


Рисунок 5. Проработка примеров

6. 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.

Построим график:

```
In [47]: plt.figure(figsize=(12, 7))
plt.plot(x, y1, c="c", label="$ x_1(t) = -4 + 2t, $" ,)
plt.plot(x, y2, c="r", label="$ x_2(t) = 8 - 1,5t $" )

plt.legend()
plt.grid(True)
plt.show()
```

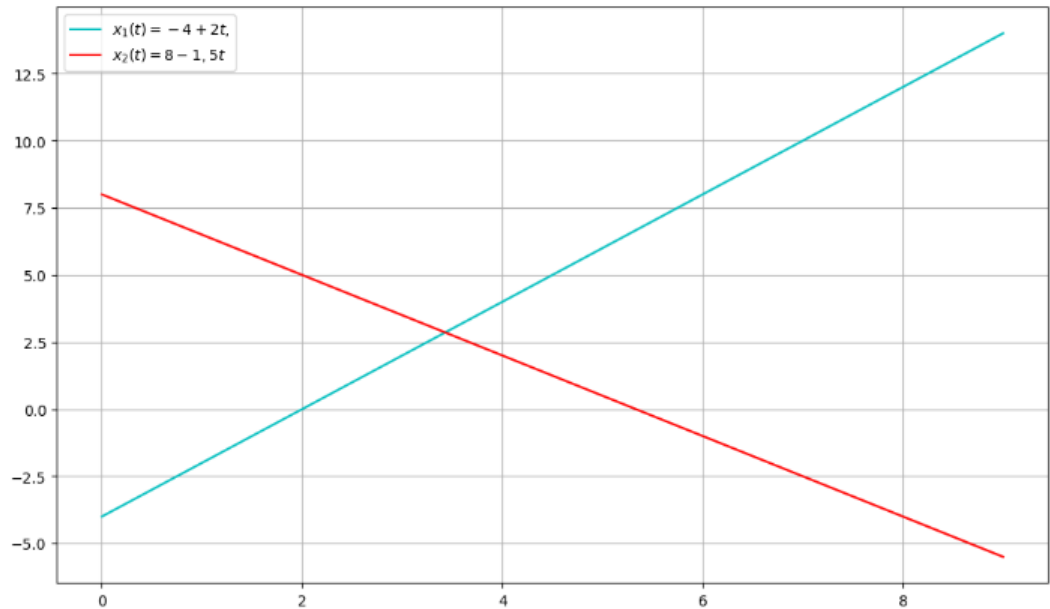


Рисунок 6. Линейный график

7. Создать ноутбук, 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.

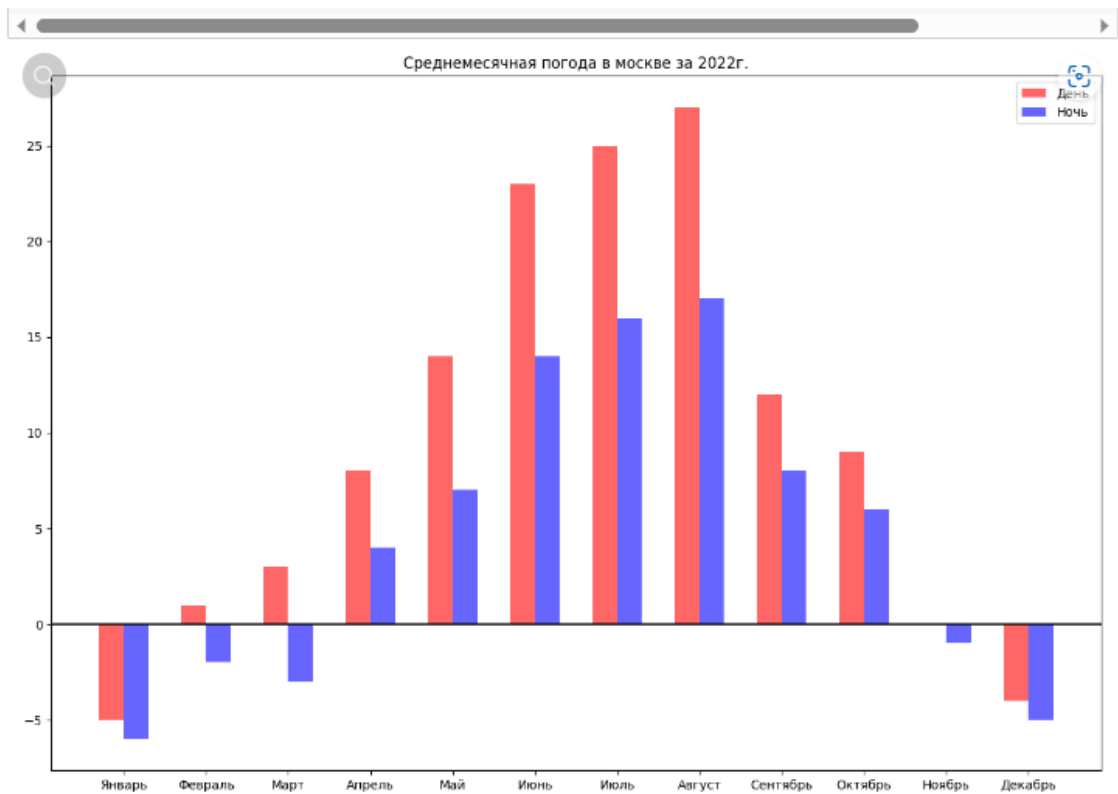


Рисунок 7. Столбчатая диаграмма

8. 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.

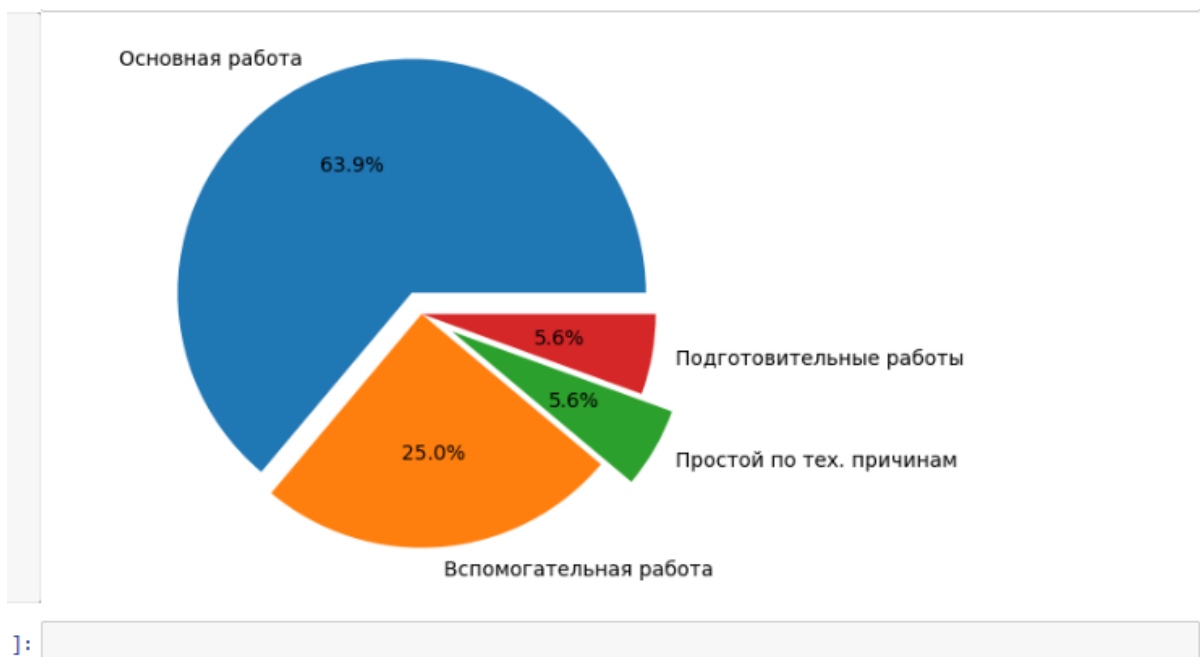


Рисунок 8. Круговая диаграмма

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

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

```
In [6]: import matplotlib.pyplot as plt
        from PIL import Image
        import requests
        from io import BytesIO

In [10]: response = requests.get('https://damion.club/uploads/posts/2022-02/1645738281_32-damion-club-p-arti-
img = Image.open(BytesIO(response.content))

ax = plt.gca ()
ax.get_xaxis ().set_visible ( False )
ax.get_yaxis ().set_visible ( False )

plt.imshow(img)
plt.show()
```



Рисунок 9. Вставка изображений

Контрольные вопросы:

1. Как выполнить построение линейного графика с помощью matplotlib?

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

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

fill_between().

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

```
fill_between(x, y1, y2=0, where=None, interpolate=False, step=None, *,
data=None, **kwargs)
```

4. Как выполнить двухцветную заливку?

```
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)
```

5. Как выполнить маркировку графиков?

marker="стиль маркера"

6. Как выполнить обрезку графиков?

Для того, чтобы отобразить только часть графика, которая отвечает определенному условию используйте предварительное маскирование данных с помощью функции `masked_where`

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

Такой график строится с помощью функции `step()`.

8. Как построить стековый график? В чем особенность стекового графика?

Для построения стекового графика используется функция `stackplot()`.

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

С помощью функции `stem()`.

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

Для отображения точечного графика предназначена функция `scatter()`.

11. Как осуществляется построение столбчатых диаграмм с помощью `matplotlib`?

Для визуализации категориальных данных хорошо подходят столбчатые диаграммы. Для их построения используются функции:

`bar()` – для построения вертикальной диаграммы

`barh()` – для построения горизонтальной диаграммы.

12. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с `errorbar` элементом?

`Errorbar` элемент позволяет задать величину ошибки для каждого элемента графика. Для этого используются параметры `xerr`, `yerr` и `ecolor` (для задания цвета):

13. Как выполнить построение круговой диаграммы средствами `matplotlib`?

Круговые диаграммы – это наглядный способ показать доли компонент в наборе. Они идеально подходят для отчетов, презентаций и т.п. Для построения круговых диаграмм в `Matplotlib` используется функция `pie()`.

14. 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.

Функции для построения цветовой сетки: `imshow()` и `pcolormesh()`.

15. Как отобразить изображение средствами `matplotlib`?

Основное назначение функции `imshow()` состоит в представлении 2d растров.

16. Как отобразить тепловую карту средствами `matplotlib`?

Рассмотрим ещё одну функцию для визуализации 2D наборов данных – `pcolormesh()`. В библиотеке `Matplotlib` есть ещё одна функция с аналогичным функционалом – `pcolor()`, в отличие от нее рассматриваемая нами `pcolormesh()` более быстрая и является лучшим вариантом в большинстве случаев. Функция `pcolormesh()` похожа по своим возможностям на `imshow()`, но есть и отличия.

Вывод: в результате выполнения работы были исследованы базовые возможности визуализации данных на плоскости средствами библиотеки `matplotlib` языка программирования Python.