

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №3.4

Основы работы с пакетом matplotlib.

по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-21-1

Лысенко И.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2023

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

Ход работы:

1. Создал репозиторий на GitHub:
https://github.com/IsSveshuD/lab_3.4.git .

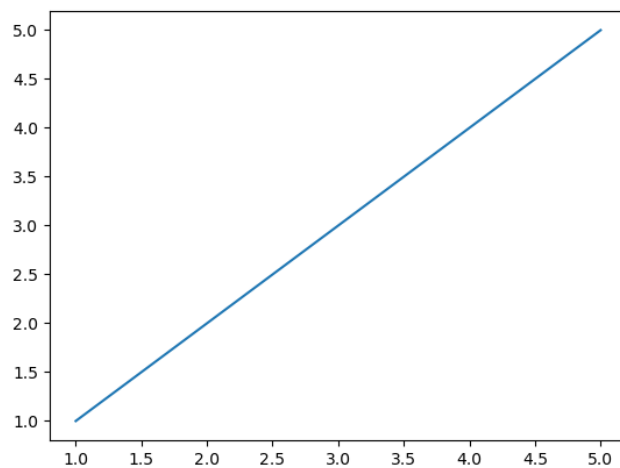
2. Проработал примеры:

Основы работы с пакетом matplotlib

```
In [9]: import matplotlib.pyplot as plt  
import numpy as np  
%matplotlib inline
```

Если вы работаете в Jupyter Notebook для того, чтобы получать графики рядом с ячейками с кодом необходимо выполнить специальную magic команду после того, как импортируете matplotlib.

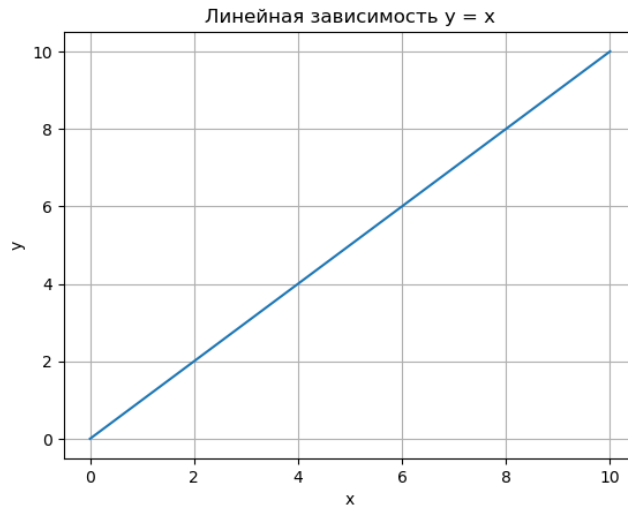
```
In [10]: plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5]);
```



Построение графика

```
In [12]: # Независимая (x) и зависимая (y) переменные
x = np.linspace(0, 10, 50)
y = x

# Построение графика
plt.title("Линейная зависимость y = x") # заголовок
plt.xlabel("x") # ось абсцисс
plt.ylabel("y") # ось ординат
plt.grid() # включение отображение сетки
plt.plot(x, y); # построение графика
```



Размещение графиков на разных полях

Существуют три основных подхода к размещению нескольких графиков на разных полях:

использование функции `subplot()` для указания места размещения поля с графиком;

использование функции `subplots()` для предварительного задания сетки, в которую будут укладываться поля;

использование `GridSpec`, для более гибкого задания геометрии размещения полей с графиками в сетке.

Работа с функцией `subplot()`

```
In [74]: # Исходный набор данных
x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [i*1.2 + 1 for i in y1]
y3 = [i*1.2 + 1 for i in y2]
y4 = [i*1.2 + 1 for i in y3]
```

```
In [75]: # Настройка размеров подложки
plt.figure(figsize=(12, 7))
```

```
Out[75]: <Figure size 1200x700 with 0 Axes>
```

```
<Figure size 1200x700 with 0 Axes>
```

```
In [77]: # Вывод графиков
plt.subplot(2, 2, 1)
plt.plot(x, y1, '-')

plt.subplot(2, 2, 2)
plt.plot(x, y2, '-.-')

plt.subplot(2, 2, 3)
plt.plot(x, y3, '-.-')

plt.subplot(2, 2, 4)
plt.plot(x, y4, ':');
```

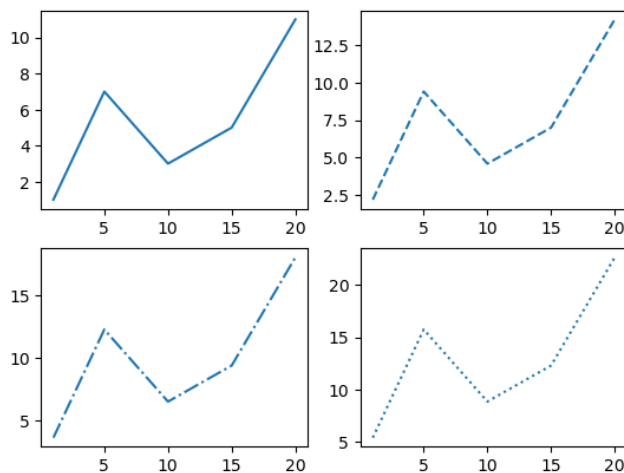


Рисунок 1 – Примеры

Ответы на 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.

Установка matplotlib через менеджер pip

Второй вариант – это воспользоваться менеджером pip и установить Matplotlib самостоятельно, для этого введите в командной строке вашей операционной системы следующие команды:

```
python -m pip install -U pip
python -m pip install -U matplotlib
```

Первая – обновит pip до последней версии. Вторая установит пакет matplotlib.

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

Если вы работаете в Jupyter Notebook для того, чтобы получать графики рядом с ячейками с кодом необходимо выполнить специальную magic команду после того, как импортируете matplotlib:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

3. Как отобразить график с помощью функции plot?

`plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])` – если работаем в ноутбуке и использовали магическую команду, то `plt.show()` писать не надо

`plt.show()` – для вывода в отдельном окне

4. Как отобразить несколько графиков на одном поле?

Построим несколько графиков на одном поле, для этого добавим квадратичную зависимость:

```
# Линейная зависимость
```

```

x = np.linspace(0, 10, 50)
y1 = x
# Квадратичная зависимость
y2 = [i**2 for i in x]
# Построение графика
plt.title("Зависимости: y1 = x, y2 = x^2") # заголовок
plt.xlabel("x") # ось абсцисс
plt.ylabel("y1, y2") # ось ординат
plt.grid() # включение отображение сетки
plt.plot(x, y1, x, y2) # построение графика

```

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

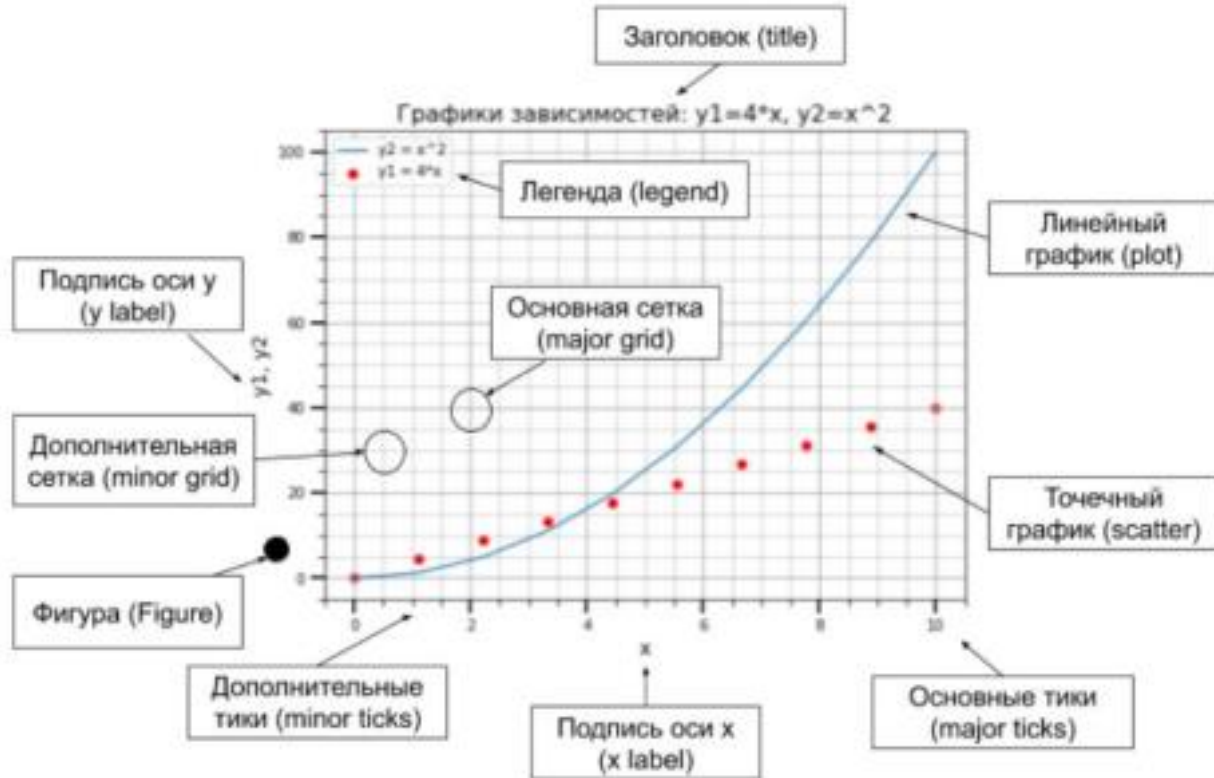
Метод: bar()

```
fruits = ["apple", "peach", "orange", "bannana", "melon"]
```

```
counts = [34, 25, 43, 31, 17]
```

```
plt.bar(fruits, counts)
```

6. Какие основные элементы графика Вам известны?



7. Как осуществляется управление текстовыми надписями на графике?

Наиболее часто используемые текстовые надписи на графике

это:

наименование осей;

наименование самого графика;

текстовое примечание на поле с графиком;

легенда.

Наименование осей

Для задания подписи оси x используется функция `xlabel()`, оси y – `ylabel()`. Разберемся с аргументами данных функций.

Функции `xlabel()/ylabel()` принимают в качестве аргументов параметры конструктора класса `matplotlib.text.Text`. Пример использования:

```
plt.plot([1, 5, 10, 15, 20], [1, 7, 3, 5, 11])
```

```
plt.xlabel('Day', fontsize=15, color='blue')
```

Заголовок графика

Для задания заголовка графика используется функция `title()`: `plt.title('Chart price', fontsize=17)`

Текстовое примечание

За размещение текста на поле графика отвечает функция `text()`, которой вначале передаются координаты позиции надписи, после этого – текст самой надписи.

```
plt.text(1, 1, 'type: Steel')
```

7. Как осуществляется управление легендой графика?

Легенда будет размещена на графике, если вызвать функцию `legend()`.

Параметр	Тип	Описание
<i>fontsize</i>	<i>int</i> или <i>float</i> или {'xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large'}	Размера шрифта надписи легенды
<i>frameon</i>	<i>bool</i>	Отображение рамки легенды
<i>framealpha</i>	<i>None</i> или <i>float</i>	Прозрачность легенды
<i>facecolor</i>	<i>None</i> или <i>str</i>	Цвет заливки
<i>edgecolor</i>	<i>None</i> или <i>str</i>	Цвет рамки
<i>title</i>	<i>None</i> или <i>str</i>	Текст заголовка
<i>title_fontsize</i>	<i>None</i> или <i>str</i>	Размер шрифта

Место расположения легенды определяется параметром *loc*, которое может принимать одно из следующих значений:

Строковое описание	Код
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8

8. Как задать цвет и стиль линий графика?

Стиль линии графика

Стиль линии графика задается через параметр *linestyle*, который может принимать значения из приведенной ниже таблицы.

Значение параметра	Описание
'-' или 'solid'	Непрерывная линия
'--' или 'dashed'	Штриховая линия
'-.' или 'dashdot'	Штрихпунктирная линия
'.' или 'dotted'	Пунктирная линия
'None' или '' или ''	Не отображать линию

Стиль линии можно передать сразу после указания списков с координатами без указания, что это параметр *linewidth*.

```
x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]

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

9. Как выполнить размещение графика в разных полях?

Работа с функцией `subplot()`

Самый простой способ представить графики в отдельных полях – это использовать функцию `subplot()` для задания их мест размещения. До этого момента мы не работали с Фигурой (*Figure*) напрямую, значения ее параметров, задаваемые по умолчанию, нас устраивали. Для решения текущей задачи придется один из параметров – размер подложки, задать вручную. За это отвечает аргумент `figsize` функции `figure()`, которому присваивается кортеж из двух *float* элементов, определяющих высоту и ширину подложки.

После задания размера, указывается местоположение, куда будет установлено поле с графиком с помощью функции `subplot()`. Чаще всего используют следующие варианты вызова `subplot`:

`subplot(nrows, ncols, index)`

- `nrows: int`
 - Количество строк.
- `ncols: int`
 - Количество столбцов.
- `index: int`
 - Местоположение элемента.

`subplot(pos)`

- `pos: int`
 - Позиция, в виде трехзначного числа, содержащего информацию о количестве строк, столбцов и индексе, например 212, означает подготовить разметку с двумя строками и одним столбцом, элемент вывести в первую позицию второй строки. Этот вариант можно использовать, если количество строк и столбцов сетки не более 10, в ином случае лучше обратиться к первому варианту.

Вывод: в результате выполнения лабораторной работы были получены необходимые теоретические сведения и практические навыки для работы с библиотекой `matplotlib` языка программирования Python.