

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Кафедра инфокоммуникаций**

**Отчет  
по лабораторной работе №11  
«Основы работы с пакетом matplotlib»  
по дисциплине:  
«Введение в системы искусственного интеллекта»**

Выполнил: студент группы ИВТ-б-о-18-1 (2)  
Криворучко Ярослав Евгеньевич

\_\_\_\_\_ (подпись)

Проверил:  
Воронкин Роман Александрович

\_\_\_\_\_ (подпись)

Ставрополь, 2022 г.

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

```
Ввод [3]: import matplotlib.pyplot as plt
%matplotlib inline
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
```

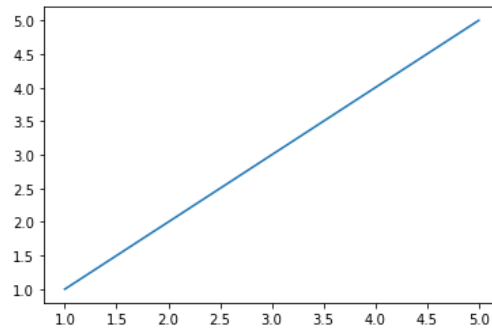
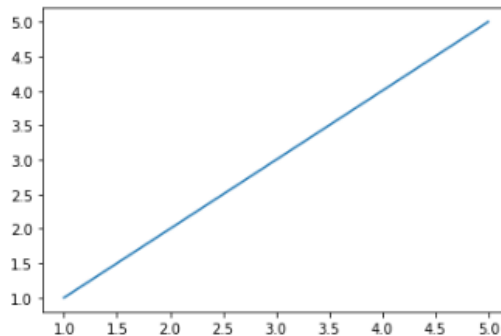


Рисунок 1 – Пример

```
Ввод [4]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
# 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
```



Out[4]: [<matplotlib.lines.Line2D at 0x256e1e78490>]

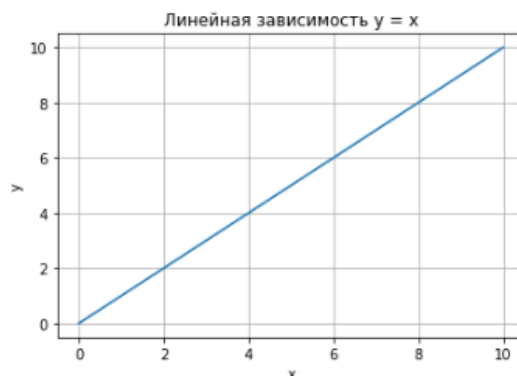
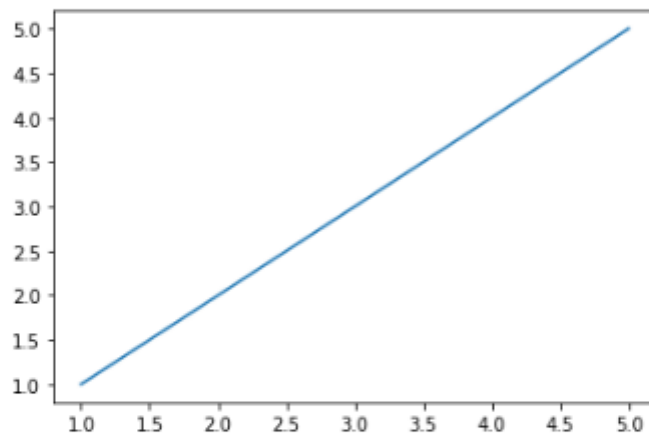


Рисунок 2 – Пример

```
вод [5]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
# 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, "r--") # построение графика
```



Out[5]: [

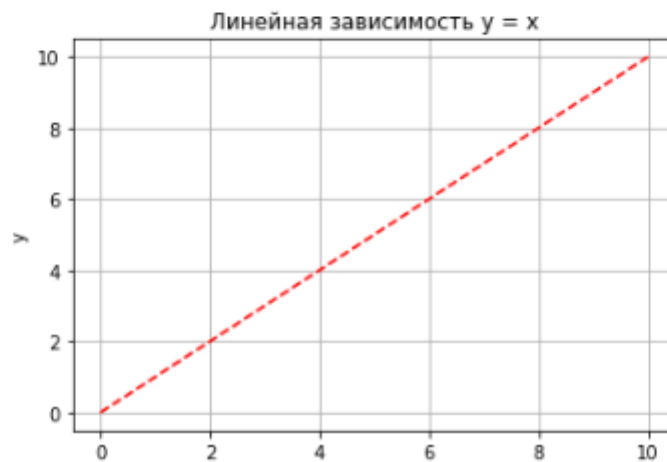


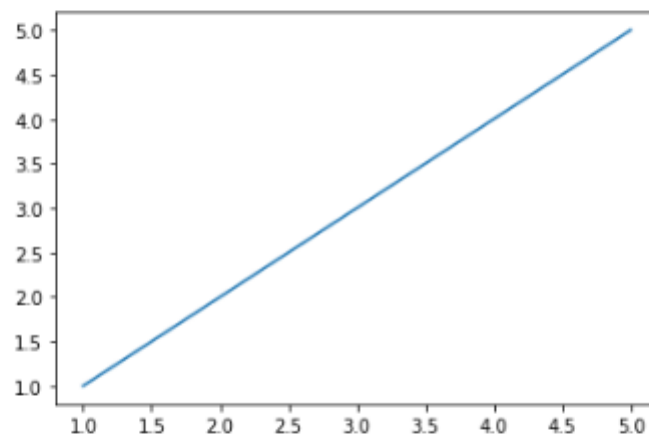
Рисунок 3 – Пример

```

Ввод [8]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
# 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

```



Out[8]: [<matplotlib.lines.Line2D at 0x256e203cd60>, <matplotlib.lines.Line2D at 0x256e203cd90>]

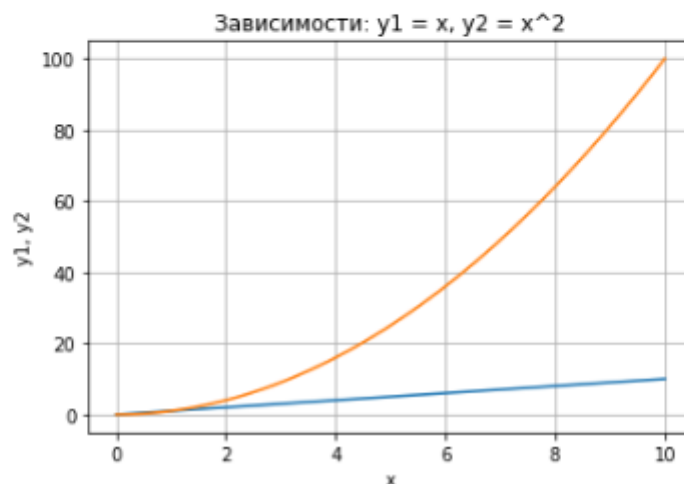


Рисунок 4 – Пример

```

Ввод [9]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
# Линейная зависимость
x = np.linspace(0, 10, 50)
y1 = x
# Квадратичная зависимость
y2 = [i**2 for i in x]
# Построение графиков
plt.figure(figsize=(9, 9))
plt.subplot(2, 1, 1)
plt.plot(x, y1) # построение графика
plt.title("Зависимости: y1 = x, y2 = 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) # ось ординат
plt.grid(True)

```

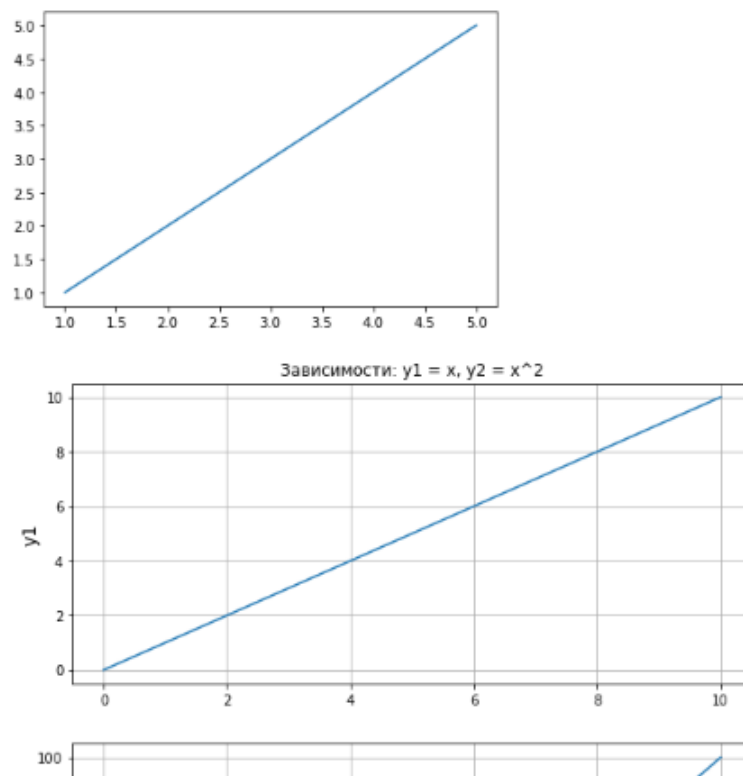


Рисунок 5 – Пример

```

Ввод [11]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
fruits = ["apple", "peach", "orange", "bannana", "melon"]
counts = [34, 25, 43, 31, 17]
plt.bar(fruits, counts)
plt.title("Fruits!")
plt.xlabel("Fruit")
plt.ylabel("Count")

```

Out[11]: Text(0, 0.5, 'Count')

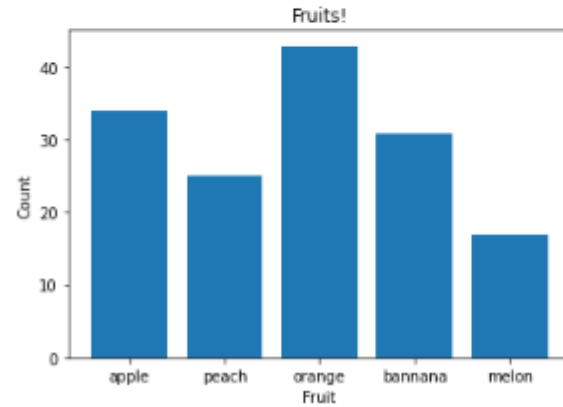


Рисунок 6 – Пример

```

Ввод [12]: import matplotlib.pyplot as plt
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
AutoMinorLocator)

import numpy as np
x = np.linspace(0, 10, 10)
y1 = 4*x
y2 = [i**2 for i in x]
fig, ax = plt.subplots(figsize=(8, 6))
ax.set_title("Графики зависимостей: y1=4*x, y2=x^2", fontsize=16)
ax.set_xlabel("x", fontsize=14)
ax.set_ylabel("y1, y2", fontsize=14)
ax.grid(which="major", linewidth=1.2)
ax.grid(which="minor", linestyle="--", color="gray", linewidth=0.5)
ax.scatter(x, y1, c="red", label="y1 = 4*x")
ax.plot(x, y2, label="y2 = x^2")
ax.legend()
ax.xaxis.set_minor_locator(AutoMinorLocator())
ax.yaxis.set_minor_locator(AutoMinorLocator())
ax.tick_params(which='major', length=10, width=2)
ax.tick_params(which='minor', length=5, width=1)
plt.show()

```

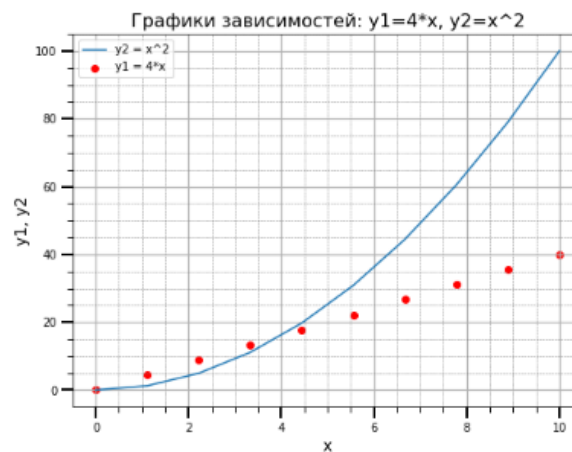


Рисунок 7 – Пример

```
Ввод [14]: #Построение графиков
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot()
plt.plot([1, 7, 3, 5, 11, 1])
```

Out[14]: [<matplotlib.lines.Line2D at 0x256e213e580>]

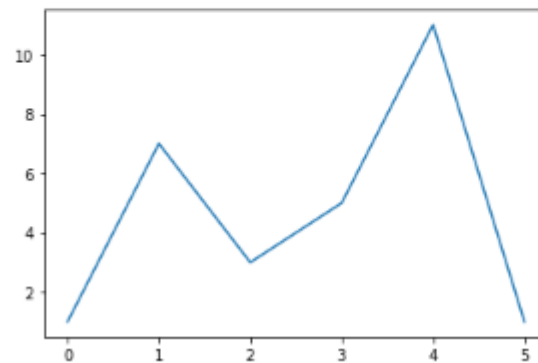


Рисунок 8 – Пример

```
Ввод [15]: #Построение графиков
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot()
x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]
plt.plot(x, y, label='steel price')
plt.title('Chart price', fontsize=15)
plt.xlabel('Day', fontsize=12, color='blue')
plt.ylabel('Price', fontsize=12, color='blue')
plt.legend()
plt.grid(True)
plt.text(15, 4, 'grow up!')
```

Out[15]: Text(15, 4, 'grow up!')

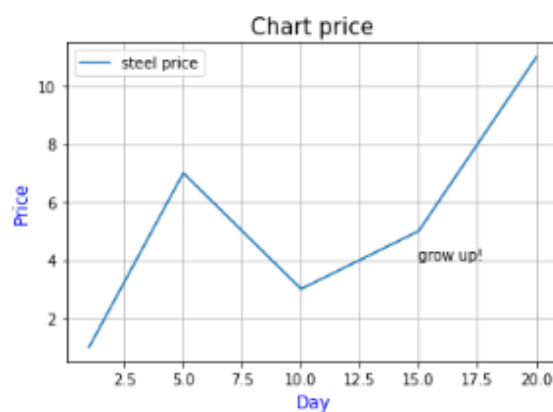


Рисунок 9 – Пример

```
Ввод [16]: #Построение графиков
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot()
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]
plt.plot(x, y1, '-', x, y2, '--', x, y3, '-.', x, y4, ':')
```

```
Out[16]: [<matplotlib.lines.Line2D at 0x256e2269e20>,
<matplotlib.lines.Line2D at 0x256e2269f40>,
<matplotlib.lines.Line2D at 0x256e2269f10>,
<matplotlib.lines.Line2D at 0x256e2275190>]
```

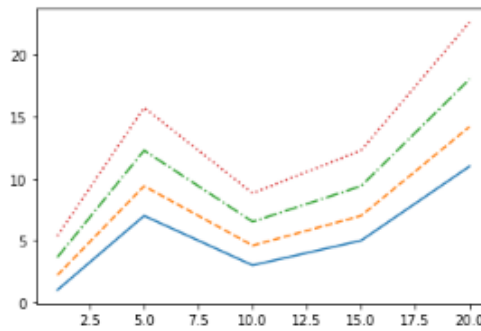


Рисунок 10 – Пример

```
Ввод [18]: #Работа с функцией subplot()
import matplotlib.pyplot as plt
%matplotlib inline
# 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]
# Настройка размеров подложки
plt.figure(figsize=(12, 7))
# 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, ':')
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x256e34fa910>]
```

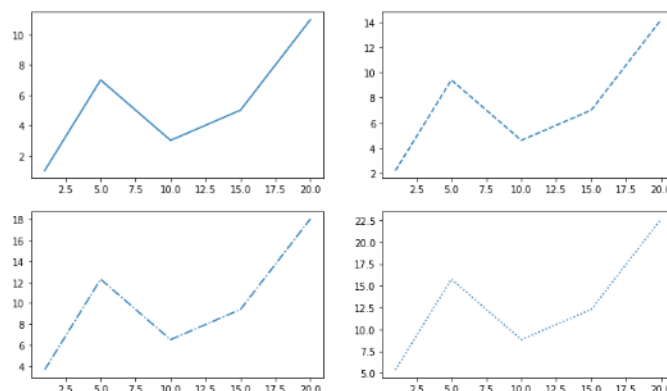


Рисунок 11 – Пример

Вывод: исследовал базовые возможности библиотеки matplotlib языка программирования Python.



## Ответы на вопросы:

### 1. Как осуществляется установка пакета matplotlib?

Сделать это очень просто, достаточно в терминале выполнить команду: `pip install matplotlib` и пакет со всеми зависимостями будет установлен. Также можно зайти в репозиторий: <https://pypi.org> набрать в поиске matplotlib и появится список доступных пакетов. По умолчанию, устанавливается последняя версия.

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

Магическая команда `%matplotlib` настраивает Jupyter Notebook для отображения графиков с помощью Matplotlib. По умолчанию используется стандартный графический бэкенд от Matplotlib, и ваши графики отображаются в отдельном окне. На заметку: Вы можете изменить бэкенд Matplotlib, передав аргумент в магическую команду `%matplotlib`.

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

Команда `plot(y)` строит график элементов одномерного массива `y` в зависимости от номера элемента; если элементы массива `y` комплексные, то строится график `plot(real(y), imag(y))`. Если `Y` - двумерный действительный массив, то строятся графики для столбцов; в случае комплексных элементов их мнимые части игнорируются.

Команда `plot(x, y)` соответствует построению обычной функции, когда одномерный массив `x` соответствует значениям аргумента, а одномерный массив `y` - значениям функции. Когда один из массивов `X` или `Y` либо оба двумерные, реализуются следующие построения:

если массив `Y` двумерный, а массив `x` одномерный, то строятся графики для столбцов массива `Y` в зависимости от элементов вектора `x`;

если двумерным является массив `X`, а массив `y` одномерный, то строятся графики столбцов массива `X` в зависимости от элементов вектора `y`;

если оба массива  $X$  и  $Y$  двумерные, то строятся зависимости столбцов массива  $Y$  от столбцов массива  $X$ .

Команда `plot(x, y, s)` позволяет выделить график функции, указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной  $s$ .

#### 4. Как отобразить несколько графиков на одном поле?

Для того чтобы отобразить несколько независимых графиков в одном окне - предназначена функция `subplot()` из пакета `pylab`.

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

`Seaborn` — библиотека для создания статистических графиков на `Python`. Она построена на основе `matplotlib` и тесно интегрируется со структурами данных `pandas`. `Seaborn` помогает вам изучить и понять данные. Его функции построения графиков работают с датасетами и выполняют все необходимые преобразования для создания информативных графиков.

#### 6. Какие основные элементы графика Вам известны?

Основные элементы графика следующие: поле графика, геометрические знаки, пространственные ориентиры, масштаб, экспликация графика.

Поле графика - пространство, в котором размещаются геометрические знаки, образующие график. Он характеризуется форматом и соотношением сторон.

#### 7. Как осуществляется управление текстовыми надписями на графике?

В части текстового наполнения при построении графика выделяют следующие составляющие:

заголовок поля (`title`);

заголовок фигуры (`suptitle`);

подписи осей (`xlabel`, `ylabel`);

тестовый блок на поле графика (`text`), либо на фигуре (`figtext`);

аннотация (`annotate`) – текст и указатель.

#### 8. Как осуществляется управление легендой графика?

Для отображения легенды на графике используется функция `legend()`

#### 9. Как задать цвет и стиль линий графика?

Чтобы задать цвет закрашивания графического объекта в python используется команда `obj.setFill ("цвет")`. Пример программы на Python, которая рисует закрашенную синюю окружность `from graphics import* win = GraphWin ("Окно для графики", 400, 400) obj = Circle (Point (200, 200), 50) obj.setFill ("blue") obj.draw (win) win.getMouse () win.close ()`.

#### 10. Как выполнить размещение графика в разных полях?

Самый простой способ представить графики в отдельных полях – это использовать функцию `subplot()` для задания их мест размещения.