

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №4 по дисциплине:
Технологии распознавания образов**

Выполнил:

студент группы ПИЖ-б-о-20-1

Примаков Вадим Дмитриевич

Проверил:

доцент кафедры

инфокоммуникаций Романкин Р.А.

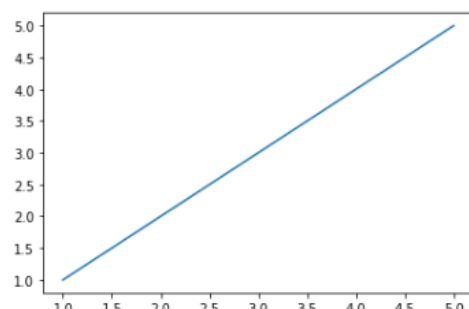
Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ:

1. Использование библиотеки matplotlib

```
In [1]: import matplotlib
%matplotlib inline
```

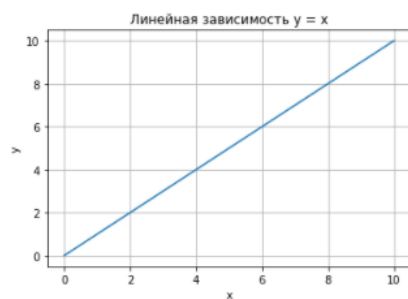
```
In [2]: import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
plt.show()
```



2. Построение графиков

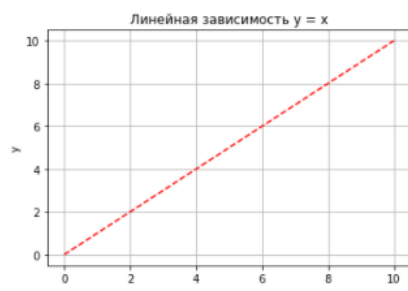
```
In [3]: import numpy as np
x = np.linspace(0, 10, 50)
y = x
plt.title("Линейная зависимость y = x")
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
plt.plot(x, y)
```

Out[3]: [



```
In [4]: plt.title("Линейная зависимость y = x")
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
plt.plot(x, y, "r--")
```

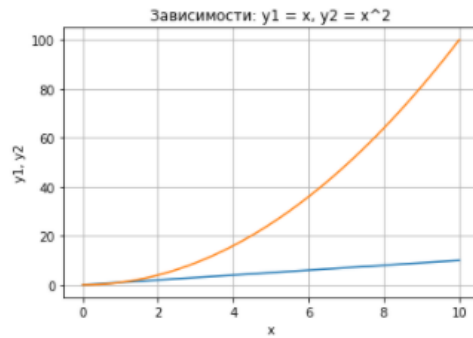
Out[4]: [



3. Несколько графиков на одном поле

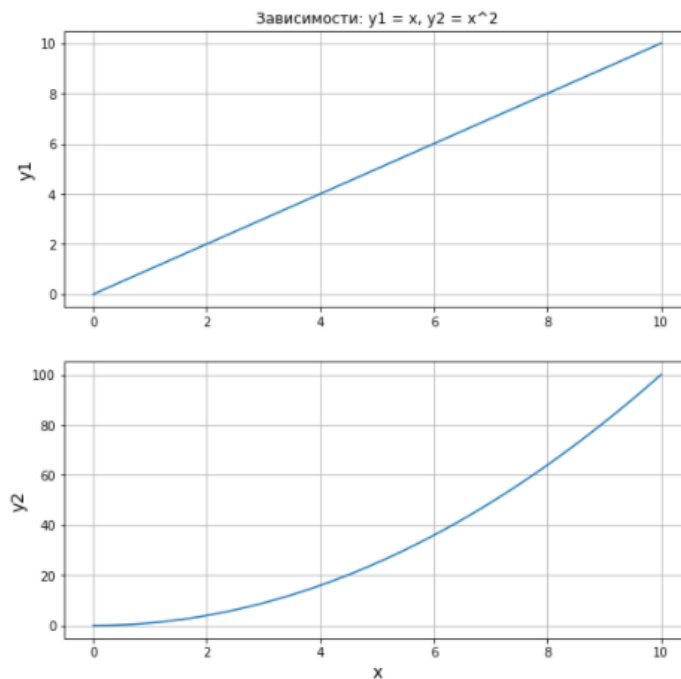
```
In [6]: x = np.linspace(0, 10, 50)
y1 = x
y2 = [i**2 for i in x]
plt.title("Зависимости: y1 = x, y2 = x^2")
plt.xlabel("x")
plt.ylabel("y1, y2")
plt.grid()
plt.plot(x, y1, x, y2)
```

```
Out[6]: [<matplotlib.lines.Line2D at 0x1d20f599040>,
<matplotlib.lines.Line2D at 0x1d20f599100>]
```



4. Несколько разделенных полей с графиками

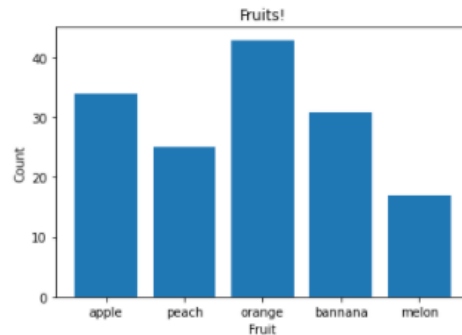
```
In [8]: x = np.linspace(0, 10, 50)
y1 = x
y2 = [i**2 for i in x]
plt.figure(figsize=(9, 9))
plt.subplot(2, 1, 1)
plt.plot(x, y1)
plt.title("Зависимости: y1 = x, y2 = x^2")
plt.ylabel("y1", fontsize=14)
plt.grid(True)
plt.subplot(2, 1, 2)
plt.plot(x, y2)
plt.xlabel("x", fontsize=14)
plt.ylabel("y2", fontsize=14)
plt.grid(True)
```



5. Построение диаграммы для категориальных данных

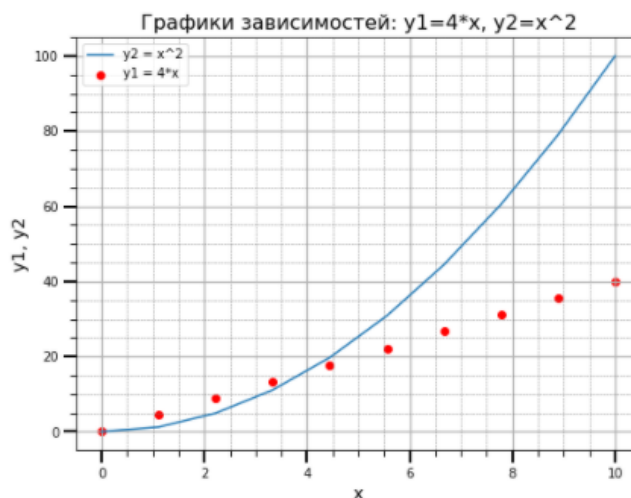
```
In [9]: fruits = ["apple", "peach", "orange", "bannana", "melon"]
counts = [34, 25, 43, 31, 17]
plt.bar(fruits, counts)
plt.title("Fruits!")
plt.xlabel("Fruit")
plt.ylabel("Count")
```

```
Out[9]: Text(0, 0.5, 'Count')
```



6. Основные элементы графика

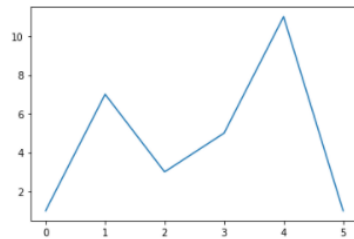
```
In [10]: import matplotlib.pyplot as plt
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinorLocator)
import numpy as np
x = np.linspace(0, 10, 10)
y1 = 4*x
y2 = [i**2 for i in x]
fig, ax = plt.subplots(figsize=(8, 6))
ax.set_title("Графики зависимостей: y1=4*x, y2=x^2", 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()
```



7. Работа с инструментом pyplot

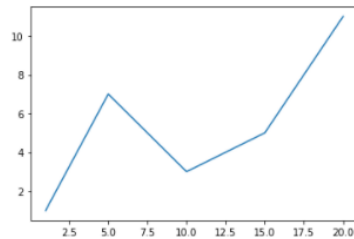
```
In [11]: import matplotlib.pyplot as plt
%matplotlib inline
plt.plot()
plt.plot([1, 7, 3, 5, 11, 1])
```

Out[11]: [<matplotlib.lines.Line2D at 0x1d20f9ff700>]



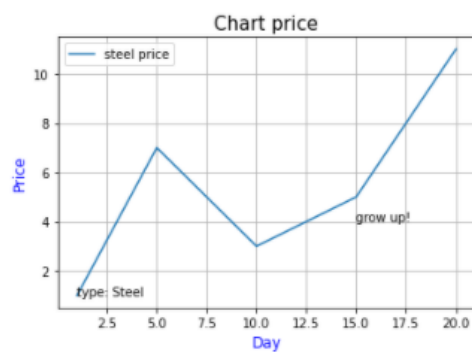
```
In [12]: plt.plot([1, 5, 10, 15, 20], [1, 7, 3, 5, 11])
```

Out[12]: [<matplotlib.lines.Line2D at 0x1d20fa55910>]



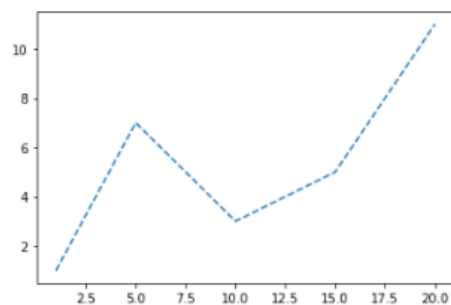
```
In [13]: plt.xlabel('Day', fontsize=15, color='blue')
plt.title('Chart price', fontsize=17)
plt.text(1, 1, 'type: Steel')
x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]
plt.plot(x, y, label='steel price')
plt.title('Chart price', fontsize=15)
plt.xlabel('Day', fontsize=12, color='blue')
plt.ylabel('Price', fontsize=12, color='blue')
plt.legend()
plt.grid(True)
plt.text(15, 4, 'grow up!')
```

Out[13]: Text(15, 4, 'grow up!')



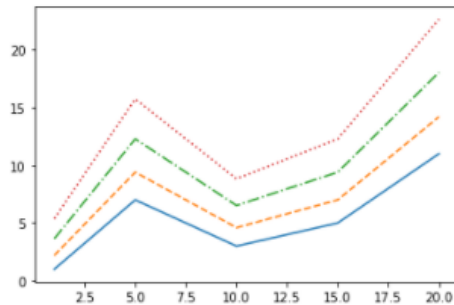
```
In [14]: x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]
plt.plot(x, y, '--')
```

Out[14]: [<matplotlib.lines.Line2D at 0x1d20f659b80>]



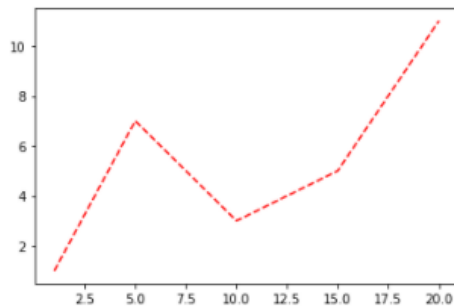
```
In [16]: x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [i*1.2 + 1 for i in y1]
y3 = [i*1.2 + 1 for i in y2]
y4 = [i*1.2 + 1 for i in y3]
plt.plot(x, y1, '-', x, y2, '--', x, y3, '-.', x, y4, ':')
```

```
Out[16]: [<matplotlib.lines.Line2D at 0x1d20d8fce50>,
<matplotlib.lines.Line2D at 0x1d20d924df0>,
<matplotlib.lines.Line2D at 0x1d20f3cdac0>,
<matplotlib.lines.Line2D at 0x1d20d924ee0>]
```



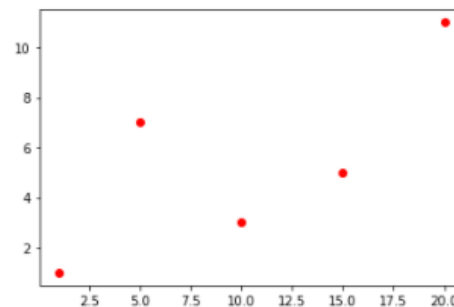
```
In [17]: x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]
plt.plot(x, y, '--r')
```

```
Out[17]: [<matplotlib.lines.Line2D at 0x1d20fb061f0>]
```



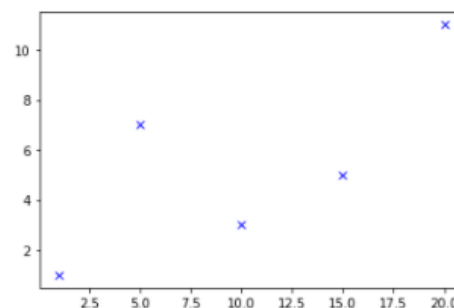
```
In [18]: plt.plot(x, y, 'ro')
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x1d20fb57c70>]
```



```
In [19]: plt.plot(x, y, 'bx')
```

```
Out[19]: [<matplotlib.lines.Line2D at 0x1d20fbb38e0>]
```



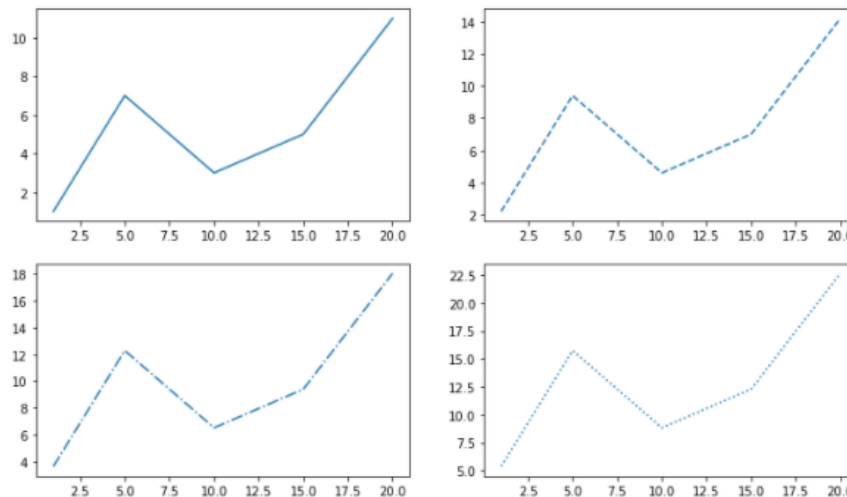
```
In [20]: x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [i*1.2 + 1 for i in y1]
y3 = [i*1.2 + 1 for i in y2]
y4 = [i*1.2 + 1 for i in y3]

plt.figure(figsize=(12, 7))

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[20]: [<matplotlib.lines.Line2D at 0x1d20fcbf1c0>]



Вопросы для защиты работы:

1. Как осуществляется установка пакета matplotlib?

Существует два основных варианта установки этой библиотеки: установить пакет Anaconda, в состав которого входит большое количество различных инструментов, или – установить Matplotlib самостоятельно, используя менеджер пакетов.

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

```

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)

```

plt.show()

4. Как отобразить несколько графиков на одном поле?

```
# Линейная зависимость
x = np.linspace(0, 10, 50)
y1 = x

# Квадратичная зависимость
y2 = [i**2 for i in x]

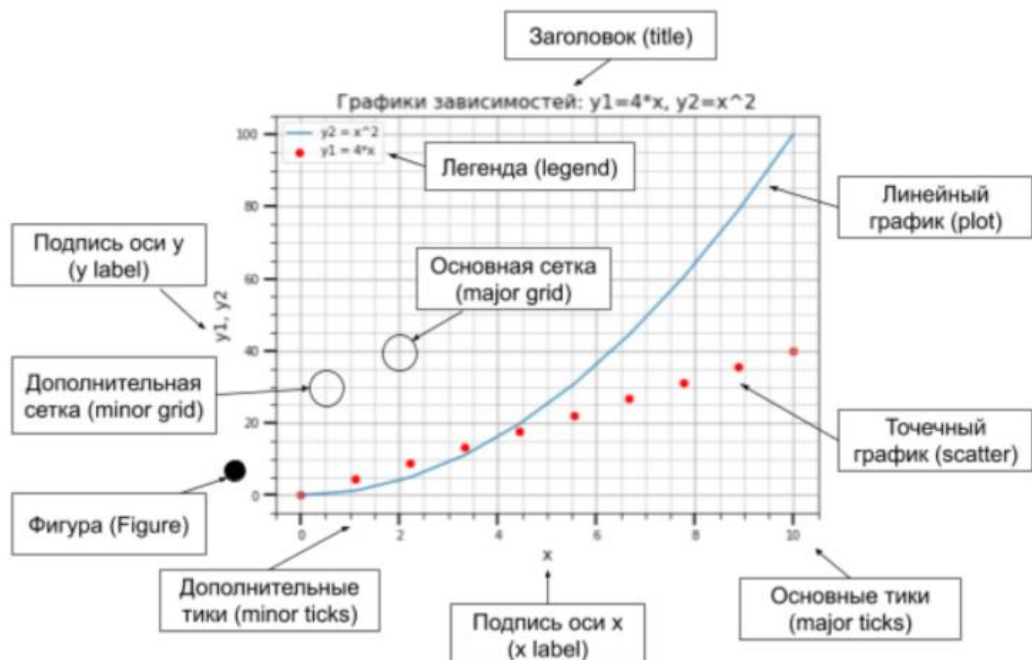
# Построение графика
plt.title("Зависимости: y1 = x, y2 = x^2") # заголовок
plt.xlabel("x") # ось абсцисс
plt.ylabel("y1, y2") # ось ординат
plt.grid() # включение отображение сетки

plt.plot(x, y1, x, y2) # построение графика
```

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

Для вывода диаграммы была использована функция bar().

6. Какие основные элементы графика вам известны?



Корневым элементом при построении графиков в системе Matplotlib является Фигура.

График. Matplotlib предоставляет огромное количество различных настроек, которые можно использовать для того, чтобы придать графику нужный вид.

Оси. Для каждой оси можно задать метку (подпись), основные (major) и дополнительные (minor) тики, их подписи, размер и толщину, также можно задать диапазоны по каждой из осей.

Сетка и легенда. Сетка также может быть основной (major) и дополнительной (minor). Каждому типу сетки можно задавать цвет, толщину линии и тип.

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.

Для задания подписи оси x используется функция `xlabel()`, оси y – `ylabel()`.

Для задания заголовка графика используется функция `title()`.

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

8. Как осуществляется управление легендой графика?

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

9. Как задать цвет и стиль линий 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

Задание цвета линии графика производится через параметр *color*.

10. Как выполнить размещение графика в 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.