

**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 «Основы программной инженерии»**

Выполнил студент группы

ПИЖ-б-о-21-1

Зиберов Александр

« » марта 2023 г.

Подпись студента _____

Работа защищена

« » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь, 2023

Цель работы:

Исследовать базовые возможности библиотеки matplotlib языка программирования Python.

Выполнение работы:

Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT, рисунок 1.

Ссылка: https://github.com/afk552/trolab_4

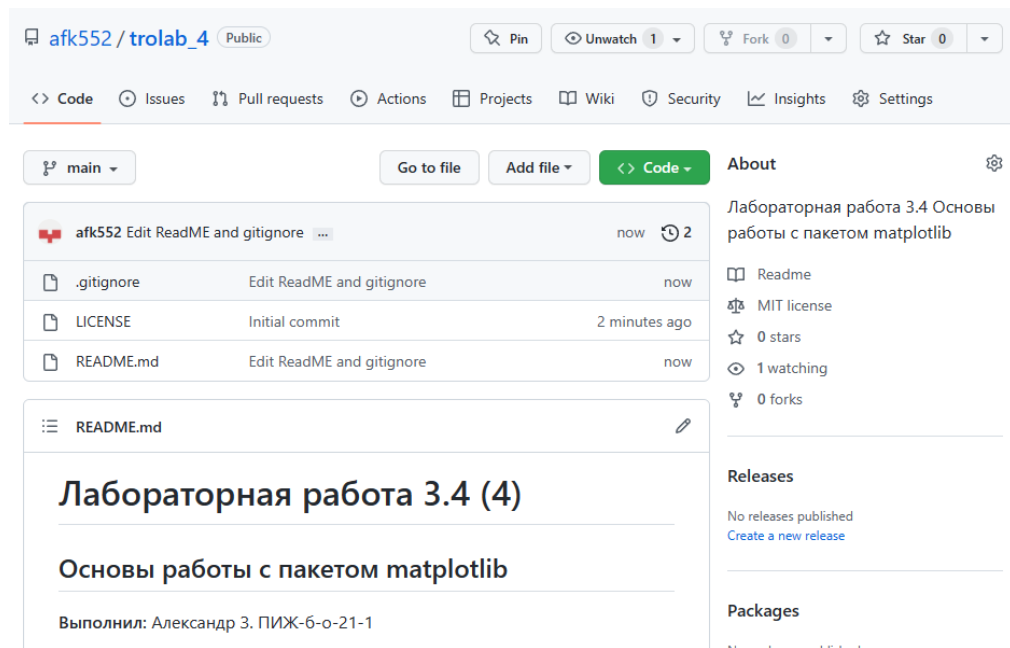


Рисунок 1 – Удаленный репозиторий на GitHub

Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm, рисунок 2.

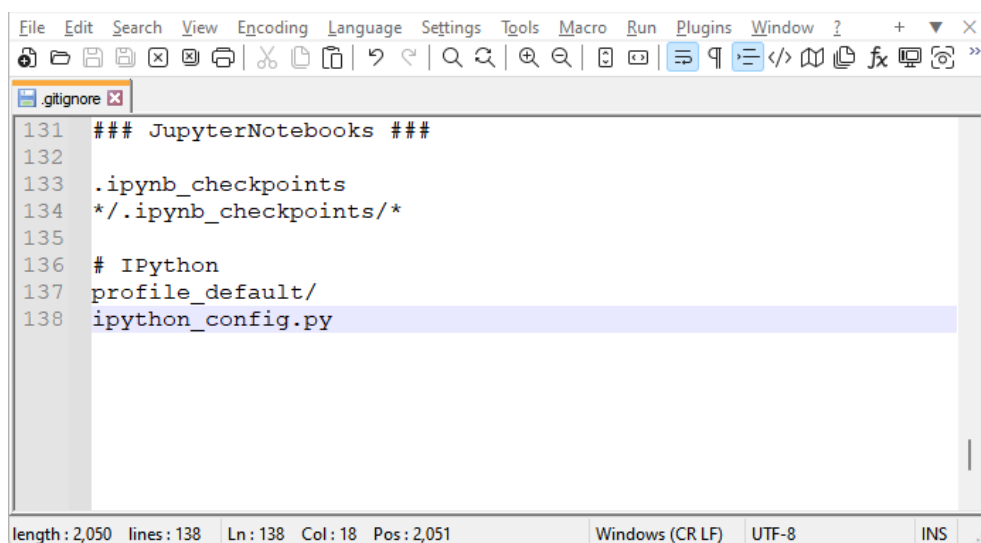


Рисунок 2 – Окно блокнота

Организируйте свой репозиторий в соответствии с моделью ветвления git-flow, рисунок 3.

```
* develop
main
```

Рисунок 3 – Окно командной строки

Проработать примеры лабораторной работы в отдельном ноутбуке

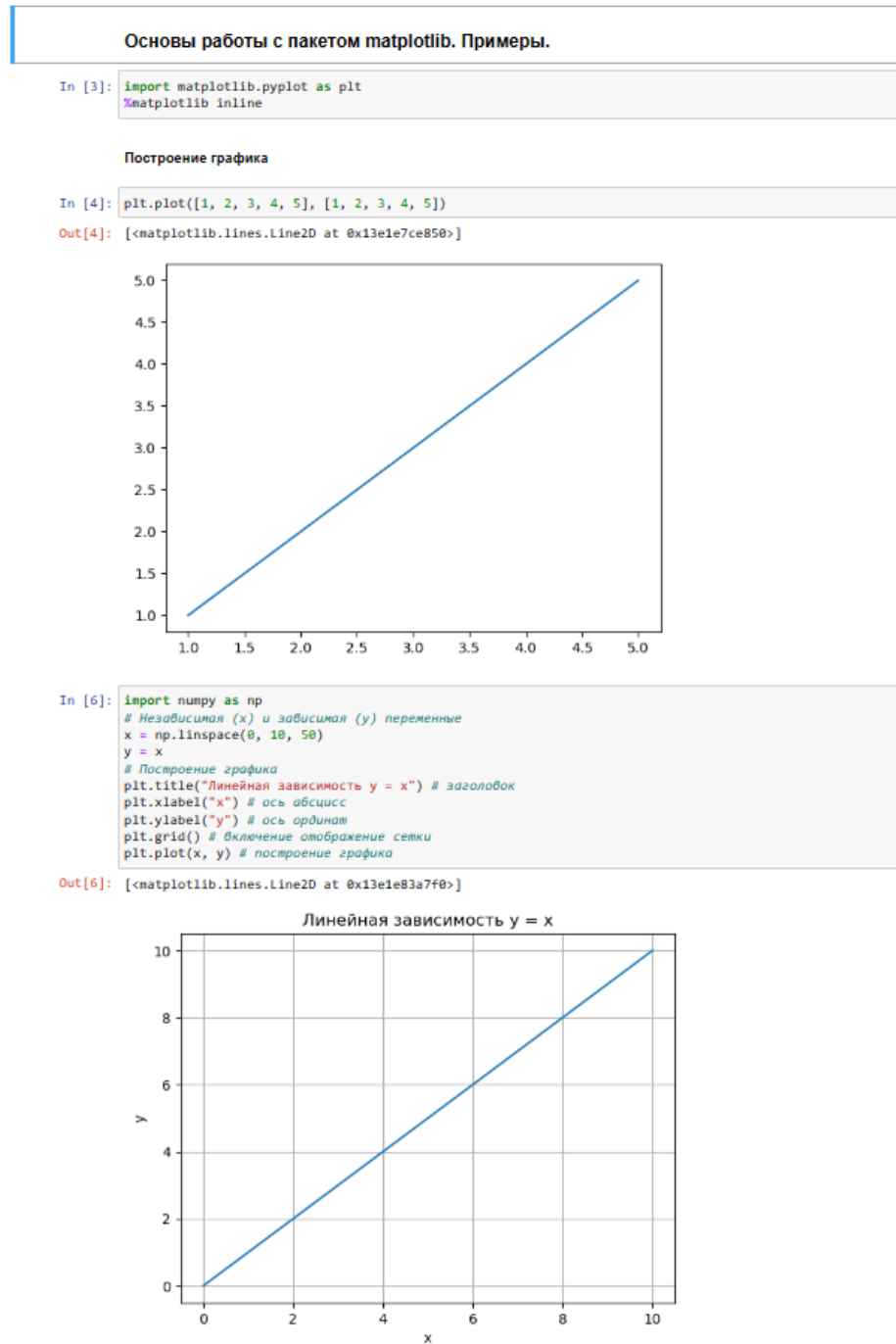
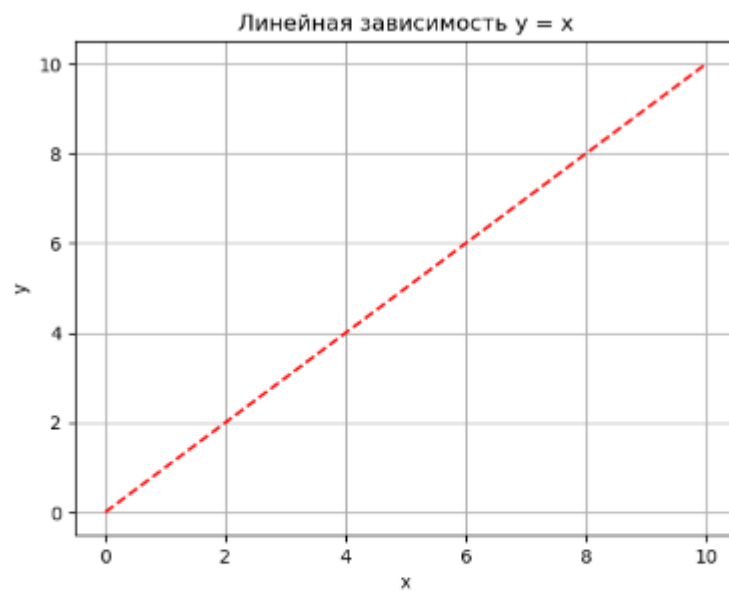


Рисунок 4 – Пример 1



Несколько графиков на одном поле

```
In [8]: # 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 0x13e1e90d8e0>,
<matplotlib.lines.Line2D at 0x13e1e90d940>]
```

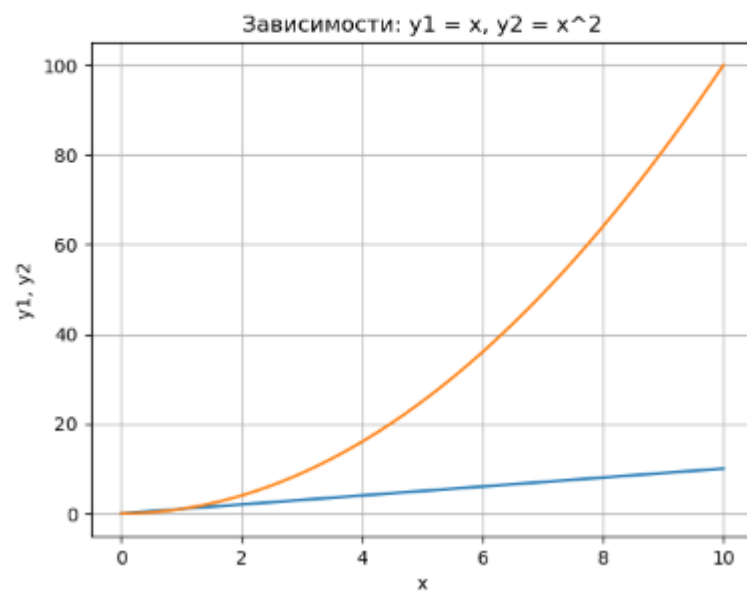


Рисунок 5 – Пример 2

Несколько разделенных полей с графиками

```
In [10]: # 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) # включение отображение сетки
```

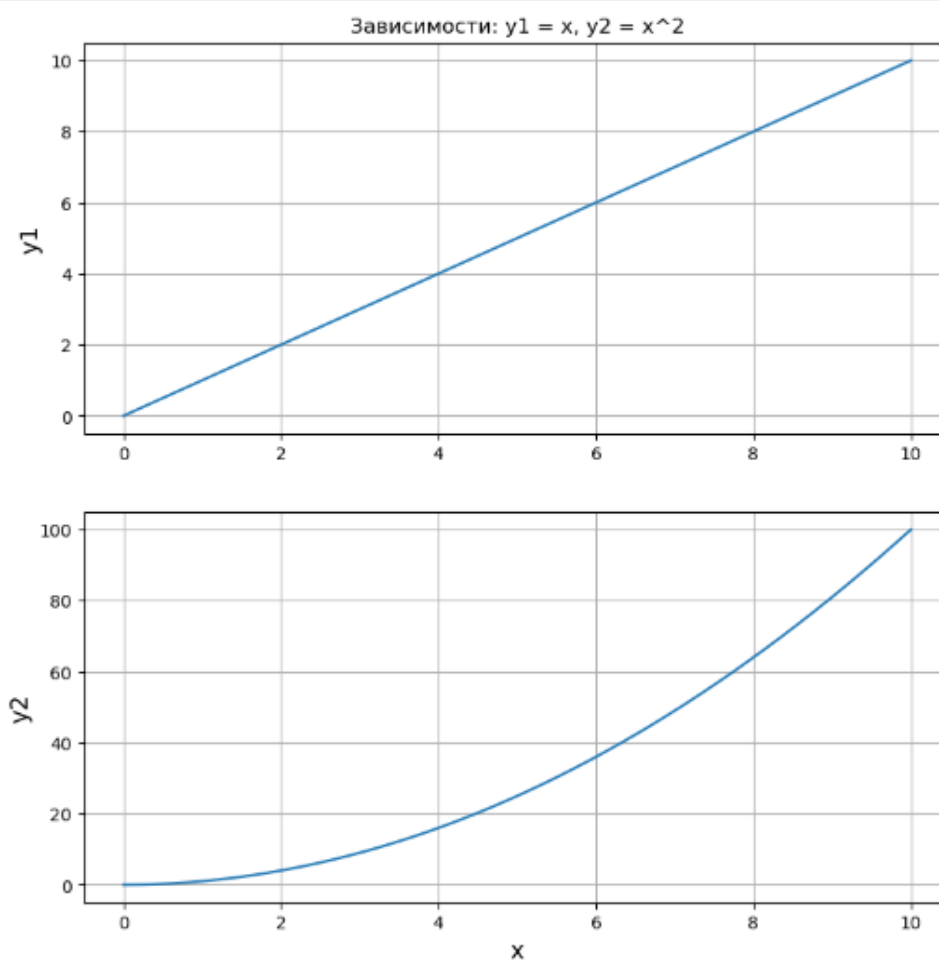


Рисунок 6 – Пример 3

Построение диаграммы для категориальных данных

```
In [11]: 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')
```

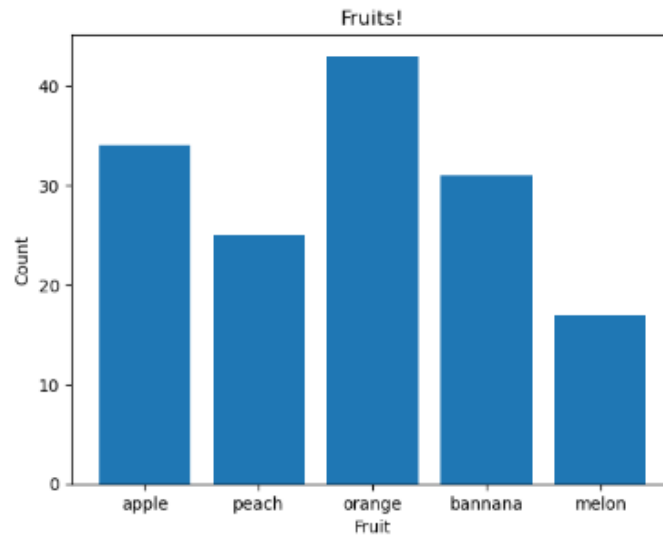


Рисунок 7 – Пример 4

Основные элементы графика

```
In [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()
```

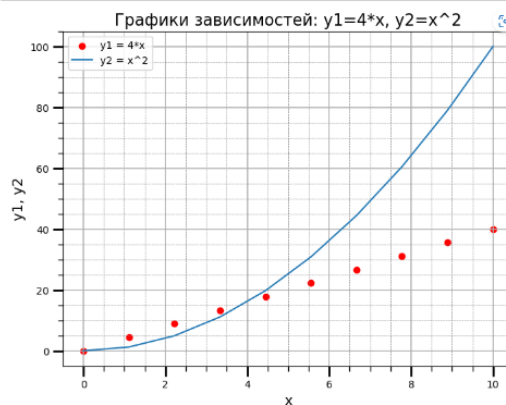


Рисунок 8 – Пример 5

Текстовые надписи на графиках

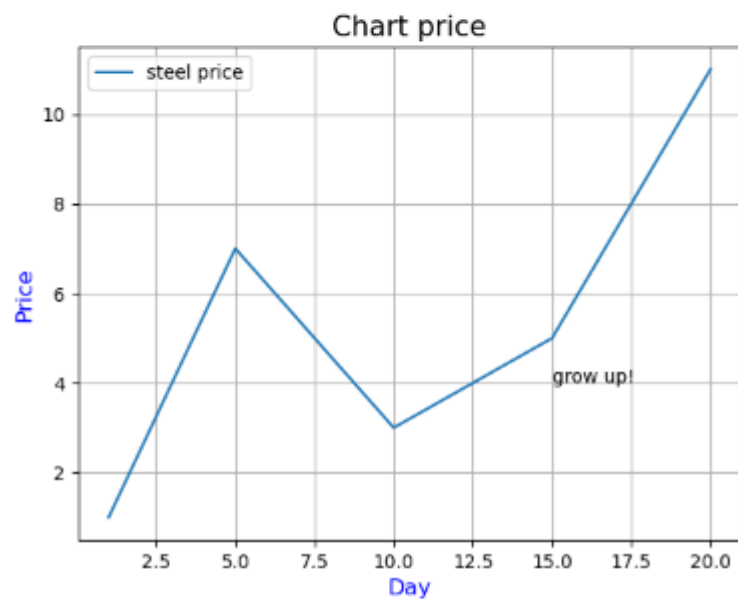
```
In [17]: x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]

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



Работа с линейным графиком

```
In [19]: x = [1, 5, 10, 15, 20]
y = [1, 7, 3, 5, 11]
plt.plot(x, y, '--')
```

Out[19]: [<matplotlib.lines.Line2D at 0x13e1eb17880>]

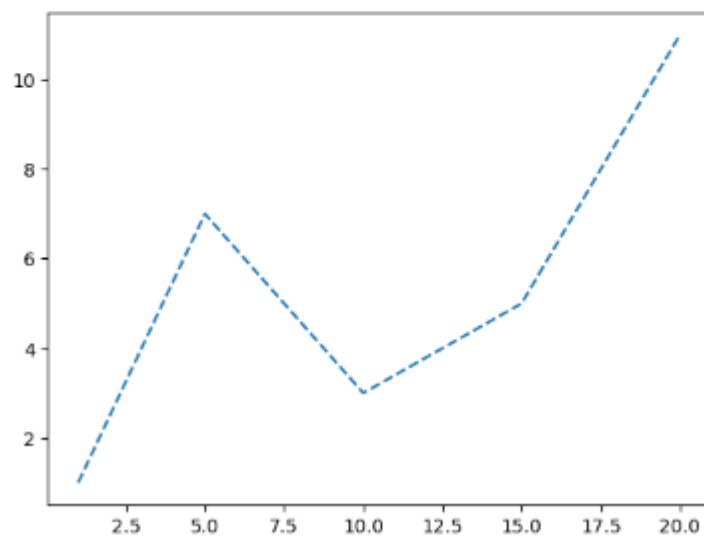
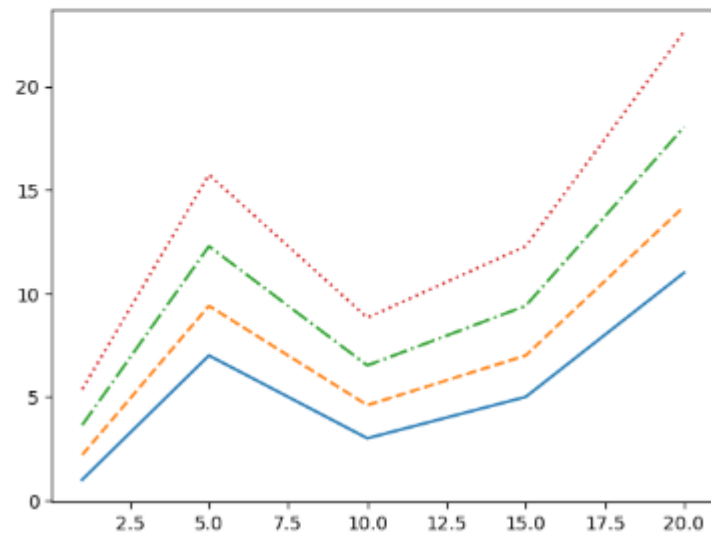


Рисунок 9 – Пример 6

```
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.plot(x, y1, '-', x, y2, '--', x, y3, '-.', x, y4, ':')
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x13e1ea916a8>,
<matplotlib.lines.Line2D at 0x13e1ea91618>,
<matplotlib.lines.Line2D at 0x13e1ea91828>,
<matplotlib.lines.Line2D at 0x13e1ede74c8>]
```



```
In [21]: plt.plot(x, y1, '-')
plt.plot(x, y2, '--')
plt.plot(x, y3, '-.')
plt.plot(x, y4, ':')
```

```
Out[21]: [<matplotlib.lines.Line2D at 0x13e1eb04d08>]
```

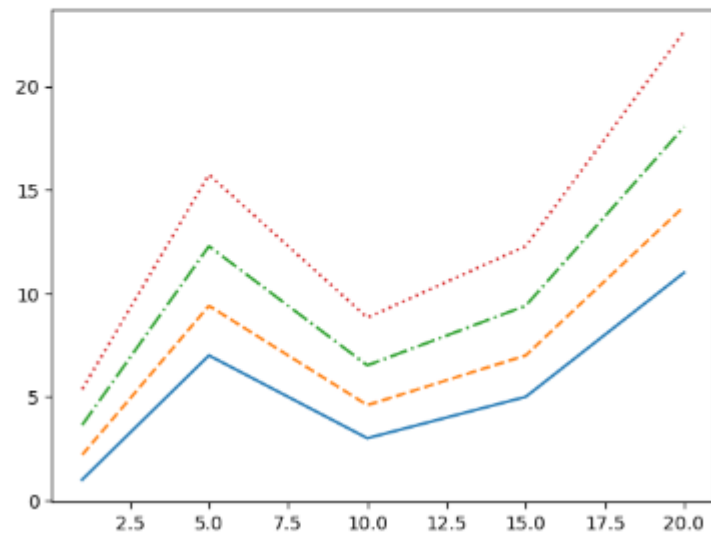
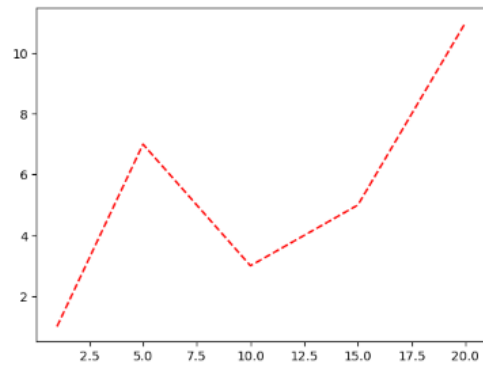


Рисунок 10 – Пример 7

Цвет линии

```
In [22]: x = [1, 5, 10, 15, 20]  
y = [1, 7, 3, 5, 11]  
plt.plot(x, y, '--r')
```

```
Out[22]: [<matplotlib.lines.Line2D at 0x13e1ebef48>]
```



Тип графика

```
In [25]: plt.plot(x, y, 'ro')
```

```
Out[25]: [<matplotlib.lines.Line2D at 0x13e1eb9438>]
```

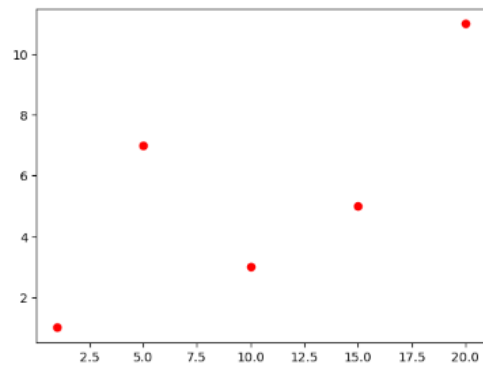


Рисунок 11 – Пример 8

```
In [26]: plt.plot(x, y, 'bx')
```

```
Out[26]: [<matplotlib.lines.Line2D at 0x13e1ef1b708>]
```

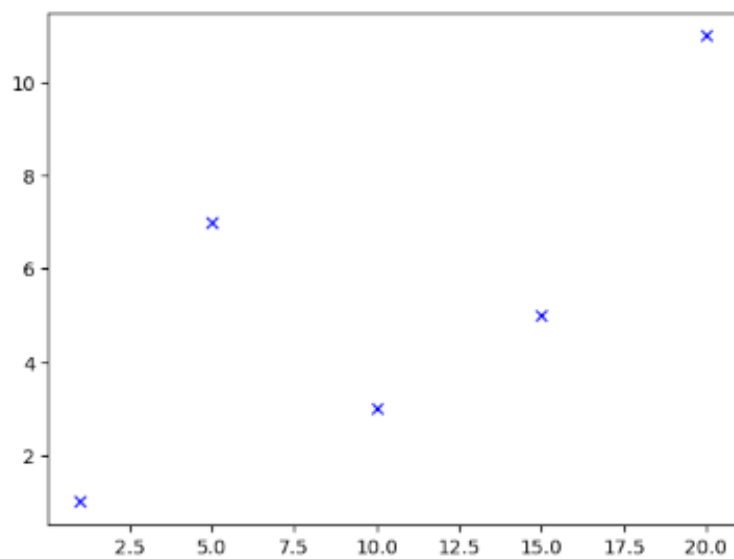


Рисунок 12 – Пример 9

Размещение графиков на разных полях

Работа с функцией subplot()

```
In [30]: # 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[30]: [matplotlib.lines.Line2D at 0x13e1f2bb880]
```

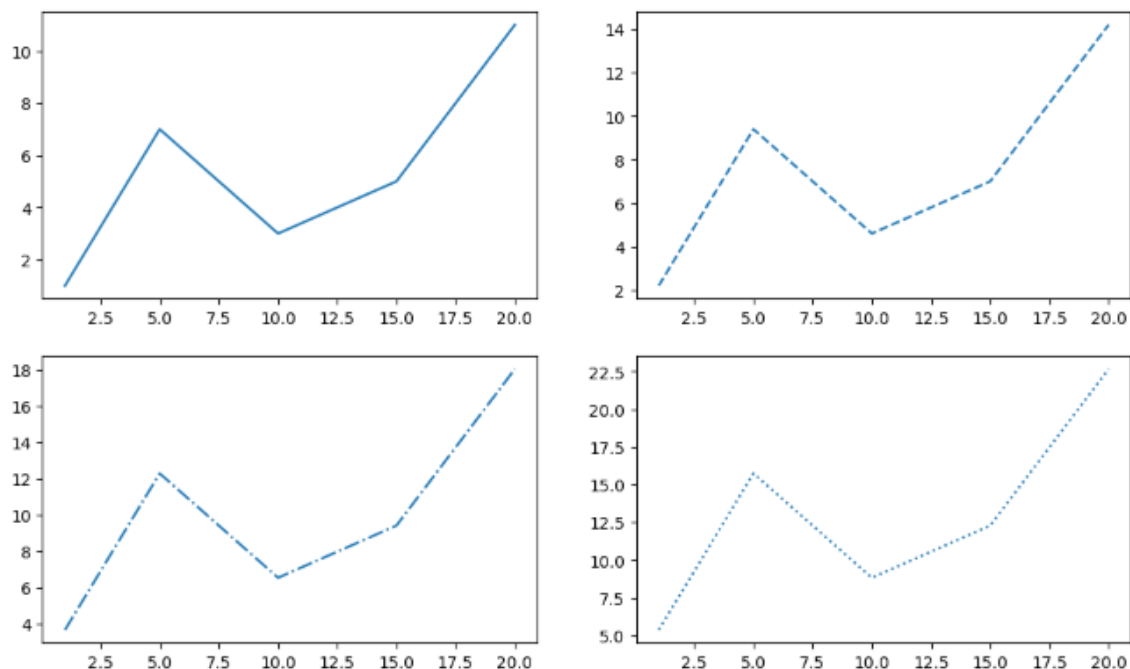


Рисунок 13 – Пример 10

Второй вариант использования subplot()

```
In [32]: # Вывод графиков
plt.subplot(221)
plt.plot(x, y1, '-')

plt.subplot(222)
plt.plot(x, y2, '--')

plt.subplot(223)
plt.plot(x, y3, '-.')

plt.subplot(224)
plt.plot(x, y4, ':')
```

Out[32]: [matplotlib.lines.Line2D at 0x13e1f43acd0]

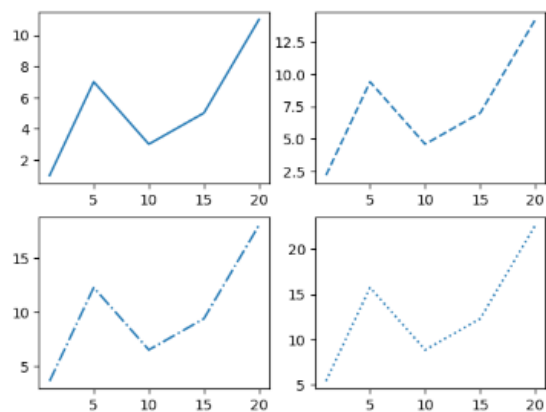


Рисунок 14 – Пример 11

Работа с функцией subplots()

```
In [34]: fig, axes = plt.subplots(2, 2, figsize=(12, 7))
axes[0, 0].plot(x, y1, '-')
axes[0, 1].plot(x, y2, '--')
axes[1, 0].plot(x, y3, '-.')
axes[1, 1].plot(x, y4, ':')
```

Out[34]: [matplotlib.lines.Line2D at 0x13e2199e3d0]

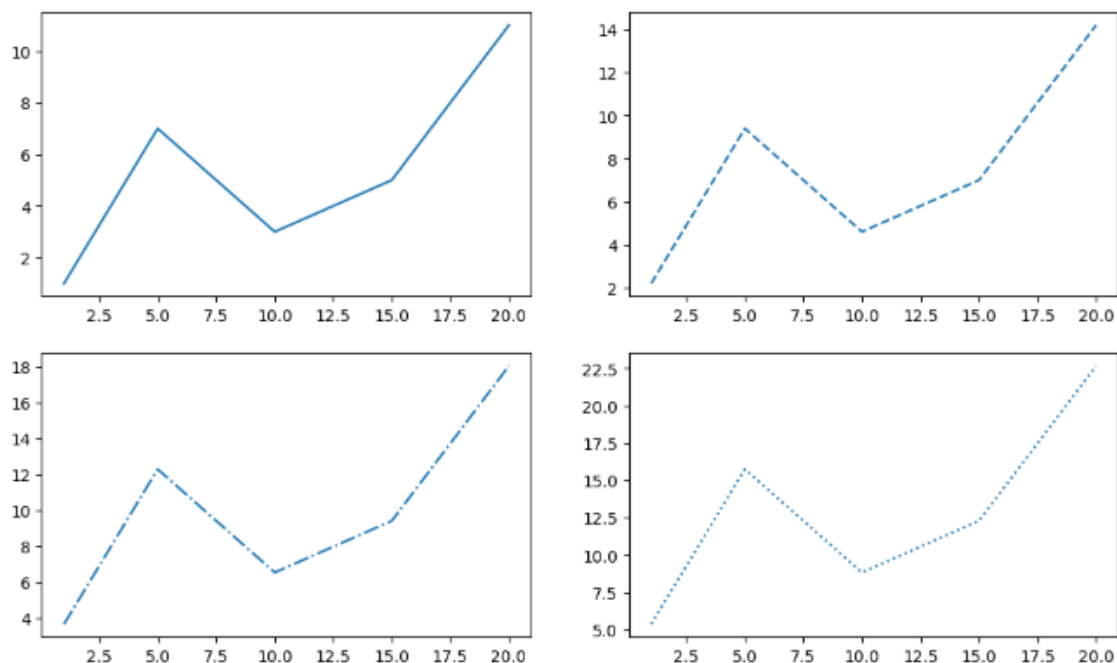


Рисунок 15 – Пример 12

Вывод: В результате выполнения работы были исследованы базовые возможности пакета matplotlib языка Python для построения графиков.

Контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Пакет доступен в Anaconda или через pip:

```
$ 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([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
```

```
plt.show()
```

4. Как отобразить несколько графиков на одном поле?

Для того, чтобы вывести несколько графиков на одном поле необходимо передать соответствующие наборы значений в функцию plot(). Построим несколько наборов данных и выведем их с использованием различных стилей линии:

```
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, ':')
```

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

Функция `bar()`.

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

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

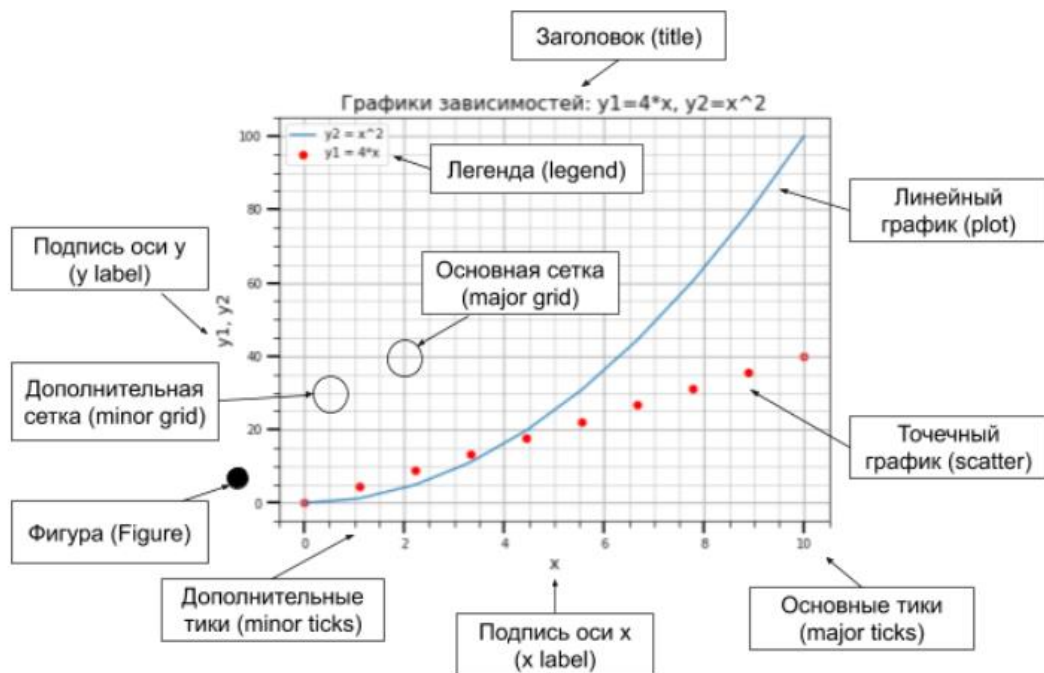
```
plt.bar(fruits, counts)
```

```
plt.title("Fruits!")
```

```
plt.xlabel("Fruit")
```

```
plt.ylabel("Count")
```

6. Какие основные элементы графика Вам известны?



7. Как осуществляется управление текстовыми надписями на графике?

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

Для задания заголовка графика используется функция `title()`. Для функции `title()` также доступны параметры конструктора класса

`matplotlib.text.Text`, часть из них представлена в описании аргументов функций `xlabel()` / `ylabel()`.

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()`.

Местоположение легенды - `loc`.

Размер шрифта - `ax.legend(fontsize=font)`

Название легенды - `plt.legend(title='My Awesome Legend')`

9. Как задать цвет и стиль линий графика?

Задание цвета линии графика производится через параметр `color` (или `c`, если использовать сокращенный вариант).

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

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.

```
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, ':')
```