

Тема 2.1.1. Визуализация данных.

Инструменты визуализации данных. Библиотека **matplotlib**

matplotlib - математическая библиотека построения диаграмм.

С помощью **matplotlib** можно строить простые диаграммы, графики, диаграммы разброса данных и т.д.

Подключение **matplotlib**:

```
import matplotlib.pyplot as plt
```

Построение простого графика

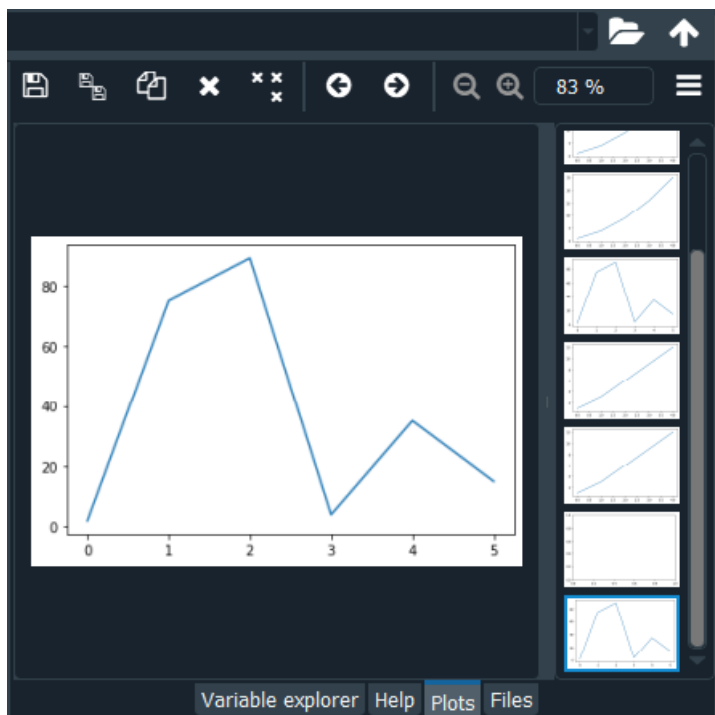
```
import matplotlib.pyplot as plt  
squares = [2, 75, 89, 4, 35, 15]  
fig, ax = plt.subplots()  
ax.plot(squares)  
plt.show()
```

Функция `subplots()` позволяет сгенерировать одну или несколько поддиаграмм на одной диаграмме.

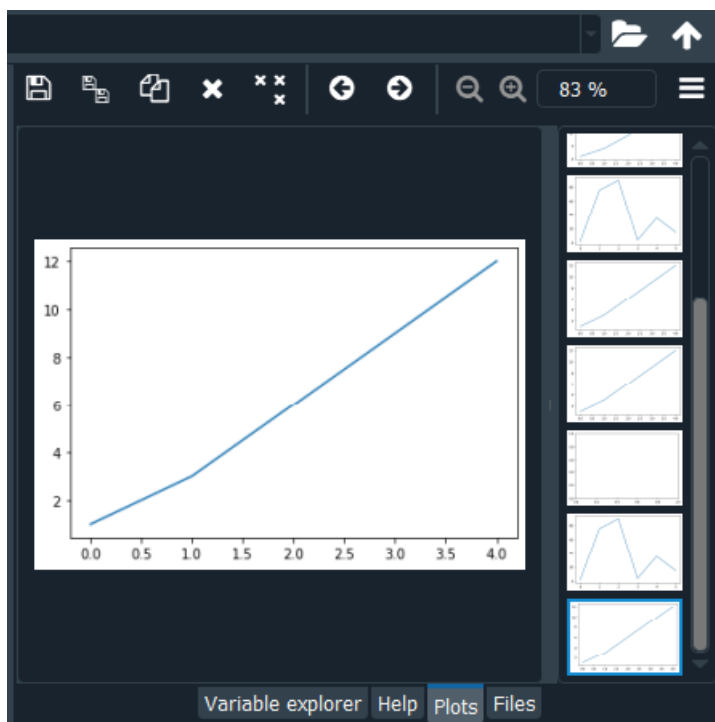
Переменная `fig` представляет весь набор генерируемых диаграмм.

Функция `plot()` строит графическое представление для заданных чисел.

Вызов `plt.show()` открывает окно просмотра и выводит график

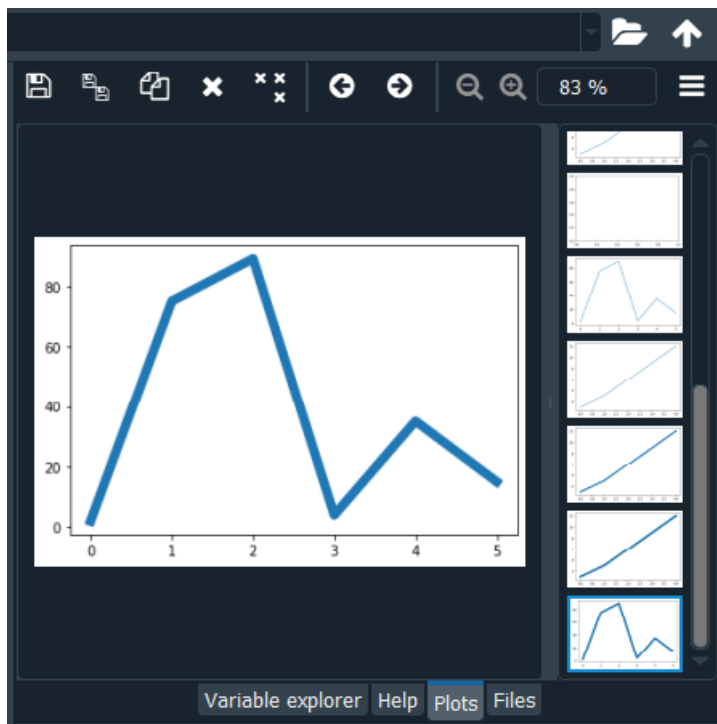


Изменим squares = [1, 3, 6, 9, 12]



Изменение типа надписей и толщины графика

```
import matplotlib.pyplot as plt  
  
squares = [2, 75, 89, 4, 35, 15]  
  
fig, ax = plt.subplots()  
  
ax.plot(squares, linewidth = 7)  
  
plt.show()
```

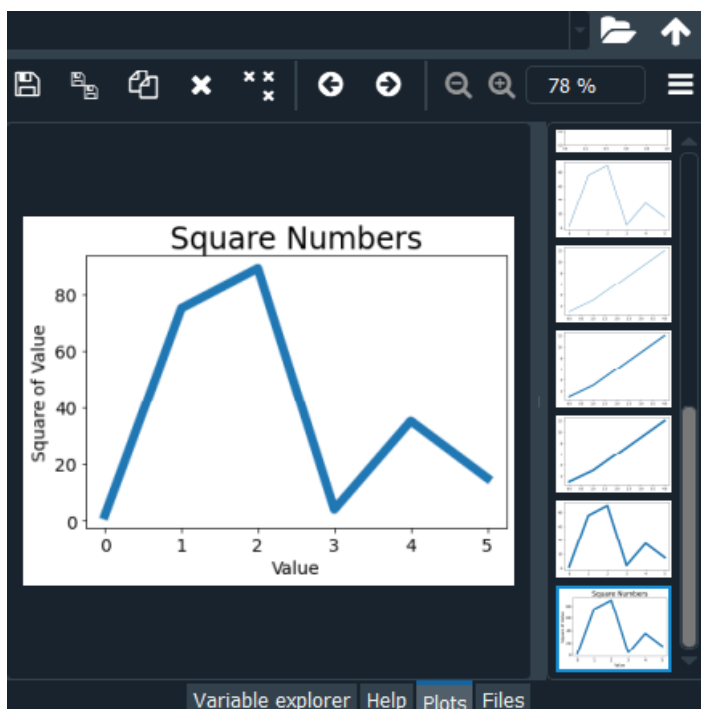


Назначение заголовка диаграммы, меток осей, размера шрифта делений на осях

```
import matplotlib.pyplot as plt  
  
squares = [2, 75, 89, 4, 35, 15]  
  
fig, ax = plt.subplots()  
  
ax.plot(squares, linewidth = 7)  
  
ax.set_title("Square Numbers", fontsize = 24)
```

```
ax.set_xlabel("Value", fontsize = 14)
ax.set_ylabel("Square of Value", fontsize = 14)
ax.tick_params(axis = 'both', labelsizе = 14)

plt.show()
```



Встроенные стили оформления диаграмм

Запустите код:

```
import matplotlib.pyplot as plt
print("Встроенные стили оформления диаграмм:")
print(plt.style.available)
```

Результат:

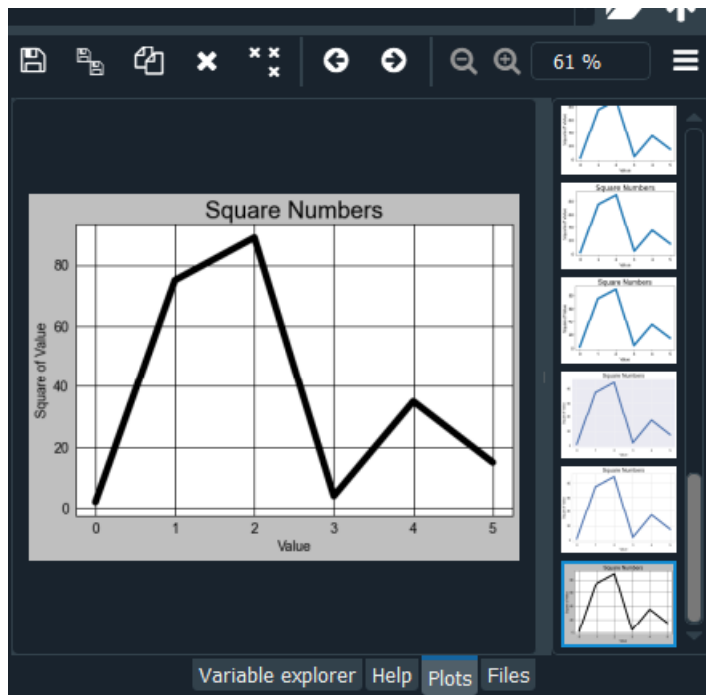
```
Встроенные стили оформления диаграмм:  
['Solarize_Light2', '_classic_test_patch',  
'bmh', 'classic', 'dark_background', 'fast',  
'fivethirtyeight', 'ggplot', 'grayscale',  
'seaborn', 'seaborn-bright', 'seaborn-  
colorblind', 'seaborn-dark', 'seaborn-dark-  
palette', 'seaborn-darkgrid', 'seaborn-  
deep', 'seaborn-muted', 'seaborn-notebook',  
'seaborn-paper', 'seaborn-pastel', 'seaborn-  
poster', 'seaborn-talk', 'seaborn-ticks',  
'seaborn-white', 'seaborn-whitegrid',  
'tableau-colorblind10']
```

Установить стиль диаграммы в программе:

```
plt.style.use('grayscale')
```



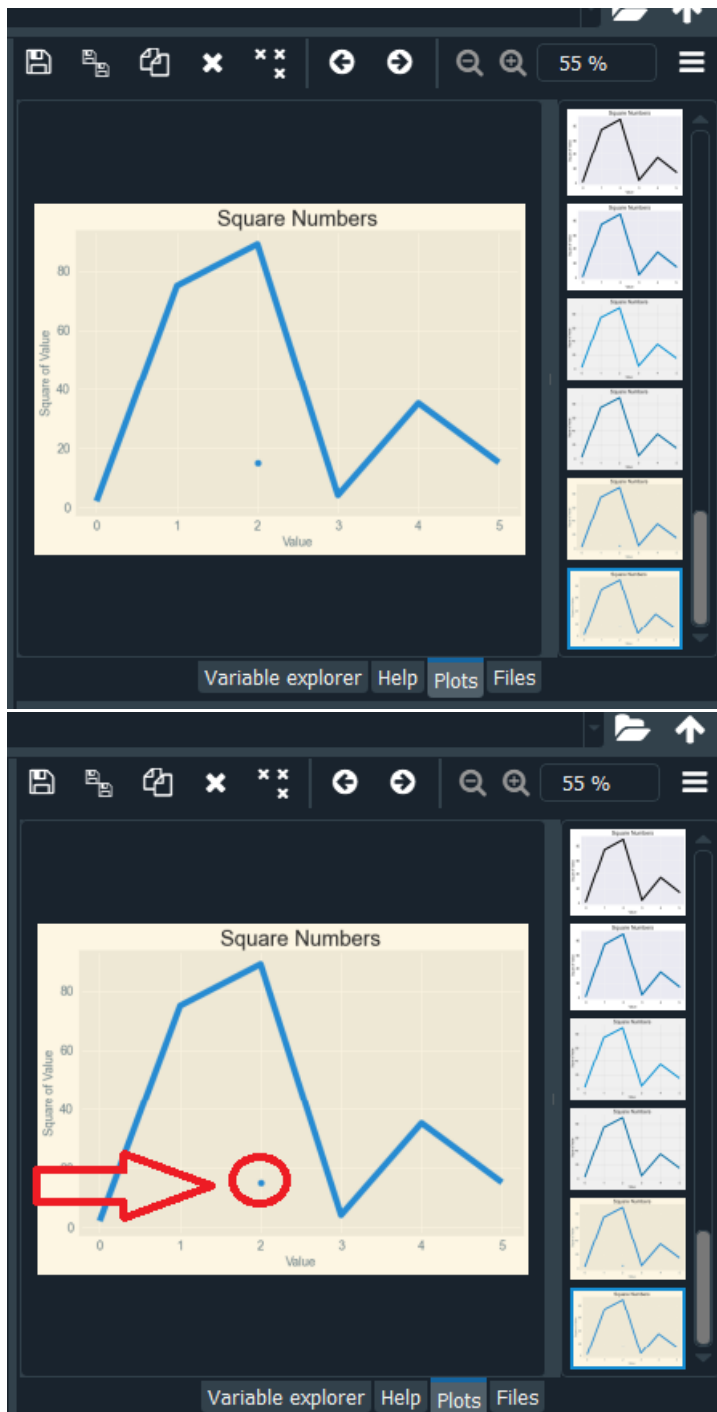
```
plt.style.use('seaborn-darkgrid')
```



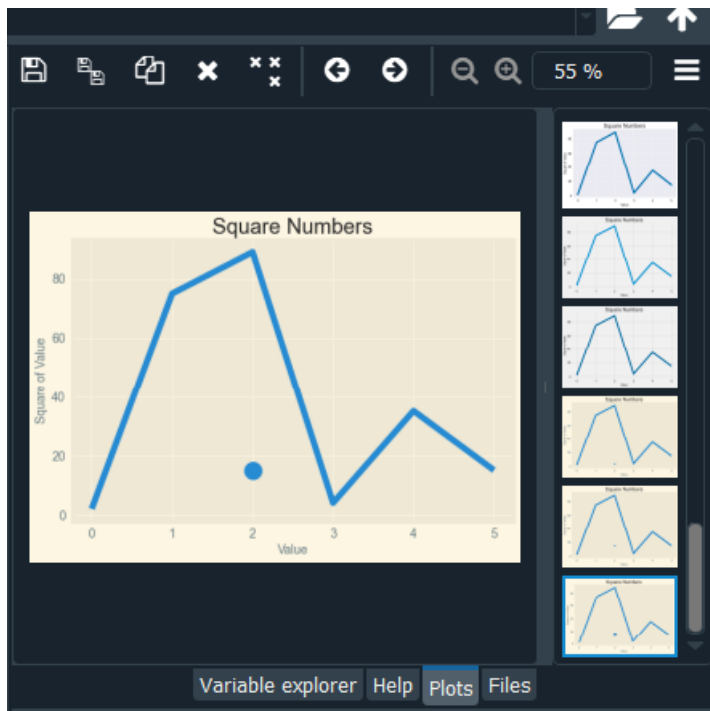
Нанесение и оформление отдельных точек. Функция `scatter()`

`plt.style.use('Solarize_Light2')`

`ax.scatter(2, 15)` # 2, 15 – координаты точки



`ax.scatter(2,15, s = 400)` # увеличим размер точки



Вывод на диаграмме серии точек

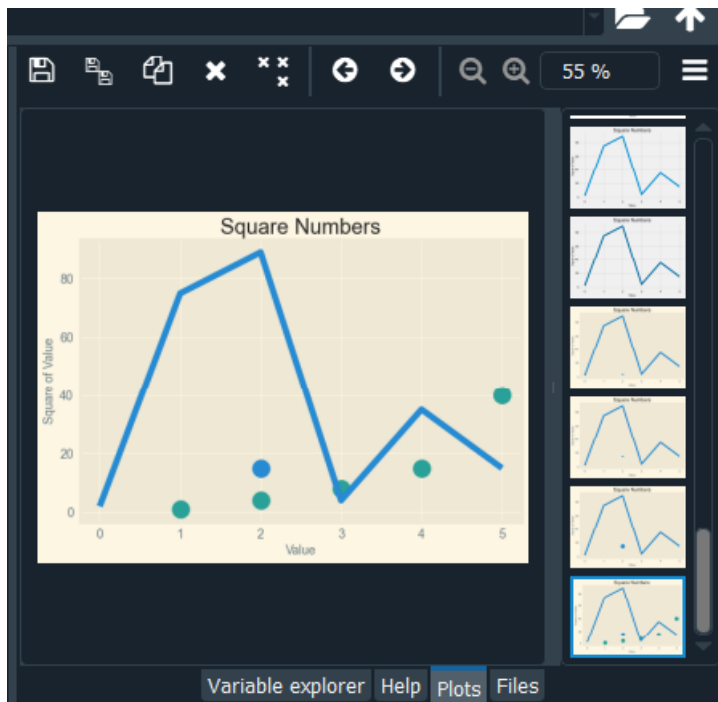
Для этого нужно передать функции `scatter` список значений координат точек.

```
x_values = [1, 2, 3, 4, 5]
```

```
y_values = [1, 4, 8, 15, 40]
```

```
ax.scatter(x_values, y_values, s = 400)
```

```
plt.show()
```

Генерация списка точек

```
import matplotlib.pyplot as plt
```

```
x_values = list(range(1, 1001))
```

```
y_values = [x**2 for x in x_values]
```

```
plt.style.use('seaborn')
```

```
fig, ax = plt.subplots()
```

```
ax.scatter(x_values, y_values, s = 20)
```

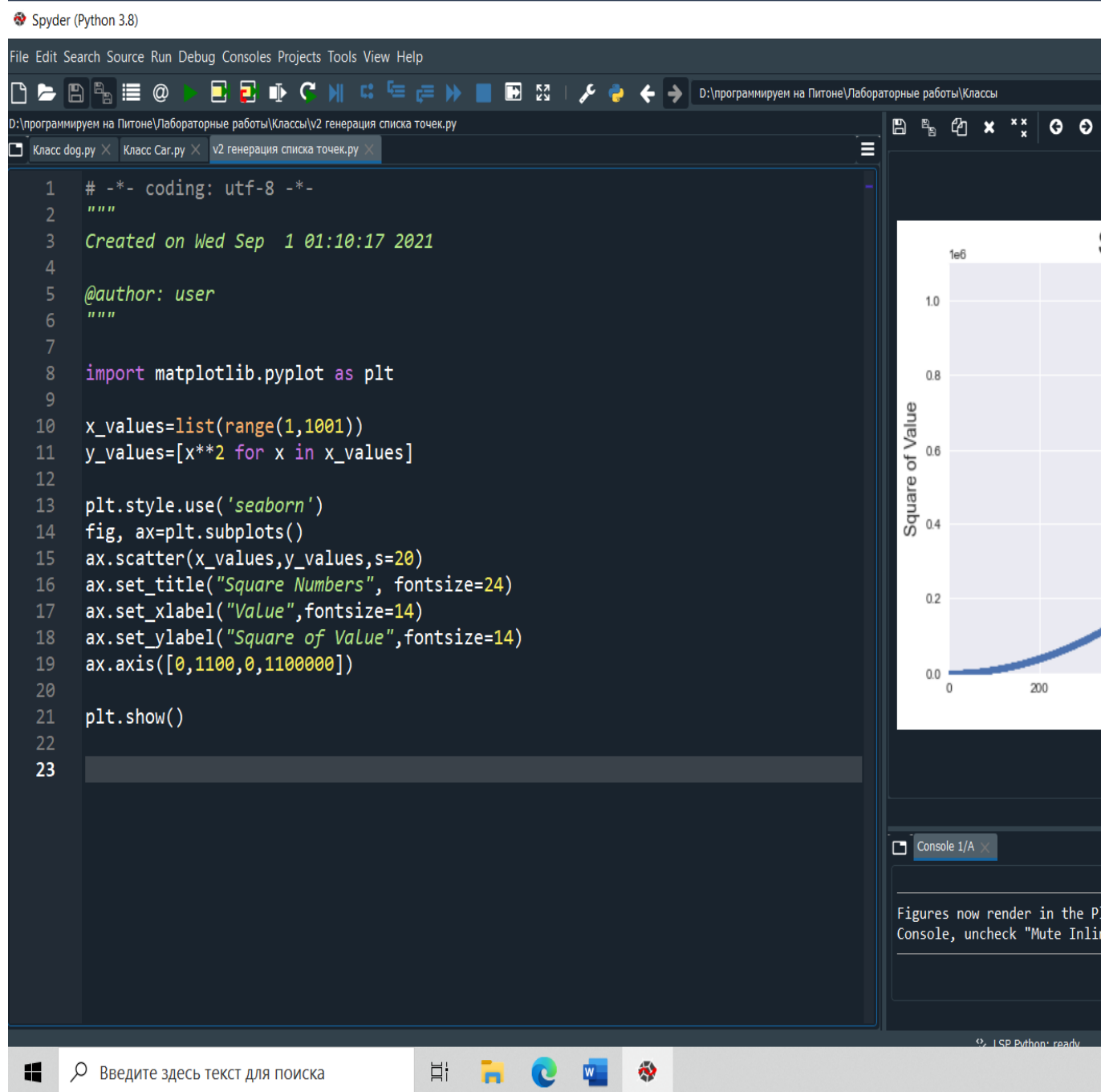
```
ax.set_title("Square Numbers", fontsize = 24)
```

```
ax.set_xlabel("Value", fontsize = 14)
```

```
ax.set_ylabel("Square of Value", fontsize = 14)
```

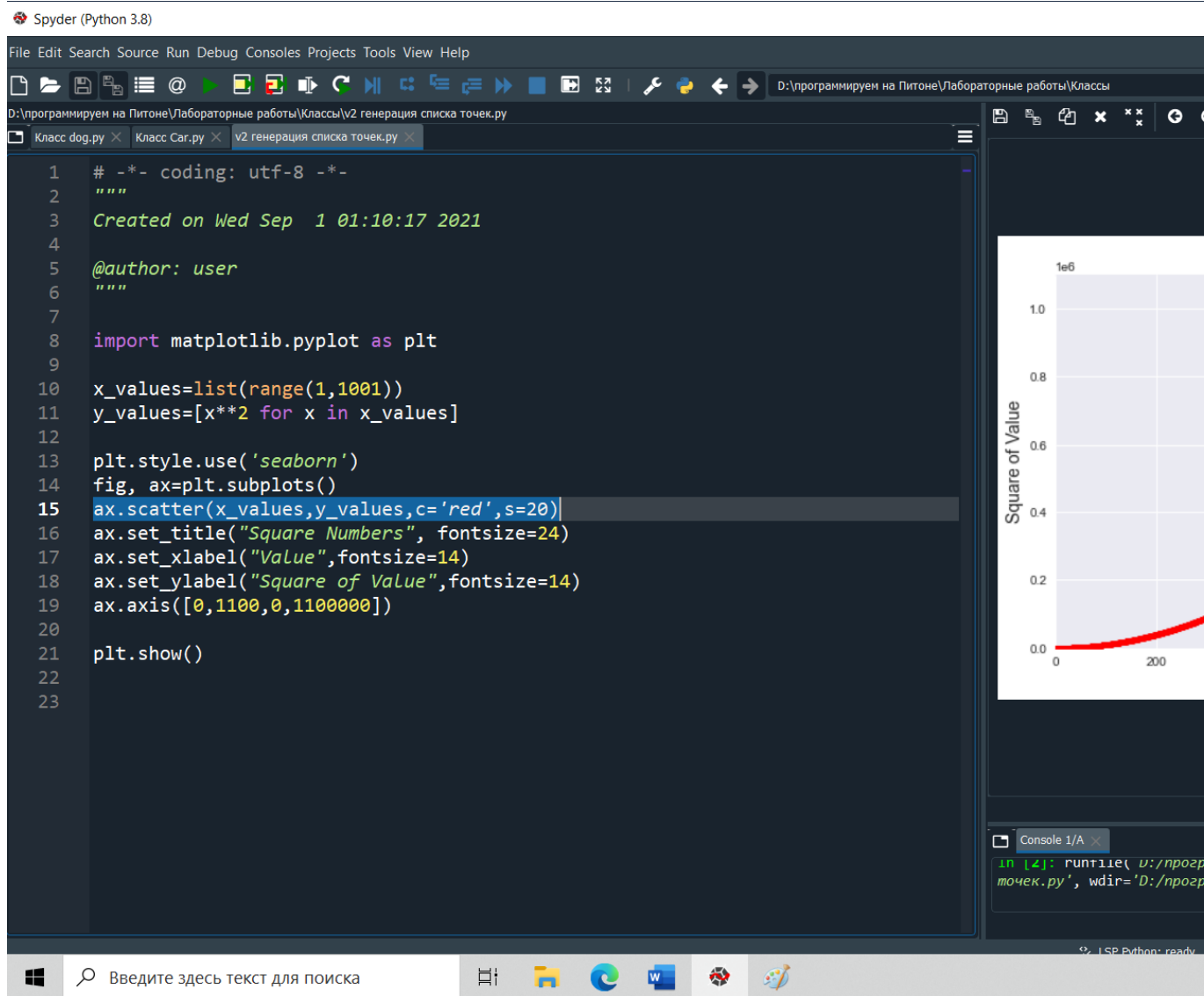
```
ax.axis([0, 1100, 0, 1100000])
```

plt.show()



Изменение цвета точек на диаграмме

`ax.scatter(x_values, y_values, c = 'red', s = 20)`



Цветовые карты (colormap)

Цветовые карты используются в визуализациях для выделения закономерностей в данных.

Например, малые значения можно выделить одним цветом, а большие – другим.

Модуль pyplot включает набор встроенных цветовых карт.

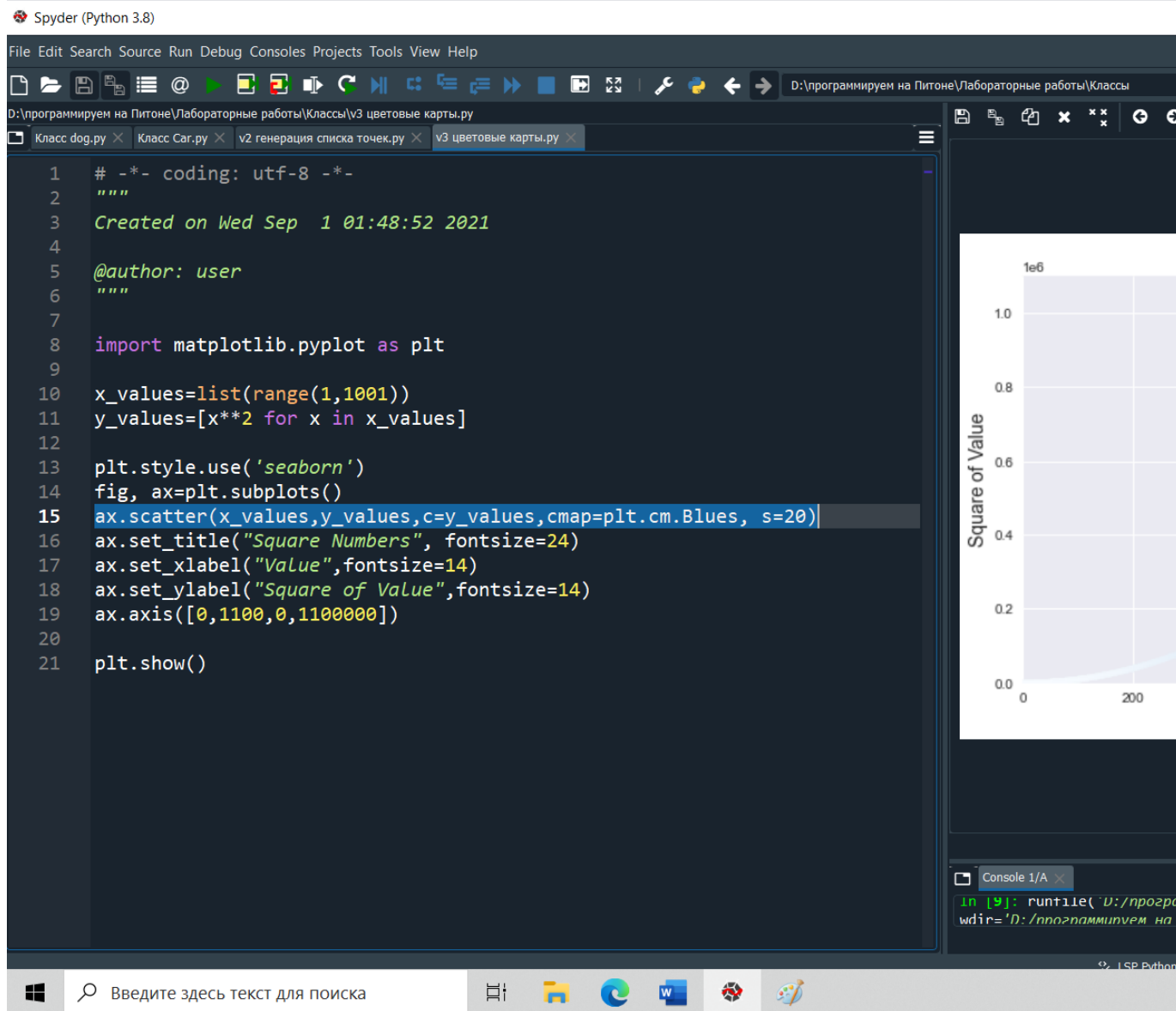
`cmaps['Sequential'] = [`

`'Greys', 'Purples', 'Blues', 'Greens', 'Oranges', 'Reds',
'YlOrBr', 'YlOrRd', 'OrRd', 'PuRd', 'RdPu', 'BuPu',`

'GnBu', 'PuBu']

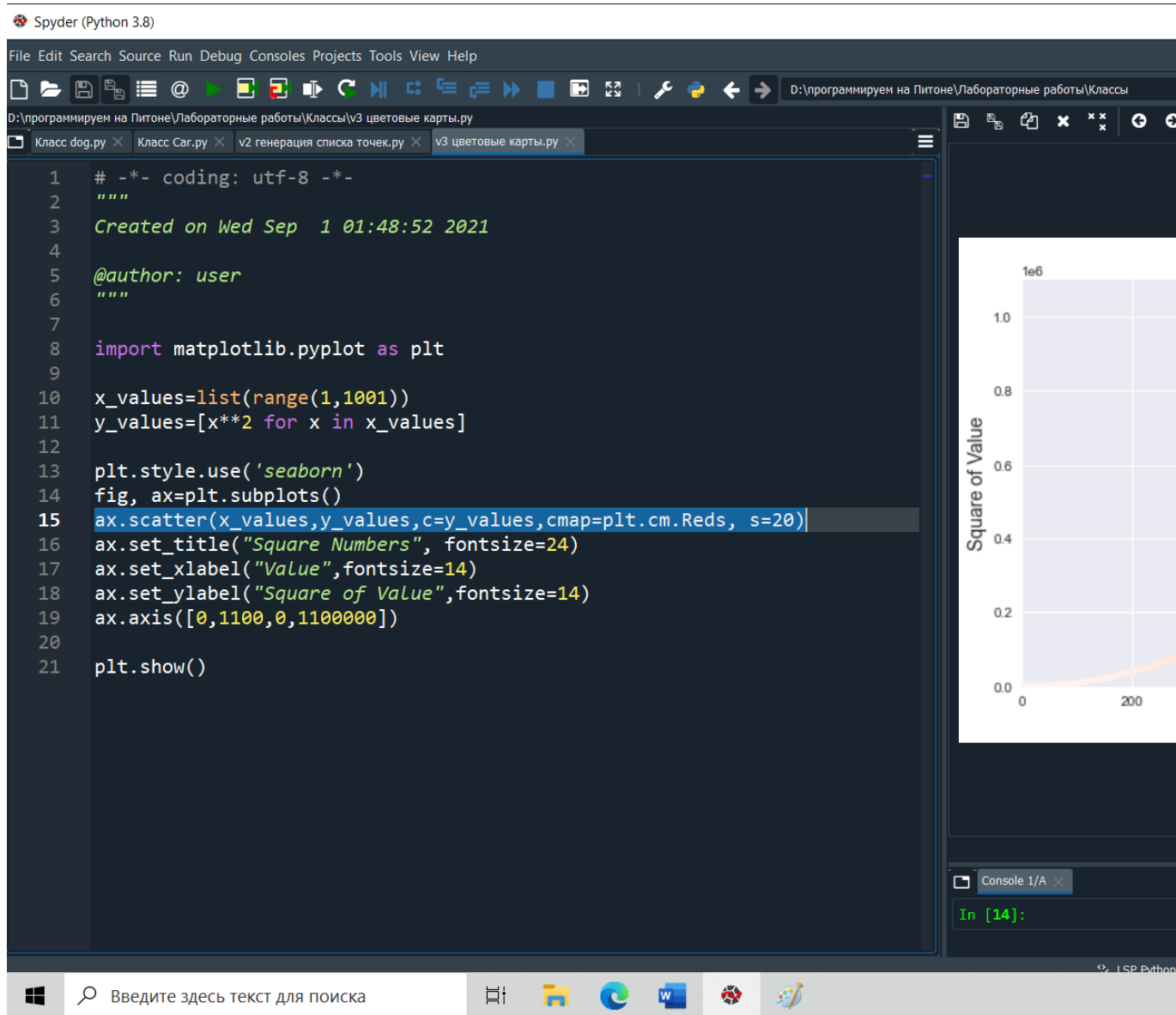
Например, используем синий цвет:

```
ax.scatter(x_values, y_values, c = y_values, cmap = plt.cm.Blues, s = 20)
```



Используем красный цвет:

```
ax.scatter(x_values, y_values, c = y_values, cmap = plt.cm.Red, s = 20)
```



Автоматическое сохранение диаграммы в файле

Для сохранения диаграммы в файле нужно вызов `plt.show()` заменить на вызов `plt.savefig()`:

```
plt.savefig('s_plot.png', bbox_inches = 'tight')
```

's_plot.png' – имя файла,

`bbox_inches = 'tight'` - отсечение лишнего пространства от диаграммы.