Тема 3. Введение в Matplotlib

Введение

matplotlib - набор дополнительных модулей (библиотек) языка Python. Предоставляет средства для построения самых разнообразных 2D графиков и диаграмм данных. Достоинства. Отличается простотой использования - для построения весьма мудреных и красочно оформленных диаграмм достаточно нескольких строк кода. При этом качество получаемых изображений более чем достаточно для их опубликования. Работа этого модуля обычно подразумевает так же использование модуля **NumPy**. Сетевой ресурс

https://github.com/whitehorn/Scientific_graphics_in_python

содержит общедоступный учебник. Много информации по этой теме размещено на http://jenyay.net/Programming/Python3d.

Установка

В Windows установка и matplotlib и NumPy не должна вызвать ни каких проблем. Используем **pip3** - систему управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python.

Если Python, например, установлен в C:\Program Files\Python36\, то в командной строке администратора (это важно) необходимо выполнить следующие команды:

cd “C:\Program Files\Python36\Scripts”

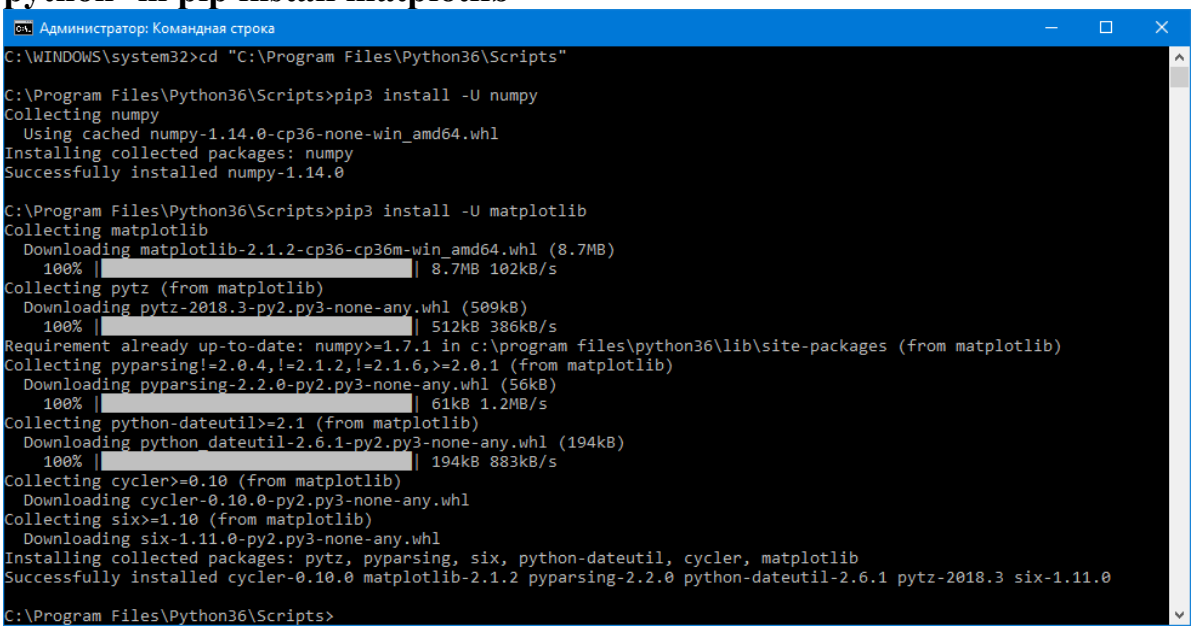
pip3 install **numpy**

pip3 install **matplotlib**

или так

python -m pip install numpy

python -m pip install matplotlib



Протокол работы:

```
C:\WINDOWS\system32>cd "C:\Program Files\Python36\Scripts"

C:\Program Files\Python36\Scripts>pip3 install -U numpy
Collecting numpy
  Using cached numpy-1.14.0-cp36-none-win_amd64.whl
Installing collected packages: numpy
Successfully installed numpy-1.14.0

C:\Program Files\Python36\Scripts>pip3 install -U matplotlib
Collecting matplotlib
  Downloading matplotlib-2.1.2-cp36-cp36m-win_amd64.whl (8.7MB)
    100% |████████████████████████████████| 8.7MB 102kB/s
Collecting pytz (from matplotlib)
  Downloading pytz-2018.3-py2.py3-none-any.whl (509kB)
    100% |████████████████████████████████| 512kB 386kB/s
Requirement already up-to-date: numpy>=1.7.1 in c:\program
files\python36\lib\site-packages (from matplotlib)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)
    100% |████████████████████████████████| 61kB 1.2MB/s
Collecting python-dateutil>=2.1 (from matplotlib)
  Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
    100% |████████████████████████████████| 194kB 883kB/s
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.10.0-py2.py3-none-any.whl
Collecting six>=1.10 (from matplotlib)
  Downloading six-1.11.0-py2.py3-none-any.whl
Installing collected packages: pytz, pyparsing, six, python-dateutil,
cycler, matplotlib
Successfully installed cycler-0.10.0 matplotlib-2.1.2 pyparsing-2.2.0
python-dateutil-2.6.1 pytz-2018.3 six-1.11.0

C:\Program Files\Python36\Scripts>
```

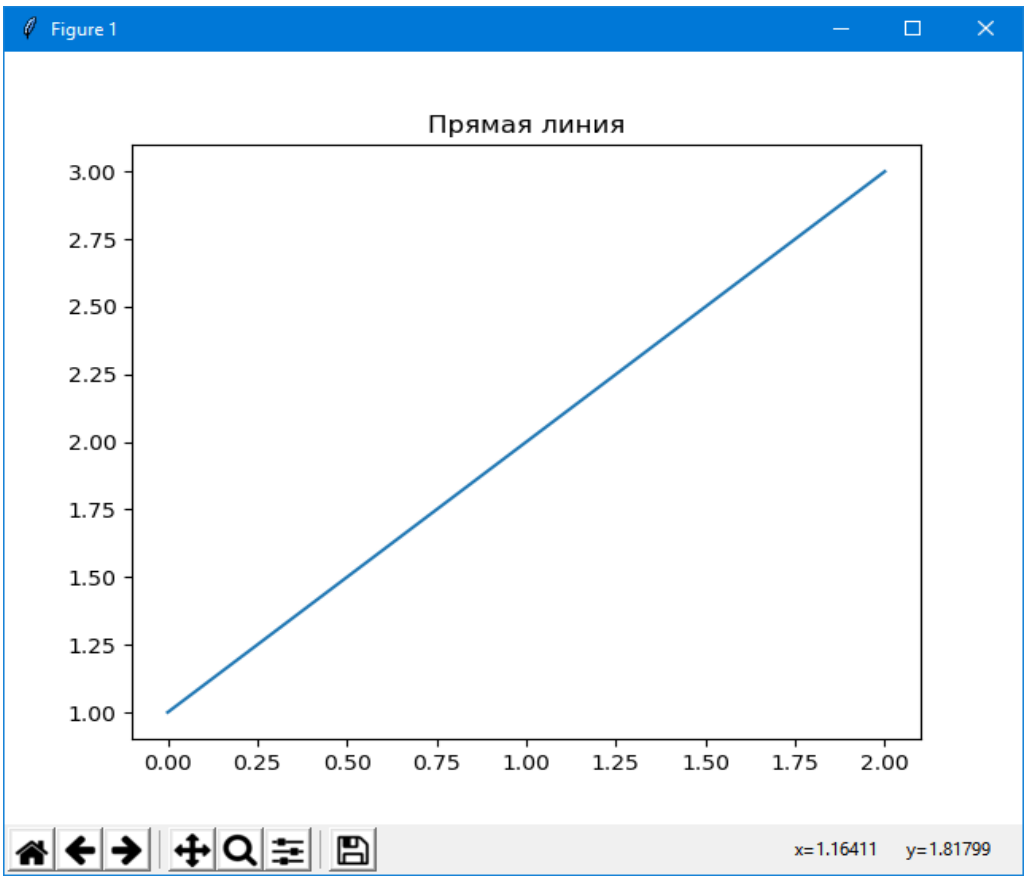
Проверка

После установки проверяем работоспособность. Запускаем консоль Python, вводим:

```
>>> import matplotlib as mpl
>>> print ('Current version on matplotlib library is',
mpl.__version__)
Current version on matplotlib library is 2.1.2

# де-факто стандарт вызова pyplot в python
import matplotlib.pyplot as plt
plt.plot([1,2,3])
[<matplotlib.lines.Line2D object at 0x0301E810>]
plt.title('Прямая линия')
<matplotlib.text.Text object at 0x03062430>
plt.show()
```

В результате должно появиться вот такое окно диаграммы:



Что было сделано?

Из пакета matplotlib импортирован модуль pyplot под именем plt.

Модуль pyplot содержит функции (похожие на команды) создания диаграмм и изменения свойств их элементов.

Функция **`plot`**() строит прямоугольные двумерные диаграммы (графики) в координатах X - Y.

Если функции **`plot`** передан один аргумент (в нашем случае - список [1,2,3]), она рассматривает его как совокупность значений откладываемых по оси Y, тогда по оси X ему будет соответствовать автоматически сгенерированный набор чисел 0,1,2...N - 1, где N - число элементов в переданном списке.

Функция **`title`**() задает заголовок диаграммы, а функция show() выводит интерактивное окно диаграммы. В общем все очень просто

Настройка

Пакет matplotlib можно легко настроить через конфигурационный файл matplotlibrc. Если Python установлен в папку C:\Program Files\Python36\, то настроечный файл располагается в C:\Program Files\Python36**Lib\site-packages\matplotlib\mpl-data\.**

Вносить изменения можно прямо здесь, но лучше скопировать файл в папку пользователя: C:\Documents and Settings\UserName\.matplotlib\, иначе при переустановке пакета конфигурационный файл будет перезаписан.

Параметры пакета задаются в файле в виде пар `свойство : значение`, символ # отделяет комментарий.

Свойство **`font.family`** определяет тип активного шрифта по умолчанию, может принимать пять значений: **`serif`**, **`sans-serif`**, **`cursive`**, **`fantasy`**, **`monospace`**.

В свою очередь свойства **`font.serif`**, **`font.sans-serif`**, **`font.cursive`**, **`font.fantasy`**, **`font.monospace`** содержат списки имен шрифтов (через запятую, в порядке уменьшения приоритета) соответствующих каждому типу.

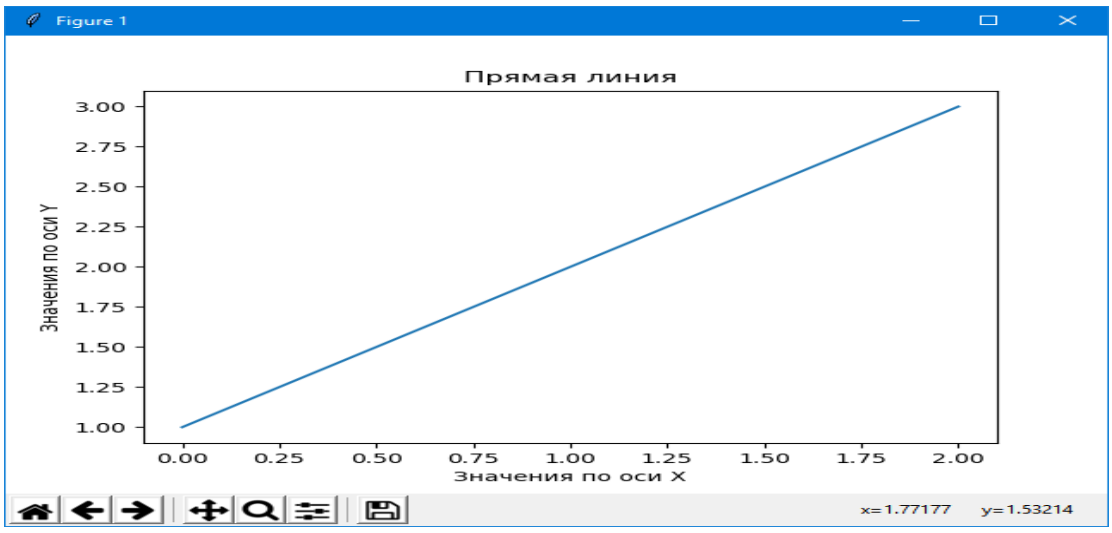
Например, для "русификации" matplotlib возможно придётся изменить имена шрифтов в списке, на имена доступных в системе шрифтов, содержащих кириллические символы. Например так:

```
font.serif : Verdana, Arial
font.sans-serif : Tahoma, Arial
font.cursive : Courier New, Arial
font.fantasy : Comic Sans MS, Arial
font.monospace : Arial
```

Заметим, что последние версии пакета инсталлируются с корректным настроечным файлом.

Рассмотрим пример:

```
>>> plt.plot([1,2,3])
[<matplotlib.lines.Line2D object at 0x000001BB6DFCA588>]
>>> plt.title('Прямая линия')
Text(0.5,1,'Прямая линия')
>>> plt.xlabel(u'Значения по оси X')
Text(0.5,0,'Значения по оси X')
>>> plt.ylabel(u'Значения по оси Y')
Text(0,0.5,'Значения по оси Y')
>>> plt.show()
```



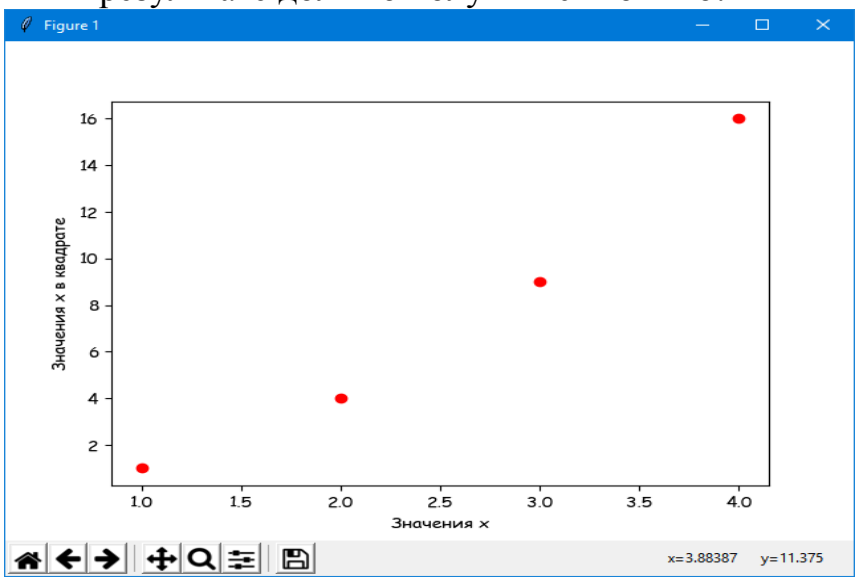
Функции `pyplot.xlabel()`, `pyplot.ylabel()`, как легко догадаться по названию, устанавливают подписи к осям X и Y соответственно.

Пакет matplotlib можно настраивать и динамически (во время выполнения программы). В пакете определена структура `matplotlib.rcParams`, с которой работают как со словарем. Ключевые слова в данном случае - имена свойств файла `matplotlibrc`.

Настройки шрифтов через `matplotlib.rcParams` можно изменить, например, так:

```
>>> import matplotlib as mpl
>>> import matplotlib.pyplot as plt
>>> mpl.rcParams['font.family'] = 'fantasy'
>>> mpl.rcParams['font.fantasy'] = 'Comic Sans MS, Arial'
>>> plt.plot([1,2,3,4], [1,4,9,16], 'ro')
[<matplotlib.lines.Line2D object at 0x000001BB6E03B470>]
>>> plt.xlabel('Значения x')
Text(0.5,0,'Значения x')
>>> plt.ylabel('Значения x в квадрате')
Text(0,0.5,'Значения x в квадрате')
>>> plt.show()
```

В результате должно получиться вот что:



Функции `pyplot.plot()` можно передать произвольное число пар аргументов типов `list` или **`array`** (тип **`array`** определен в модуле **`numpy`**).

В примере выше передана одна пара: [1,2,3,4], [1,4,9,16]. Первый элемент каждой пары функция рассматривает как значения, откладываемые по оси X, второй элемент как значения по оси Y. Соответственно для каждой пары элементов строиться своя кривая на диаграмме.

После каждой пары данных может быть передан дополнительный аргумент типа `string` - строка форматирования, которая определяет внешний вид кривой.

Формат строки позаимствован из системы Matlab. Строка включает три подстроки.

- первая подстрока задает цвет графического элемента,
- вторая стиль маркера,
- третья стиль линии.

В примере 'ro' = 'r' + 'o', где 'r' - красный, 'o' - кружок. По умолчанию 'b-' - синяя непрерывная линия.

Назначении кнопок интерактивного окна диаграммы.

Кнопка 
(pan/zoom) предназначена для прокрутки/масштабирования диаграммы.

Нажимаем на кнопку, указатель мыши приобретает вид перекрещенных стрелок.

Если двигать мышь и удерживать левую кнопку - диаграмма будет прокручиваться по направлению движения указателя.

Если двигать мышь и удерживать правую кнопку - масштаб диаграммы будет изменяться, при движении вверх/вправо - уменьшаться, вниз/влево - увеличиваться.

При нажатых клавишах "**x**" и "**y**" перемещение/масштабирование диаграммы будет происходить по осям X и Y соответственно.

При нажатой клавише **Ctrl** диаграмма будет изменяться так, чтобы сохранились её пропорции (aspect ratio).

Кнопка 
при нажатии включает режим прямоугольного масштабирования.

Если нажать на кнопку указатель мыши приобретет вид креста.

При нажатой левой кнопки мыши на диаграмме можно выделить прямоугольную область. После освобождения кнопки масштаб диаграммы будет изменен так, чтобы выделенная область заняла как можно большую площадь диаграммы.

Кнопки 
позволяют перемещаться по истории изменения диаграммы. Все изменения вносимые в диаграмму пользователем (масштабирование, прокрутка) запоминаются.

Кнопки со стрелками позволяют перемещаться вперед/назад по имеющимся вариантам диаграммы.

Кнопка с домиком возвращает диаграмму к исходному состоянию.

Кнопка 

вызывает стандартный для системы диалог сохранения файла.

Позволяет сохранить диаграмму в файл. Можно выбрать следующие форматы файла диаграммы: **png**, **pdf**, **svg**, **eps**, **ps**.

Примеры построения графиков

Во всех примерах импортируем модули:

```
import matplotlib as mpl
# де-факто стандарт вызова pyplot в python
import matplotlib.pyplot as plt
```