



Matplotlib



Matplotlib

- С помощью визуализации данных можно быстро увидеть тренды в данных и связи между данными
- Matplotlib – одна из наиболее популярных библиотек для рисования графиков в Python



Matplotlib

- Matplotlib лежит в основе других библиотек по рисованию графиков и визуализации в Python – например, seaborn и визуализации, встроенные в Pandas



Matplotlib

- Matplotlib по синтаксису рисования графиков напоминает язык программирования MatLab
- С его помощью можно нарисовать практически любой график, и поменять практически любые элементы графика



Matplotlib

- Однако возможность поменять практически любые элементы графика может быть неудобна для новичков, поскольку синтаксис может выглядеть запутанно.
- Отчасти причина в том, что есть два разных способа создания графиков – функциональными методами, и методами объектно-ориентированного программирования



Matplotlib

- В данном разделе мы постараемся исключить запутанности, чётко разделив эти два подхода:
 - Основы Matplotlib
 - Функциональный метод
 - Фигуры и графики Matplotlib
 - Метод объектно-ориентированного программирования



Matplotlib

- В этом разделе мы пройдем:
 - Основы и функции Matplotlib
 - Matplotlib Figures
 - Matplotlib Subplots
 - Стилизация Matplotlib
 - Упражнения по Matplotlib с решениями



Matplotlib

- Мы не будем изучать специализированные графики, например гистограммы, поскольку мы пройдем их позже в разделе про seaborn
- Важно сначала изучить Matplotlib, поскольку seaborn построен на основе Matplotlib



Matplotlib

- В данном разделе мы будем ссылаться на документацию Matplotlib – она очень хорошая:
 - <https://matplotlib.org>
- А также на полезную галерею графиков и примеров кода:
 - <https://matplotlib.org/stable/gallery/index.html>



Matplotlib

- Мы будем решать две задачи в Matplotlib:
 - Нарисовать функциональную связь переменных
 - $y=2x$
 - Нарисовать график на основе набора точек
 - $x=[1,2,3,4]$
 - $y=[2,4,6,8]$



Давайте начнём!



Основы Matplotlib



Основы Matplotlib

- Самый простой способ рисовать графики в Matplotlib – с помощью функции `plot`:
 - `plt.plot(x,y)`
- Такие вызовы функций просты, однако не позволяют детально настраивать параметры графиков.



Основы Matplotlib

- Так что вызовы `plt.plot()` хороши для быстрой визуализации данных
- Позже мы изучим объектно-ориентированный API Matplotlib, в нём больше возможностей по настройке графиков.



Основы Matplotlib

- Замечание!
 - Построение графиков немного отличается в блокноте Jupyter Notebook и в запуске скрипта Python
 - Если Вы запускаете скрипты .py, а не блокноты .ipynb, то Вам нужно добавить команду `plt.show()` для отображения графика



Объект Matplotlib Figure

часть 1 – figure



Объект Matplotlib Figure

- Объектно-ориентированный API Matplotlib использует объект Figure
- Далее к объекту Figure добавляются оси (axis)
- И на осях уже рисуются графики
- Такой подход позволяет контролировать каждый шаг построения графика



Объект Matplotlib Figure

- Давайте дадим визуальное описание объекта Figure перед тем, как писать код в Python...
- Замечание:
 - объект Figure невидим до тех пор, пока на него не будут добавлены оси



Объект Matplotlib Figure

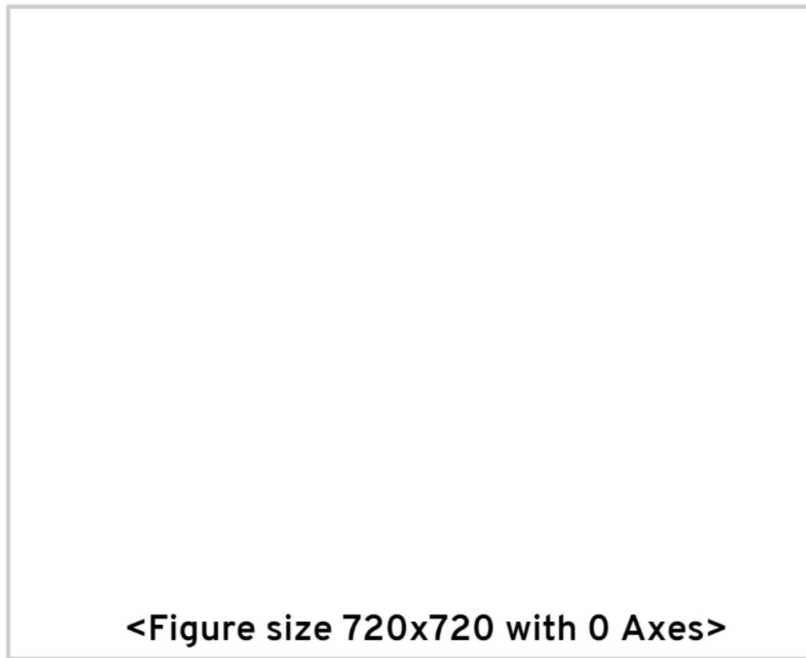
- `plt.figure()`

<Figure size 432x288 with 0 Axes>



Объект Matplotlib Figure

- `plt.figure(figsize=(10,10))`





Объект Matplotlib Figure

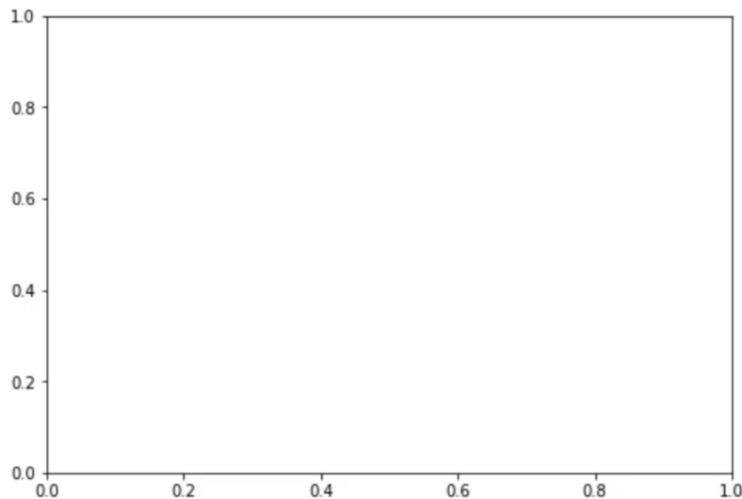
- `fig = plt.figure()`
- Мы получили пустое пространство, на которое можно добавлять оси (axes) для рисования графиков

<Figure size 432x288 with 0 Axes>



Объект Matplotlib Figure

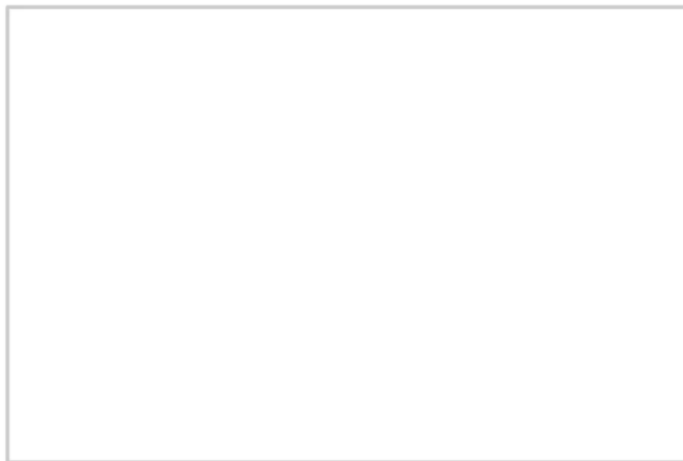
- `fig = plt.figure()`
`axes = fig.add_axes([0,0,1,1])`





Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0,1,1])`



FIGURE

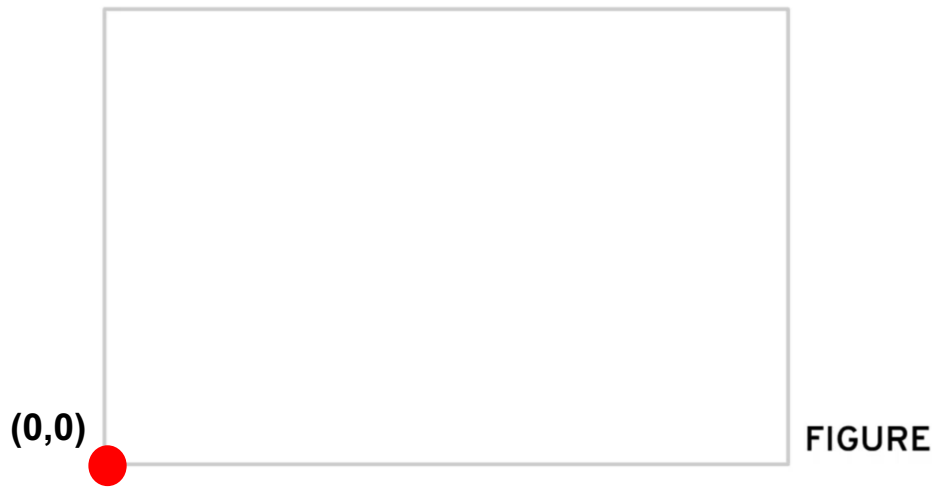


Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0,1,1])`

(x,y)

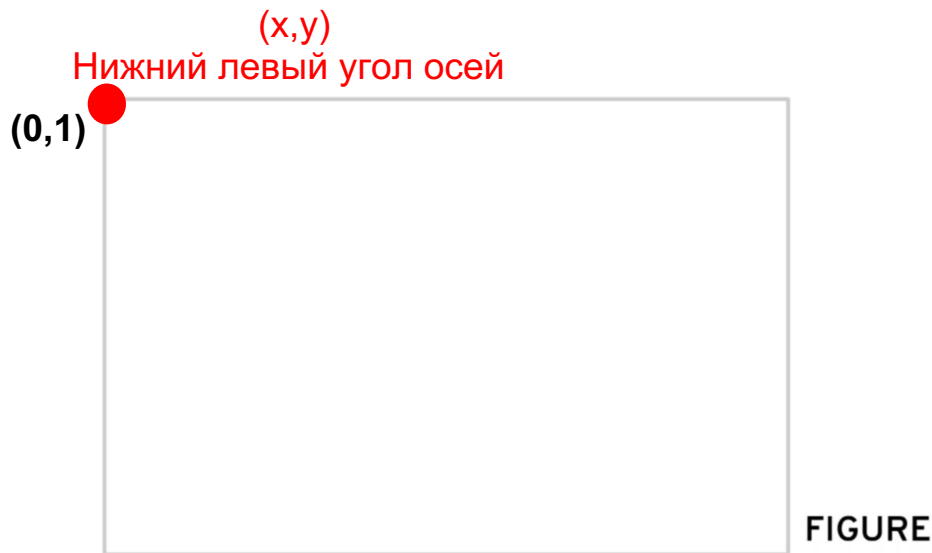
Нижний левый угол осей





Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,1,1,1])`



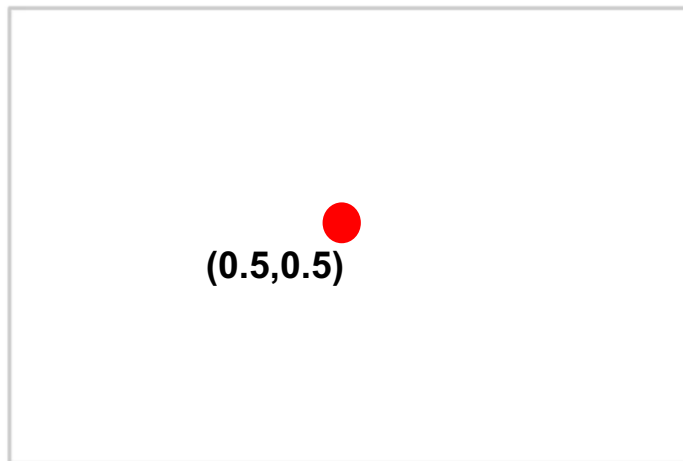


Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([ 0.5, 0.5, 1, 1])`

(x,y)

Нижний левый угол осей



FIGURE



Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0,1,1])`

(width,height)

Ширина и высота осей



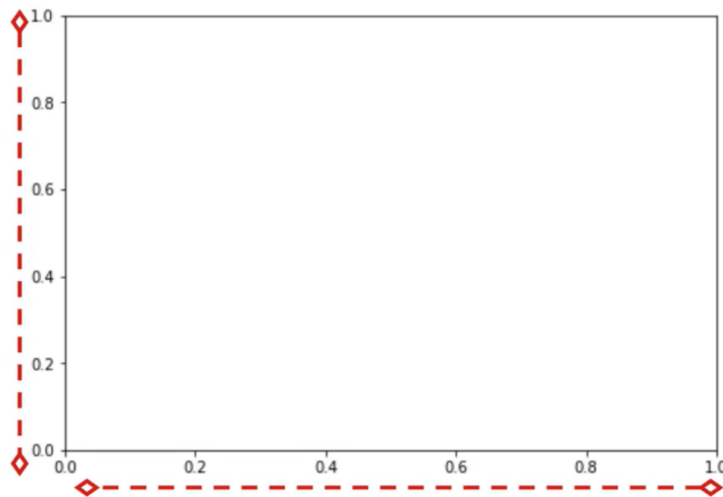


Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0,1,1])`

(width,height)

Ширина и высота осей



FIGURE

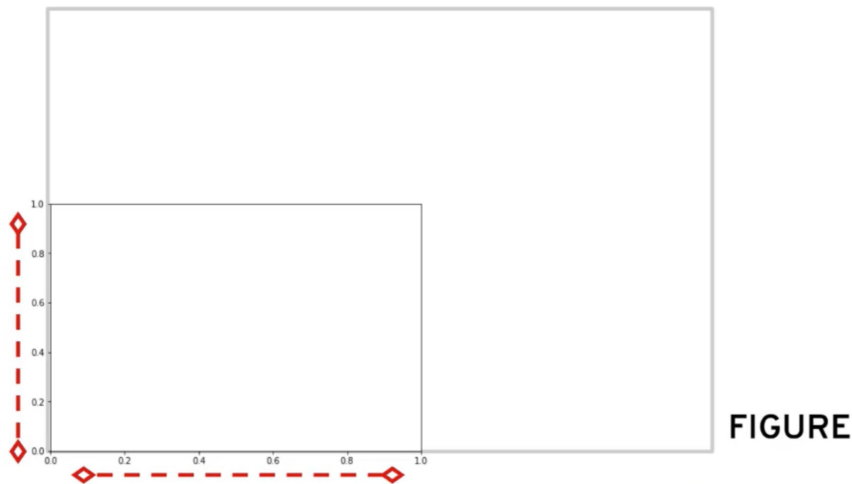


Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 0.5, 0.5])`

(width,height)

Ширина и высота осей



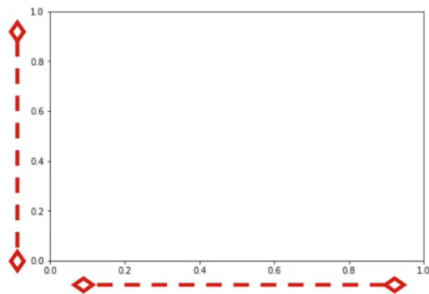


Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 0.5, 0.5])`

(width,height)

Ширина и высота осей





Объект Matplotlib Figure

- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 0.5, 1])`

(width,height)

Ширина и высота осей

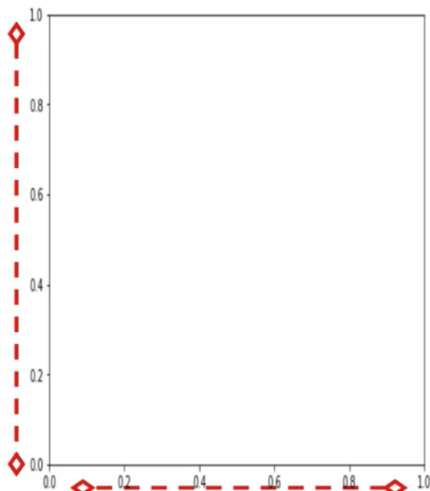


FIGURE



Объект Matplotlib Figure

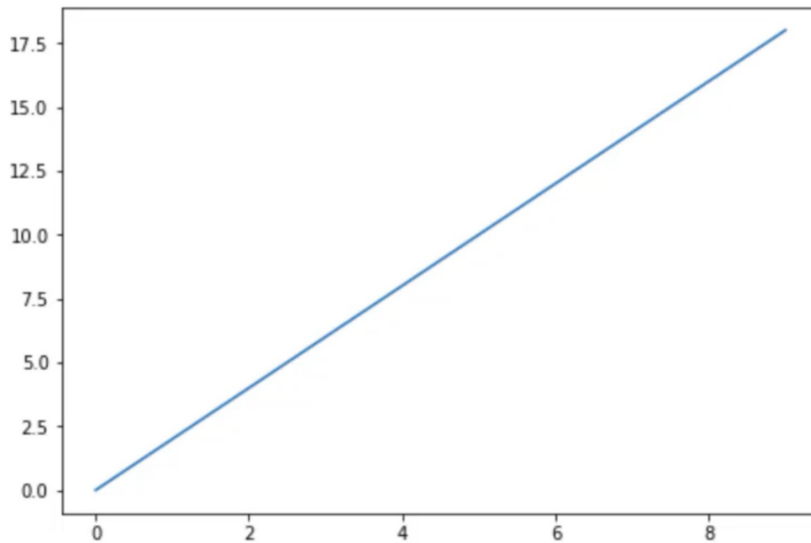
- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 0.5, 1])`
(width,height)
Ширина и высота осей





Объект Matplotlib Figure

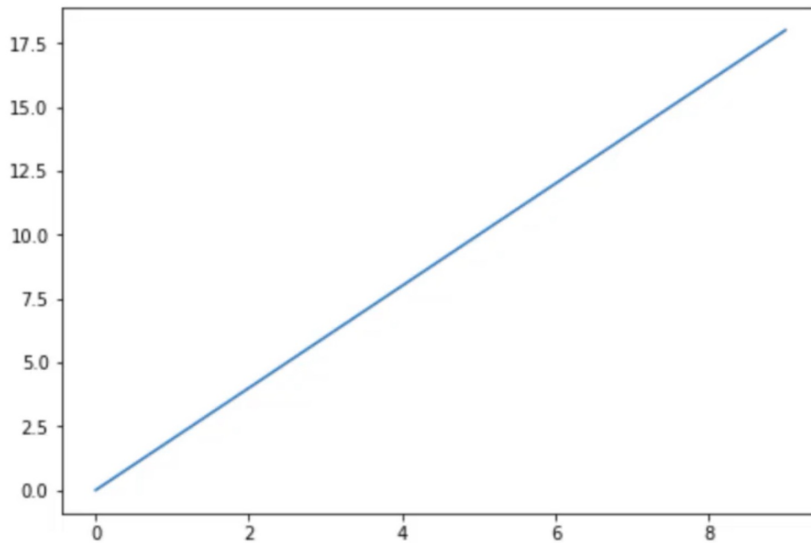
- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 1, 1])`
`axes.plot(x,t)`





Объект Matplotlib Figure

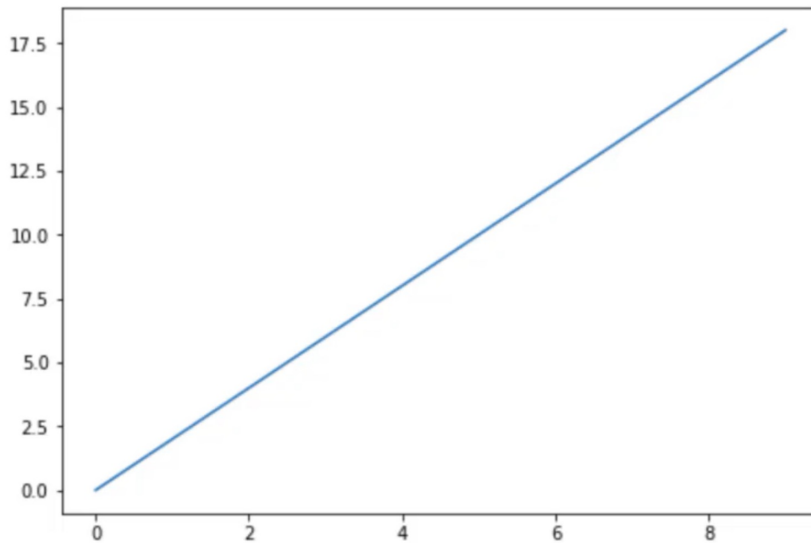
- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 1, 1])`
`axes.plot(x,t)`





Объект Matplotlib Figure

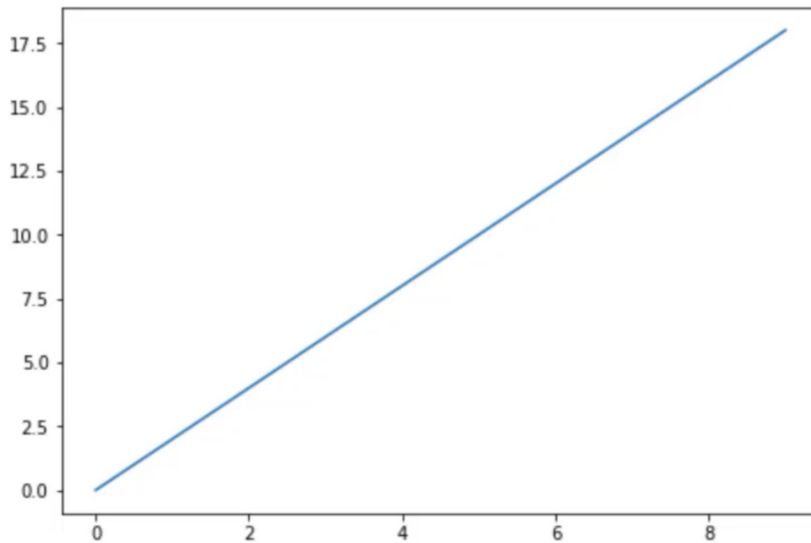
- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 1, 1])`
`axes.plot(x,t)`





Объект Matplotlib Figure

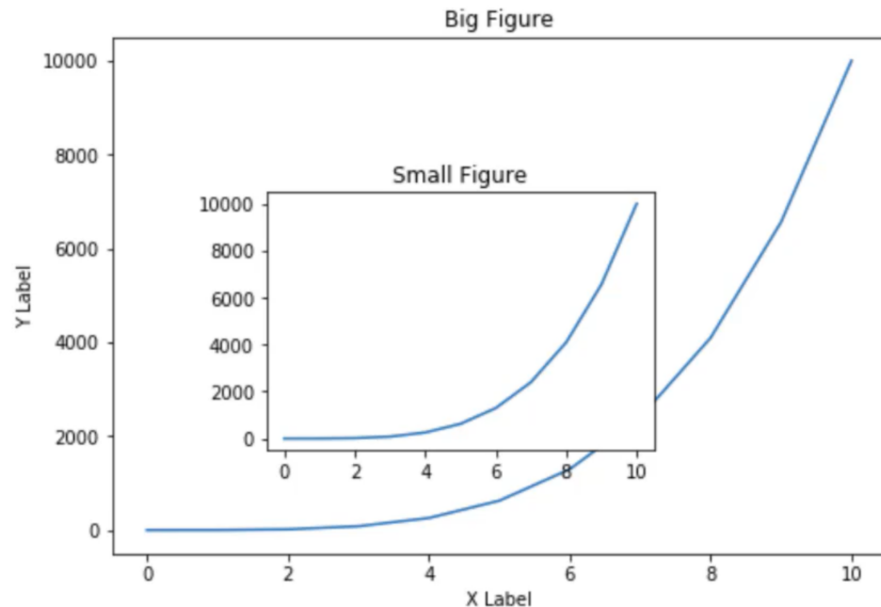
- `fig = plt.figure()`
`axes = fig.add_axes([0,0, 1, 1])`
`axes.plot(x,t)`





Объект Matplotlib Figure

- Такой подход позволяет добавлять несколько наборов осей, менять их положение и размеры





Объект Matplotlib Figure

- Теоретически можно с помощью `plt.figure()` создать несколько наборов осей рядом друг с другом, но удобнее это сделать с помощью `plt.subplots()`
- Позже мы вернёмся к теме построения нескольких графиков рядом друг с другом. А сейчас давайте посмотрим применение объекта Figure!



Объект Matplotlib Figure

часть 2 – figure и axes



Объект Matplotlib Figure

часть 3 – параметры figure



Matplotlib Subplots



Matplotlib Subplots

- Чтобы создать несколько графиков рядом друг с другом, мы могли бы создать объект `Figure` и затем несколько наборов осей рядом друг с другом
- Однако для этой цели в Matplotlib есть готовая функция `plt.subplots()`, которая делает эту работу автоматически!



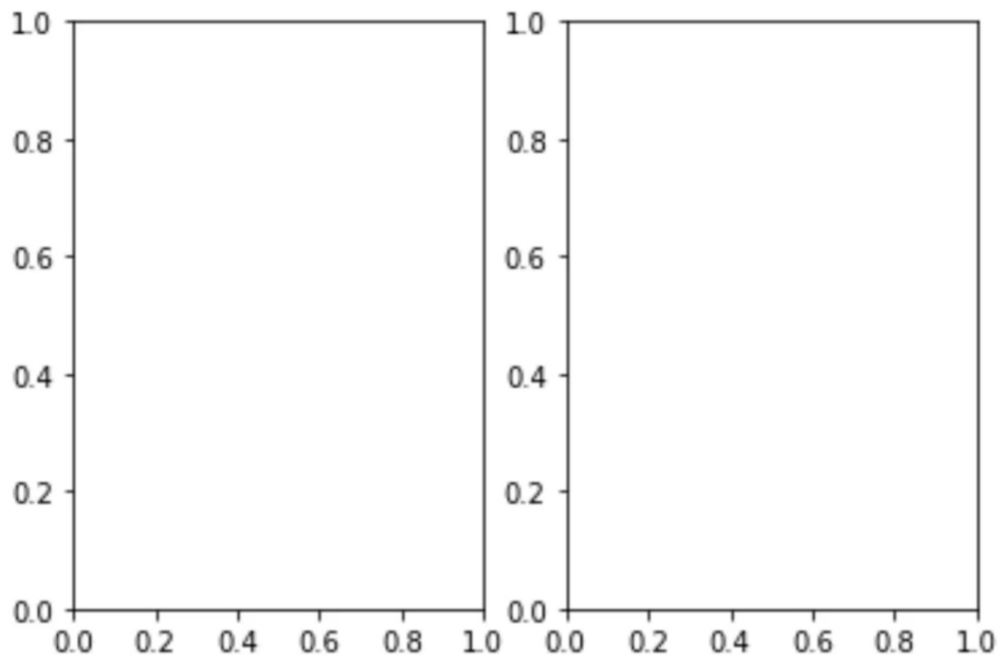
Matplotlib Subplots

- `plt.subplots()` позволяет создавать объекты `figure` и `axes` так, чтобы оси были рядом друг с другом
- `plt.subplots()` возвращает кортеж, в котором находятся объект `figure` и массив `numpy` с объектами `axes`



Matplotlib Subplots

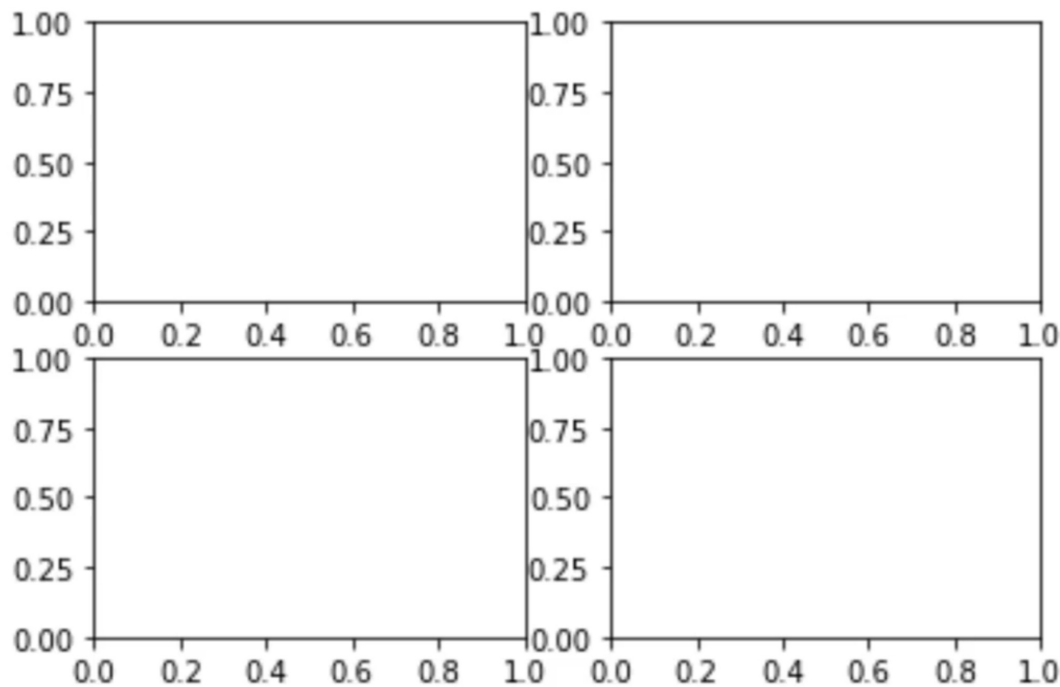
- `fig, axes = plt.subplots(nrows=1, ncols=2)`





Matplotlib Subplots

- `fig, axes = plt.subplots(nrows=2, ncols=2)`





Matplotlib Subplots

- `plt.subplots()` возвращает кортеж, который обычно обозначается `(fig, axes)`:
 - `fig` – это весь объект `Figure`
 - `axes` – массив `numpy` с отдельными парами осей, в порядке их расположения на объекте `figure`



Matplotlib Subplots

- Давайте с помощью `plt.subplots()` нарисуем наборы графиков!



Стилизация Matplotlib

часть 1 - легенда



Стилизация Matplotlib

- В Matplotlib есть разнообразные возможности по стилизации графиков – настройка цветов, легенд, ширины линий, типов маркеров и многое другое!
- Замечание: поскольку вариантов очень много, в этой лекции мы будем с помощью copy-paste копировать код из блокнота, чтобы сэкономить время на ввод кода с клавиатуры во время видео.



Стилизация Matplotlib

- Основные темы по стилизации:
 - Легенды
 - Стилизация внешнего вида
 - Цвета
 - Линии – цвета, ширина, стили
 - Маркеры – цвета, размеры, типы, форма



Стилизация Matplotlib

часть 2



Дополнительные команды Matplotlib



Дополнительные команды Matplotlib

- Matplotlib – очень большая библиотека!
- Мы подготовили для Вас дополнительный блокнот, в котором Вы самостоятельно можете изучить дополнительные темы по Matplotlib
- Эти темы не используются далее в этом курсе, поэтому выбор Ваш – можете посмотреть их, а можете пропустить



Дополнительные команды Matplotlib

- Важно отметить, что ответы на многие вопросы по Matplotlib можно найти в интернете на сайте StackOverflow, или в галерее примеров на сайте Matplotlib
- Используйте эти ресурсы и примеры кода для экономии Вашего времени, чтобы не запоминать те или иные команды!



Упражнения по Matplotlib



Упражнения по Matplotlib

РЕШЕНИЯ